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

Identificação de snoRNAs usando Aprendizagem de Máquina

João Victor de Araujo Oliveira

Dissertação apresentada como requisito parcial para
conclusão do Mestrado em Informática

Orientadora
Prof.ª Dr.ª Maria Emília M. T. Walter

Brasília
2016

Ficha catalográfica elaborada automaticamente,
com os dados fornecidos pelo(a) autor(a)

de Araujo Oliveira, João Victor
dD285i Identificação de snoRNAs usando Aprendizagem de Máquina / João Victor de Araujo Oliveira; orientador Maria Emilia Machado Telles Walter. -- Brasília, 2016.
105 p.

Dissertação (Mestrado - Mestrado em Informática) - Universidade de Brasília, 2016.

1. Bioinformática. 2. Inteligência Artificial. 3. Aprendizagem de Máquina. 4. RNAs não-codificadores. 5. snoRNAs. I. Machado Telles Walter, Maria Emilia, orient. II. Título.

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

Identificação de snoRNAs usando Aprendizagem de Máquina

João Victor de Araujo Oliveira

Dissertação apresentada como requisito parcial para
conclusão do Mestrado em Informática

Prof.ª Dr.ª Maria Emília M. T. Walter (Orientadora)
CIC/UnB

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

Prof. Dr. Marcelo de Macedo Brigido
Universidade de Brasília

Prof.ª Dr.ª Célia Ghedini Ralha
Coordenadora do Programa de Pós-graduação em Informática

Brasília, 29 de Janeiro de 2016

Dedicatória

Dedico este trabalho a todos que me ajudaram e me apoiaram nestes 7 anos de UnB. Primeiramente a Deus, pois nunca me deixou na mão, principalmente nestes tempos de mestrado, onde sempre senti sua companhia em todos os momentos. Também dedico a minha família, por todo incentivo a continuar estudando e, por fim, a minha namorada, amigos e professores, especialmente a professora Maria Emília, por sempre acreditar em meu potencial.

"Comece fazendo o que é necessário, depois o que é possível, e em breve estarás fazendo o impossível." São Francisco de Assis

Agradecimentos

Agradeço as instituições de ensino que me deram a chance de fazer esta dissertação de mestrado: as universidades de Brasília, de Leipzig e de Freiburg, das quais me deram a chance de ver o mundo com outros olhos, melhorando o meu eu cientista e o meu eu humano. Também agradeço ao grupo de bioinformática da UnB e da Universidade de Freiburg, pela amizade e apoio nestes tempos de pesquisa e muito estudo. Por fim agradeço as pessoas que com pequenos gestos e atitudes me ajudaram a seguir em frente, em especial o Dr. Christian Schulz-Huotari e os professores Fabrizio Costa, Rolf Backofen e Jana Hertel.

Obrigado!!

Resumo

Métodos de aprendizagem de máquina vêm sendo amplamente usados na identificação e classificação de diferentes famílias de RNAs não-codificadores (*ncRNAs*). Muitos desses métodos são baseados na aprendizagem supervisionada, onde atributos anteriormente conhecidos, chamados *features*, são extraídos de uma sequência e usados em um classificador. Nesta dissertação, apresentamos dois métodos para a identificação das duas classes principais de *snoRNAs*, *C/D box* e *H/ACA box snoRNAs*: snoReport 2.0, uma melhoria significativa da primeira versão do snoReport; e o snoRNA-EDeN, um novo método baseado no EDeN, que é um *kernel* decomposicional de grafos. O snoReport 2.0 é um método que, usando *features* extraídas de sequências candidatas em genomas, combina predição de estrutura secundária de *ncRNAs* com Máquina de Vetores de Suporte (*Support Vector Machine - SVM*), para identificar *C/D box* e *H/ACA box snoRNAs*. Seu classificador de *H/ACA box snoRNA* mostrou um F-score de 93% (uma melhoria de 10% em relação à primeira versão do snoReport), enquanto o classificador de *C/D box snoRNA* obteve F-score de 94% (melhoria de 14%). Alem disso, ambos os classificadores tiveram todas as medidas de performances acima de 90%. Na fase de validação, o snoReport 2.0 identificou 67,43% dos *snoRNAs* de vertebrados de ambas as classes. Em Nematóides, o snoReport 2.0 identificou 29,6% dos *C/D box snoRNAs* e 69% dos *H/ACA box snoRNAs*. Para as Drosofilídeas, foram identificados 3,2% dos *C/D box snoRNAs* e 76,7% dos *H/ACA box snoRNAs*. Esses resultados mostram que o snoReport 2.0 é eficiente na identificação de *snoRNAs* em organismos vertebrados, e também para *H/ACA box snoRNAs* de organismos invertebrados. Por outro lado, em vez de usar *features* de uma sequência (em geral, difíceis de identificar), uma abordagem recente de aprendizagem de máquina é descrita a seguir. Dada uma região de interesse de uma sequencia, o objetivo é gerar um vetor esparso que pode ser usado como *micro-features* em algum algoritmo de aprendizado de máquina, ou pode ser usado para a criação de *features* poderosas. Essa abordagem é usada no EDeN (*Explicit Decomposition with Neighbourhoods*), um *kernel* decomposicional de grafos baseado na técnica *Neighborhood Subgraph Pairwise Distance Kernel* (NSPDK). O EDeN transforma um grafo em um vetor esparso, decompondo-o em todos os pares de subgrafos vizinhos de raios pequenos, a distâncias crescentes. Baseado no

EDeN, foi desenvolvido um método chamado snoRNA-EDeN. Na fase de testes, para *C/D box snoRNAs*, o snoRNA-EDeN obteve um F-score de 93,4%, enquanto que para *H/ACA box snoRNAs* o F-score foi de 85.12%. Na fase de validação, para *C/D box snoRNA*, o snoRNA-EDeN mostrou uma grande capacidade de generalização, identificando 94,61% de *snoRNAs* de vertebrados e 63,52% de invertebrados, um resultado significantemente melhor em comparação ao snoReport 2.0, que identificou apenas 52,92% dos vertebrados e 14,6% dos invertebrados. Para o *H/ACA box*, o snoReport 2.0 identificou 79,9% dos *snoRNAs* de vertebrados e 73,3% dos *snoRNAs* de Nematóides e Drosofilídeos, enquanto que o snoRNA-EDeN identificou 95,4% dos vertebrados e 57.8% dos nematóides e drosofilas. Ambos os métodos estão disponíveis em: http://www.biomol.unb.br/snoreport e http://www.biomol.unb.br/snorna_eden.

Palavras-chave: RNAs não codificadores, snoRNAs, snoReport, Aprendizagem de Máquina, SVM, Kernel decomposicional de grafos, EDeN

Abstract

Machine learning methods have been widely used to identify and classify different families of non-coding RNAs. Many of these methods are based on supervised learning, where some previous known attributes, called features, are extracted from a sequence, and then used in a classifier. In this work, we present two methods to identify the two main classes of snoRNAs, C/D box and H/ACA box: snoReport 2.0, a significant improvement of the original snoReport version; and snoRNA-EDeN, a new method based on EDeN, a decompositional graph kernel. On one hand, snoReport 2.0 is a method that, using features extracted from candidate sequences in genomes, combines secondary structure prediction with Support Vector Machine (SVM) to identify C/D box and H/ACA box snoRNAs. H/ACA box snoRNA classifier showed a F-score of 93% (an improvement of 10% regarding to the previous version), while C/D box snoRNA classifier a F-Score of 94% (improvement of 14%). Besides, both classifiers exhibited performance measures above 90%. In the validation phase, snoReport 2.0 predicted 67.43% of vertebrate organisms for both classes. SnoReport 2.0 predicted: for Nematodes, 29.6% of C/D box and 69% of H/ACA box snoRNAs; and for Drosophilids, 3.2% of C/D box and 76.7% of H/ACA box snoRNAs. These results show that snoReport 2.0 is efficient to identify snoRNAs in vertebrates, and also H/ACA box snoRNAs in invertebrates organisms. On the other hand, instead of using known features from a sequence (difficult to find in general), a recent approach in machine learning is described as follows. Given a region of interest of a sequence, the objective is to generate a sparse vector that can be used as micro-features in a specific machine learning algorithm, or it can be used to create powerful features. This approach is used in EDeN (Explicit Decomposition with Neighbourhoods), a decompositional graph kernel based on Neighborhood Subgraph Pairwise Distance Kernel (NSPDK). EDeN transforms one graph in a sparse vector, decomposing it in all pairs of neighborhood subgraphs of small radius at increasing distances. Based on EDeN, we developed a method called snoRNA-EDeN. On the test phase, for C/D box snoRNAs, snoRNA-EDeN showed a F-score of 93.4%, while for H/ACA box snoRNAs, the F-score was 72%. On the validation phase, for C/D box snoRNAs, snoRNA-EDeN showed a better capacity of generalization, predicting 94.61% of vertebrate C/D box snoRNAs and

63.52% of invertebrates, a significantly better result compared to snoReport 2.0, which predicted only 52.92% of vertebrates and 14.6% of invertebrates. For H/ACA box snoRNAs, snoReport 2.0 predicted 79.9% of vertebrate snoRNAs and 73.3% of Nematode and Drosophilid sequences, while snoRNA-EDeN predicted 95.4% of vertebrate snoRNAs and 57.8% of Nematode and Drosophilid sequences. Both methods are available at http://www.biomol.unb.br/snoreport and http://www.biomol.unb.br/snorna_eden.

Keywords: non-coding RNAs, snoRNAs, snoReport, Machine Learning, SVM, Decompositional Graph Kernel, EDeN

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Em 1953, os estudos de Watson e Crick [91] permitiram estabelecer a estrutura espacial da molécula de DNA. Anos mais tarde, na década de 1990, o projeto Genoma Humano [47, 85] iniciou um novo período de pesquisas do DNA de diversas espécies. Esse conhecimento serviu como base para os atuais projetos de sequenciamento genômico, que vem possibilitando ampliar o conhecimento das funções de diversas moléculas em organismos.

A Biologia Molecular é a área responsável pelo estudo de ácidos nucléicos, estrutura de proteínas, processos relacionados e outros atores envolvidos, tais como organelas celulares e enzimas [11]. Os ácidos nucléicos têm a principal função de armazenar informação necessária, prover mecanismos para a produção de proteínas e também de possibilitar a transferência desta informação para outros organismos, através de processos de reprodução celular [78]. Na natureza encontramos dois tipos de ácidos nucléicos, o DNA (ácido desoxirribonucléico) e o RNA (ácido ribonucléico). Proteínas são moléculas formadas por um conjunto de aminoácidos e exercem uma vasta quantidade de funções vitais para os seres vivos, tais como acelerar reações químicas (como é o caso das enzimas), transportar nutrientes, eliminar resíduos tóxicos e também construir estruturas complexas [78].

O avanço dos estudos de Watson e Crick permitiu que fosse proposto o Dogma Central da Biologia Molecular [19], que mostra como determinadas regiões da molécula de DNA, através do processo de transcrição, são transformadas em uma molécula de RNA mensageiro (*mRNA*) e esse, pelo processo de tradução, é sintetizado em uma proteína, por meio de dois RNAs, o ribossomal (*rRNA*) e o transportador (*tRNA*).

Contudo, grande parte do material do DNA não codifica proteínas (por exemplo, no ser humano, apenas 2% do genoma gera RNAs codificadores de proteína [81]). Tais regiões são denominadas RNAs não-codificadores de proteínas (*ncRNAs*), que hoje são objeto de pesquisa em todo o mundo. Por meio de experimentos biológicos, foi possível perceber uma relação de ncRNAs com a regulação gênica, funções catalíticas e estruturais, sendo algumas delas relacionadas com a transcrição e tradução de *mRNAs*, como o *tRNA* e o

rRNA [81, 25, 34, 58].

Métodos computacionais para a identificação e classificação de *ncRNAs* vêm sendo propostos e aprimorados nos últimos anos [63, 34, 5, 95, 13]. A tarefa de identificar e classificar *ncRNAs* é bastante desafiadora, devido a dificuldade de confirmar experimentalmente a função de um *ncRNA*, pois essa está associada à sua estrutura espacial (estruturas secundária e terciária), o que impede o uso de métodos de predição de genes codificadores de proteínas que usam apenas a informação de sua estrutura primária (sequência de nucleotídeos).

A identificação de *ncRNAs* também motivou os estudos em diversos organismos, com o objetivo de construir diferentes classes de *ncRNAs*. Em particular, os *Small nucleolase RNAs* (*snoRNAs*), são pequenos *ncRNAs* de tamanho variando de 60 a 300 nucleotídeos que se acumulam no nucléolo[1] e realizam modificações químicas em outros RNAs como em RNA transportador (*tRNA*), RNA ribossomal (*rRNA*) e *Small nuclear RNAs* (*snRNA*). Eles são classificados de acordo com elementos característicos de sequência, denominados *boxes*, em duas classes principais: *H/ACA box snoRNAs* e *C/D box snoRNAs* [25]. As Figuras 1.1 e 1.2 mostram, respectivamente, as estruturas secundárias de *H/ACA box snoRNAs* e *C/D box snoRNAs*.

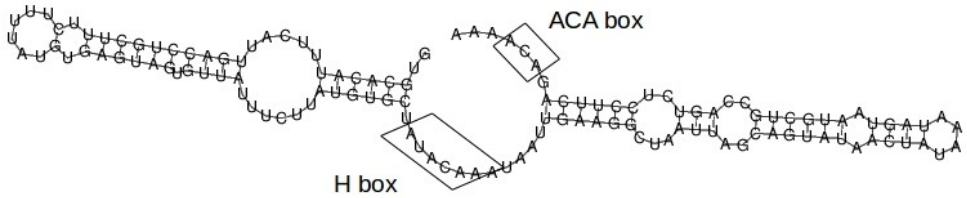


Figura 1.1: Estrutura bidimensional de *H/ACA box snoRNA*.

Um método eficiente usado para a identificação de ambas as classes de *snoRNAs* é o *snoReport* [34]. Esta ferramenta usa a combinação de predição de estrutura secundária e uma técnica aprendizagem de máquina (neste caso, máquina de vetores de suporte, abreviada como SVM). Em contraste com métodos anteriores de identificação de *snoRNAs*

[1] nucléolo: local dentro do núcleo celular em células de eucariotos, onde o *rRNA* é sintetizado [14, 18].

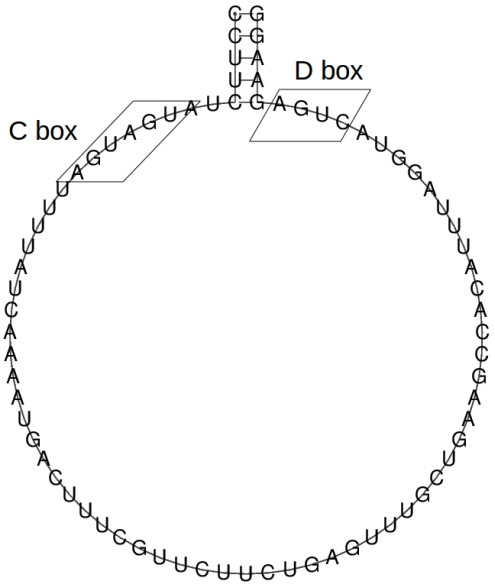


Figura 1.2: Estrutura bidimensional de *C/D box snoRNA*.

(com exceção do *SnoSeeker* [95]), a predição feita pelo snoReport não usa informação de regiões de complementariedade entre snoRNAs e RNAs alvo, também chamadas de região antisenso (esta informação pode dramaticamente aumentar a performance de métodos de identificação de *snoRNAs*). Entretanto, muitos snoRNAs órfãos[2] vêm sendo descobertos, não possuindo região antisenso conhecida, ou possuindo complementariedade com RNAs mensageiros (*mRNA*) específicos, o que sugere outras funções, por exemplo, interferência na edição *A-to-I* [87, 34, 43, 25, 45]. Logo, tais *snoRNAs* poderiam não ser identificados em métodos que usam informação de região alvo [34, 43].

Por outro lado, métodos de aprendizado tais como os usados no *snoReport* têm sido amplamente usados na identificação e classificação de diferentes classes de *ncRNAs* [34, 95, 94, 86, 63]. Muitos desses métodos são baseados em aprendizado supervisionado, onde alguns atributos previamente conhecidos, chamados *features*, são coletados de uma sequência de RNA, de suas estruturas primária e secundária e, então, usados em um classificador. Nesta modalidade de aprendizado, cada amostra possui uma classe, ou rótulo, junto com seus atributos.

Uma técnica recente de aprendizagem de máquina aplicada à Bioinformática é descrita a seguir. Dada uma região de interesse de uma sequência, o objetivo é a geração de um vetor esparso que pode ser usado como *micro-features* em algoritmos de aprendizagem de máquina, ou pode ser utilizado para criar *macro-features* (formada por *micro-features*). Um método que usa esta abordagem é o *EDeN* (*Explicit Decomposition with Neighbourhoods*), um *kernel* decomposicional de grafos baseado na técnica *Neighborhood*

[2] *snoRNAs* órfãos não possuem complementaridade com *rRNAs* e *snRNAs* [34, 95]

Subgraph Pairwise Distance Kernel (NSPDK) [15], que transforma um grafo em um vetor esparso de números reais, decompondo-o em todos os pares de subgrafos vizinhos de raios pequenos, a distâncias crescentes. Esse vetor esparso pode ser usado como conjunto de *features* em algoritmos de aprendizagem de máquina.

1.1 Motivação

Identificar *snoRNAs* em organismos é uma tarefa importante pois ajuda na caracterização de um genoma, revelando similaridades e distinções entre organismos. Além disso diversos *snoRNAs* não canônicos vem sendo descobertos sugerindo diferentes funções, por exemplo: o SNORD115 (HBII-52), que mostrou complementariedade com o *pre-mRNA* receptor da serotonina 2C, causando um *splicing* alternativo deste *pre-mRNA*; e alguns snoRNAs como o SNORD32A (U32A), SNORD33 (U33), and SNORD35A (U35A) que, embora canônicos (possuem complementariedade com *rRNAs*), se acumulam no citosol em situações de estresse celular, sugerindo a atuação dos snoRNAs fora do nucléolo [25].

O *snoReport*, em sua primeira versão, foi treinado exclusivamente com sequências de mamíferos, porém, está disponível um volume bem maior de snoRNAs nos dias atuais, não apenas de mamíferos, mas de diversos vertebrados. Além disso, novas versões de vários componentes usados no *snoReport*, como o pacote Vienna RNA [35] e a biblioteca libSVM [10], trazem novas informações que podem aprimorar o método.

O surgimento de novas técnicas de aprendizagem de máquina, voltadas para Bioinformática, como o EDeN, podem ser utilizadas para aumentar a capacidade de generalização dos algoritmos de predição de snoRNAs, auxiliando na descobertas de novos *snoRNAs*, com funções diferentes das conhecidas.

1.2 Problemas

O método do *snoReport*, em sua primeira versão, foi treinado apenas com sequências de mamíferos e usa meta-parâmetros *default* da SVM. Além disso, utiliza em seu conjunto de treinamento negativo, sequências de *miRNAs* que podem ser derivadas de *snoRNAs* [83], o que pode inserir erro no classificador SVM.

Além disso, visto que o *snoReport* utiliza atributos conhecidos de *snoRNAs* canônicos, a capacidade de descobrir *snoRNAs* com diferentes propriedades ou funções é diminuída, o que sugere o uso de diferentes algoritmos que possam gerar suas próprias *features* de acordo com seu conjunto de treinamento, como é o caso do EDeN.

1.3 Objetivos

Este trabalho tem como objetivo geral a construção de dois métodos computacionais de identificação de snoRNAs usando técnicas de aprendizagem de máquina:

1. Desenvolver o *snoReport 2.0*: extraindo novas *features* de ambas as classes de snoRNAs (*H/ACA box snoRNA* e *C/D box snoRNA*); utilizando uma técnica mais sofisticada na fase de treinamento adotando uma abordagem mais sofisticada para encontrar bons meta-parâmetros da SVM; e usando dados recentes de vertebrados;
2. Desenvolver o snoRNA-EDeN, um método de identificação de snoRNAs usando o EDeN: transformando regiões específicas de estruturas secundárias de *snoRNAs* em uma representação em grafos, que são então transformados em um vetor esparso, que pode ser usado em diferentes algoritmos de aprendizagem de máquina, como o gradiente descendente estocástico (SGD).

Os objetivos específicos relacionados ao desenvolvimento do *snoReport 2.0* são:

1. Incluir novos atributos de ambas as classes de *snoRNAs* no vetor de *features*;
2. Aprimorar a fase de treinamento da SVM;
3. Aprimorar a fase de identificação de candidatos a snoRNAs;
4. Validar o método com organismos previamente identificados na literatura, tanto para organismos vertebrados, quanto para organismos invertebrados, a fim de verificar a capacidade de generalização do método.

Para o desenvolvimento de um método de identificação de snoRNAs usando o EDeN, foram definidos os seguintes objetivos específicos:

1. Identificar características que podem ser utilizadas nos grafos gerados pelo EDeN para identificar *snoRNAs*;
2. Implementar e validar o método com *snoRNAs* de organismos vertebrados e invertebrados;
3. Comparar os resultados obtidos com o novo método com o snoReport 2.0.

1.4 Descrição dos Capítulos

No capítulo 2, são apresentados conceitos básicos de Biologia Molecular com foco em *snoRNAs*. Primeiramente serão apresentados conceitos necessários para o entendimento

do Dogma Central da Biologia Molecular, tais como ácidos nucléicos e proteínas. Após isso, serão descritos conceitos básicos de *ncRNAs* e *snoRNAs*, seguidos de uma revisão de literatura sobre os principais métodos de identificação e classificação de *snoRNAs*, além de bancos de dados de *snoRNAs*. No capítulo 3, são apresentados conceitos gerais de aprendizagem de máquina. Em seguida, será descrito o funcionamento das duas técnicas de aprendizagem de máquina utilizadas neste projeto: SVM e EDeN. No Capítulo 4, é apresentado o método usado no *snoReport 2.0*, detalhando todos os passos necessários para a sua construção. Além disso, são discutidos os resultados obtidos. No Capítulo 5, é apresentado o método de identificação de snoRNAs usando o método EDeN. Por fim,no Capítulo 6, conclui-se esta dissertação, destacando as contribuições e sugerindo trabalhos futuros.

Capítulo 2

snoRNAs

Neste capítulo, são apresentados conhecimentos básicos de Biologia Molecular, em particular de *ncRNAs* e de *snoRNAs*. Na seção 2.1, é descrito o Dogma Central da Biologia Molecular. Na seção 2.2, são apresentados conceitos sobre *ncRNAs*. Já na seção 2.3, são descritas as características biológicas de um *snoRNA* e é feita uma revisão de literatura contendo tanto métodos de identificação quanto bancos de dados de de *snoRNAs*.

2.1 Biologia Molecular e Bioinformática

A Biologia Molecular é a área da Biologia responsável pelos estudos de ácidos nucleicos, estrutura de proteínas, processos relacionados e outros atores envolvidos, tais como organelas celulares e enzimas [11]. Para dar suporte aos estudos referentes à Biologia Molecular, devido a enorme quantidade de dados produzidos por sequenciadores automáticos que devem ser analisados, surgiu a Bioinformática, uma nova área que, de acordo com o *Nacional Institute of Health* (NIH), pode ser definida como a "pesquisa, desenvolvimento ou aplicação de ferramentas computacionais e abordagens para expandir o uso de dados biológicos, médicos, comportamentais ou de saúde, incluindo adquirir, armazenar, organizar, analisar ou visualizar tais dados" [69].

Nesta seção, são apresentados conceitos básicos de Biologia Molecular: ácidos nucleicos, proteínas, e por fim o Dogma Central da Biologia Molecular.

2.1.1 Ácidos nucleicos

Os ácidos nucleicos têm a função principal de armazenar informação necessária e prover mecanismos para a criação de proteínas, possibilitando também a transferência dessas informações para outros organismos, através de processos de reprodução celular [78].

Na natureza, encontramos dois tipos de ácidos nucleicos, o DNA (ácido desoxirribonucleico) e o RNA (ácido ribonucleico). Essas macromoléculas são formadas por monômeros[1] chamados nucleotídeos. Um nucleotídeo é formado por um açúcar composto por 5 átomos de carbono (pentose), que se liga a um grupo fosfato e a uma base nitrogenada. A diferença encontrada entre a molécula de DNA e de RNA no nível estrutural é a de que o RNA possui como pentose a ribose e o DNA possui a desoxirribose. A figura 2.1 mostra a diferença entre a pentose encontrada no DNA (desoxirribose) e a pentose ligada ao RNA (ribose), que consiste na presença ou ausência de uma hidroxila (OH) no carbono 2'.

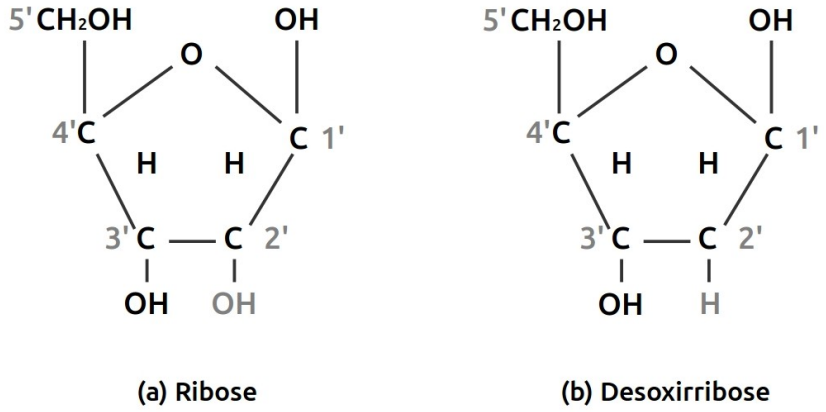


Figura 2.1: Diferença entre as pentoses presentes no RNA e no DNA, respectivamente. (a) A ribose possui uma hidroxila (OH) ligada ao carbono 2'. (b) A desoxirribose possui um átomo de hidrogênio (H) ligado ao carbono.

As pentoses do DNA e do RNA podem se ligar a quatro diferentes tipos de bases nitrogenadas que são: A - Adenina, C - Citosina, G - Guanina e T - Timina, caso seja DNA, ou U - Uracila, caso se trate de um RNA, no lugar da Timina. O DNA é formado por uma dupla fita, disposta espacialmente em formato helicoidal, enquanto o RNA é formado apenas por uma fita.

A ligação entre os diversos nucleotídeos para a formação de algum ácido nucleico é feita através dos grupos fosfatos, por meio de uma ligação chamada fosfodiéster [78]. Essa ligação possibilita a ligação do carbono 3' de um nucleotídeo a um grupo fosfato, que por sua vez se liga ao carbono 5' de outro nucleotídeo. É por esta razão que convencionalmente os ácidos nucleicos são sintetizados na direção canônica 5' $\rightarrow$ 3'.

Como dito anteriormente, moléculas de DNA são formadas por duas fitas complementares na forma helicoidal, onde uma fita com sentido 5' $\rightarrow$ 3' (fita codificadora) liga-se a uma fita no sentido 3' $\rightarrow$ 5' (fita molde), como se pode ver na figura 2.2. Essas fitas são unidas através de ligação de pontes de hidrogênio devido a complementaridade par a par

[1] monômeros são pequenas moléculas formadoras de moléculas maiores

entre duas bases nitrogenadas. A base A é dita complementar à T (e vice versa) e a base C complementar à G (e vice versa), isto é, esses pares de bases complementares ligam-se através de pontes de hidrogênio, fazendo assim a ligação entre as duas fitas de DNA.

Estes pares são conhecidos como *Watson-Crick base pairs* [78] ou simplesmente pares de bases. Pares de bases provêm uma unidade de medida amplamente utilizada para descrever o tamanho de uma molécula de DNA ou RNA e pode ser abreviada para *pb* [78].

Quanto ao nível funcional destas moléculas, o DNA possui como função principal o armazenamento de informações necessárias para a síntese de proteínas ou de *ncRNAs* em um organismo. Essas informações que codificam transcritos estão contidas em regiões do DNA denominadas genes. Existem genes codificadores de proteína e aqueles não codificadores de proteína, chamados de ncRNAs. Já o RNA possui diversas funções em um organismo tais como a constituição do ribossomo (rRNA), o transporte de aminoácido (tRNA), o transporte de informações para a síntese de proteína (mRNA), e diversos papéis em processos de regulação gênica [78].

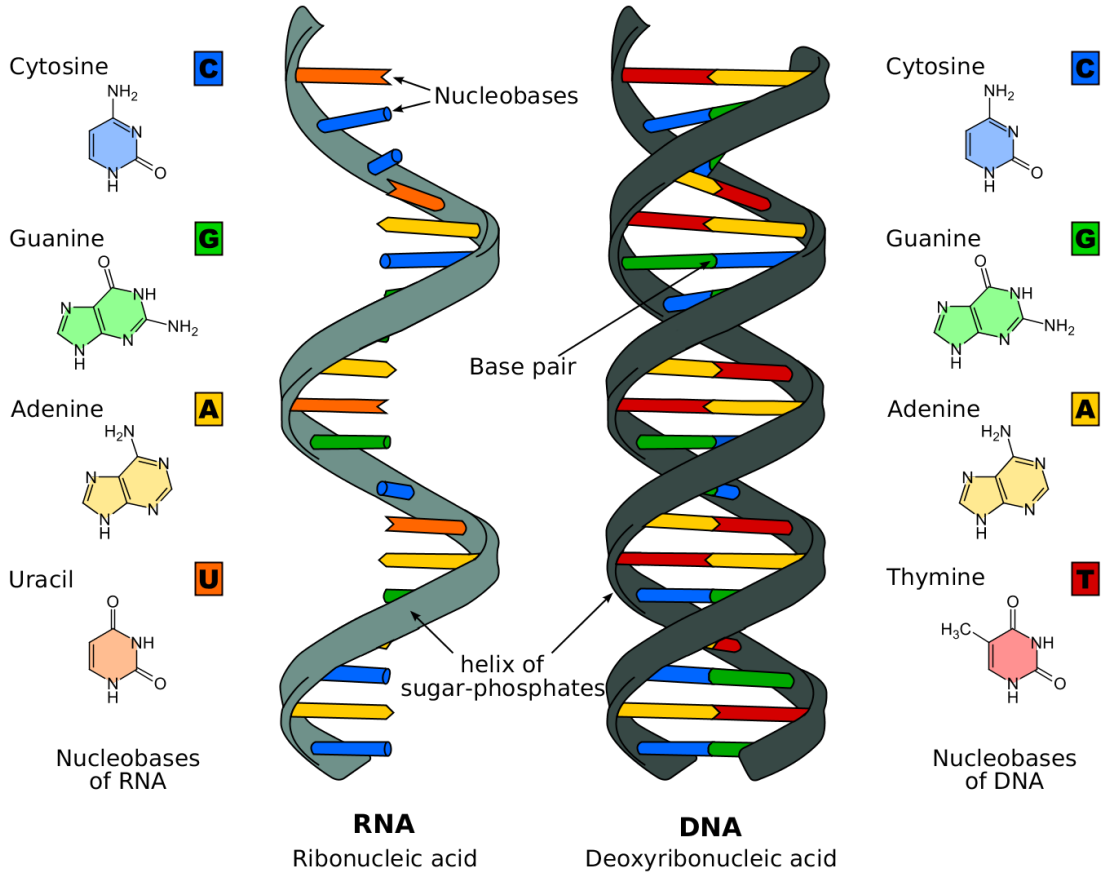


Figura 2.2: Estrutura espacial das moléculas de RNA e DNA [92].

2.1.2 Proteínas

Proteínas são macromoléculas formadas por uma cadeia de moléculas denominadas aminoácidos e possuem diversas funções vitais para os seres vivos, tais como acelerar reações químicas, transporte de nutrientes, eliminação resíduos tóxicos e também na construção estruturas complexas [78]. As enzimas, por exemplo, são proteínas de grande importância para os seres vivos, pois aceleram a ocorrência de várias reações químicas, as quais poderiam demorar muito tempo ou até nunca se completar, dificultando a manutenção da vida.

Um aminoácido é formado por um carbono central, também chamado de carbono alfa (C_α), ligado a um grupo amina (NH_2), a um grupo carboxila (COOH), a um átomo de hidrogênio (H) e a uma cadeia lateral R (figura 2.3). É essa cadeia lateral que distingue um aminoácido de outro. Um cadeia lateral pode ser tão simples como um átomo de hidrogênio (é o caso da glicina), ou tão complicado como dois anéis de carbono (é o caso do triptofano) [78]. Existem cerca de 20 diferentes tipos de aminoácidos, que são listados na tabela 2.1.

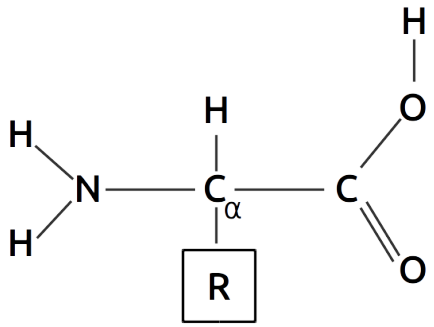


Figura 2.3: Estrutura química de um aminoácido, onde R representa a cadeia lateral.

Para a formação de uma proteína, os aminoácidos são ligados através da chamada ligação peptídica. Na ligação peptídica, o átomo de carbono pertencendo ao grupo carboxila de um aminoácido liga-se ao átomo de nitrogênio de outro aminoácido, liberando neste processo uma molécula de água (H_2O). Por isso podemos dizer que uma proteína é uma cadeia de polipeptídeos formadas não por aminoácidos, mas por resíduos de aminoácidos resultantes de ligações peptídicas. Uma sequência linear de resíduos de aminoácidos que formam uma proteína é denominada estrutura primária.

Forças moleculares são exercidas entre aminoácidos próximos dando a proteína uma estrutura espacial bem definida. Podemos classificar as seguintes formas espaciais de uma molécula da seguinte maneira: (i) Estrutura primária, que é a sequência de aminoácidos; (ii) Estrutura secundária, na qual interações entre aminoácidos próximos resultam em uma estrutura local, por exemplo em forma de hélice; (iii) Estrutura terciária, resultante da

Tabela 2.1: Código, nome e abreviação dos 20 diferentes tipos de aminoácidos.

Código	Nome	Abreviação
A	Ala	Alanina
C	Cys	Cisteína
D	Asp	Ácido Aspártico
E	Glu	Ácido Glutâmico
F	Phe	Fenilanina
G	Gly	Glicina
H	His	Histidina
I	Ile	Isoleucina
K	Lys	Lisina
L	Leu	Leucina
M	Met	Metionina
N	Asn	Asparagina
P	Pro	Prolina
Q	Gln	Glutamina
R	Arg	Arginina
S	Ser	Serina
T	Thr	Treonina
V	Val	Valina
W	Trp	Triptofano
Y	Tyr	Tirosina

interação entre aminoácidos distantes fisicamente; e por fim (iv) a Estrutura quaternária, onde a forma espacial da molécula ocorre pela interação entre forças vindas de diferentes proteínas próximas entre si. A figura 2.4 exemplifica a classificação estrutural de uma proteína.

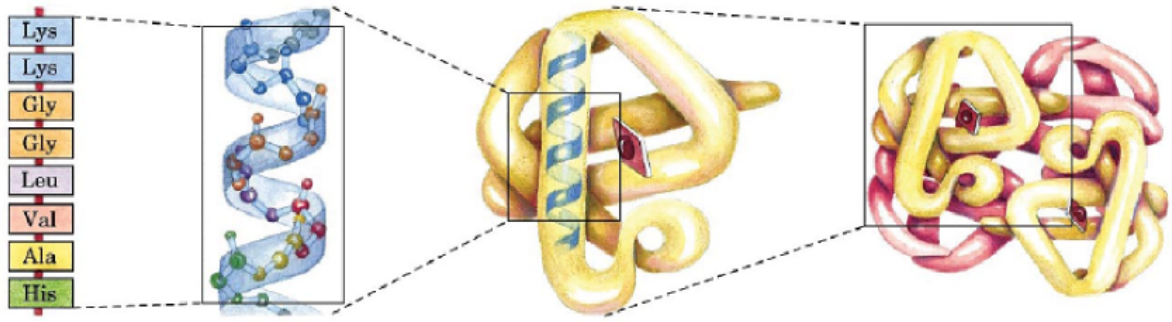


Figura 2.4: Estruturas primária, secundária, terciária e quaternária de proteínas [41].

A determinação da estrutura tridimensional de uma proteína é uma importante área de pesquisa em Biologia Molecular devido a três razões [78]. A primeira razão é a de que a estrutura tridimensional de uma proteína está intimamente relacionada com sua função, por exemplo os anticorpos do sistema imunológico humano reconhecem um antígeno por terem uma superfície complementar ao desse antígeno [11]. A segunda razão se dá pelo fato das proteínas serem formadas por 20 tipos diferentes de aminoácidos e, portanto, poderem assumir formatos bastante complexos e não simétricos. Por fim, a terceira razão é a de que nenhum método simples e preciso é conhecido para determinar a estrutura tridimensional de uma proteína.

2.1.3 Dogma Central da Biologia Molecular

O Dogma Central da Biologia Molecular (figura 2.5) refere-se a três processos: replicação, onde uma molécula de DNA é duplicada; transcrição, onde uma porção da fita de DNA traz informações que permitem gerar uma fita de RNA; e tradução, onde o RNA gerado na transcrição será utilizado como molde para a síntese de uma proteína.

A replicação inicia-se com a separação das fitas de DNA. A RNA polimerase[2] identificará na fita molde (3'→5') uma região codificadora de proteína, denominada gene. A RNA polimerase reconhece essa região, que é normalmente precedida por uma sequência característica, denominada promotora, que geralmente possui uma sequência de TA's (chamada de *TATA box*) [11]. Tendo identificado a região promotora, a RNA polimerase irá conduzir a transcrição do DNA em um RNA mensageiro (*mRNA*) ou em um RNA

[2]RNA polimerase é uma enzima responsável pela catalização da transcrição do DNA

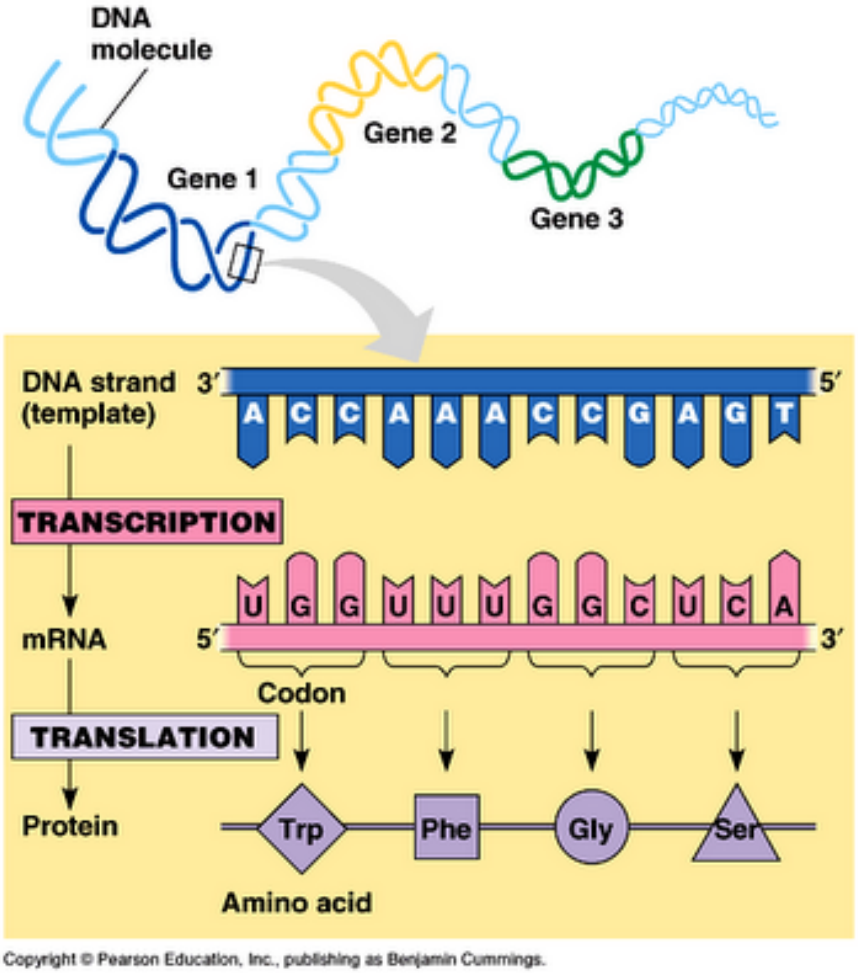


Figura 2.5: Dogma Central da Biologia Molecular [16].

transcrito. A transcrição ocorre no sentido 5'→3' onde os nucleotídeos A, T, C, G são traduzidos para T, U, G, C, respectivamente.

Neste ponto, vale ressaltar que, caso o organismo seja procarioto (como é o caso das bactérias), o RNA mensageiro produzido já estará pronto para ser utilizado na tradução. Caso seja um organismo eucarioto (como é o caso dos seres humanos, por exemplo), haverá um processo denominado *splicing*. Basicamente no processo de *splicing* irá ocorrer a remoção de algumas regiões gênicas, chamadas de introns, e a ligação das regiões codificadoras, denominadas exons.

Os introns são removidos em forma de laços, que são abertos e subsequentemente degradados, porém *ncRNAs* denominados *snoRNAs*, que atuam na modificação de *rRNAs, tRNAs* e *snRNAs*, escapam desta degradação e formam um complexo protéico [25]. Existe também o *splicing alternativo*, uma combinação entre exons não contíguos, que pode gerar uma diversidade de possíveis transcritos a partir de um mesmo gene.

Após o processo de *splicing*, seja ele alternativo ou não, podem ser inseridos ou removidos alguns nucleotídeos, fenômeno conhecido como *RNA editing* [11]. Esse procedimento pode ocorrer pois a sequência de nucleotídeos a ser produzida deve corresponder a um determinado padrão, a fim de produzir uma proteína, posteriormente.

Terminada a transcrição, é iniciado o processo de tradução, onde o *mRNA* é sintetizado em uma proteína. Neste ponto, é necessário relembrar que uma proteína é formada basicamente por aminoácidos, e tais aminoácidos serão transportados ao rRNA pelo RNA transportador (*tRNA*), que pode ser visto na figura 2.6. Em uma extremidade de um *tRNA* há um aminoácido e na outra extremidade há uma sequência de três nucleotídeos, denominado anticódon. Este anticódon será ligado a outra trinca denominada códon existente no *mRNA*. A correspondência entre um aminoácido presente em um *tRNA* e um códon é chamado de código genético, apresentado na figura 2.7.

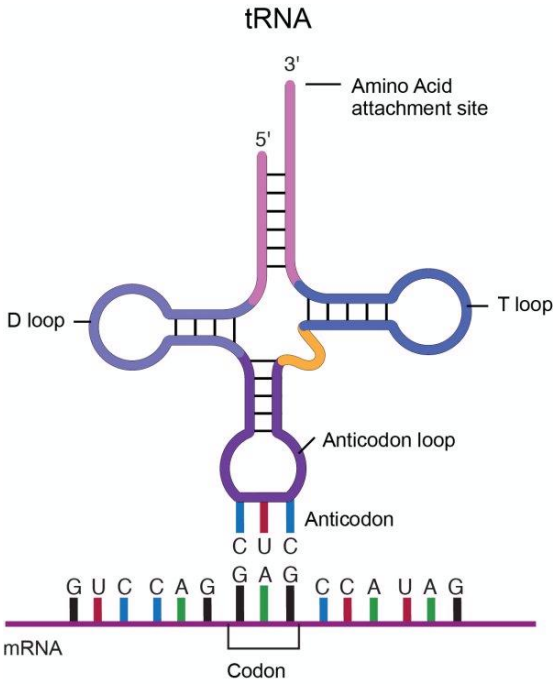


Figura 2.6: Estrutura molecular de um RNA transportador [48].

A síntese do *mRNA* ligado a *tRNAs* ocorre nos ribossomos, que são complexos citoplasmáticos constituídos de RNAs ribossomais (*rRNAs*) e proteínas. Os ribossomos funcionam como uma linha de montagem de uma fábrica, usando como entradas o mRNA e o tRNA e como saída uma cadeia linear de uma proteína [78].

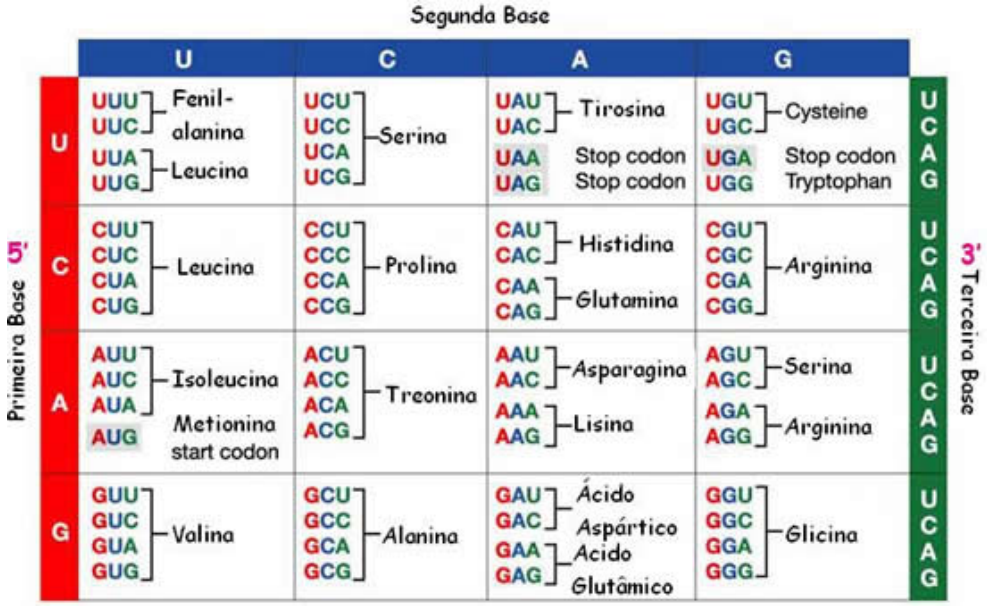


Figura 2.7: Código genético mapeando códons para aminoácidos [1].

2.2 ncRNAs

Projetos genoma iniciados no século XX, tal como o Projeto Genoma Humano [47, 85], tinham como objetivo identificar sistematicamente genes [21]. Contudo os métodos utilizados para essa identificação não eram capazes de identificar todas as classes de genes, em particular genes não-codificadores de proteínas, ou *non-coding RNAs* (*ncRNA*) *genes*. Os *ncRNAs* tem funções estruturais, catalíticas ou regulatórias, e não produzem *mRNAs*, que geram proteínas.

Por muito tempo, regiões não codificadoras de proteína eram denominadas de *DNA lixo*, por não haver uma razão particular para estarem lá, a não ser para proteger os genes. Contudo, várias pesquisas nas últimas décadas vêm mostrando que estas regiões desempenham papéis importantíssimos nos organismos [78, 25, 34, 81]. Sabe-se hoje que existe uma relação intrínseca entre a quantidade de material não codificador no DNA e a complexidade de um organismo, como pode ser evidenciado pela comparação da porcentagem de sequências que codificam proteínas entre diferentes organismos. Por exemplo, bactérias, organismos eucariotos unicelulares, invertebrados e mamíferos possuem respectivamente em média 95%, 30%, 20% e 2% de material codificador de proteínas [81].

RNAs não-codificadores que atuam em atividades regulatórias têm sido identificados em todos os domínios de vida e estão envolvidos em numerosos mecanismos de controle de expressão gênica, em todos os níveis de transmissão da informação genética do DNA para a criação de uma proteína [81].

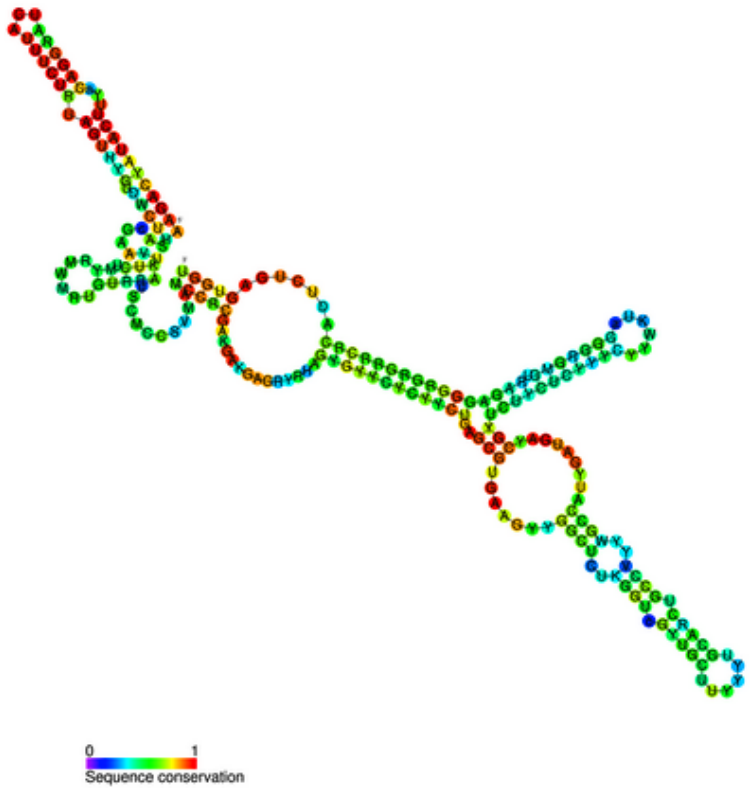


Figura 2.8: Estrutura secundária do U3 snRNA [26].

Porém, o estudo de *ncRNAs* vem se mostrando bastante desafiador, devido a dificuldade de se verificar experimentalmente qual função desempenha determinado gene não-codificador de proteínas em um organismo [58]. Métodos computacionais para a identificação de ncRNAs possuem dificuldades similares aos métodos experimentais, onde a Bioinformática não possui um método único para a identificação e classificação de ncRNAs. Isso se dá principalmente devido a essas moléculas possuirem uma alta conservação de estrutura secundária e não de estrutura primária, inviabilizando o uso de métodos que utilizam similaridade de estrutura primária, como é o caso dos métodos para identificação de proteínas [58]. A figura 2.8 mostra um exemplo de estrutura secundária de um U3 snoRNA, uma classe específica de *ncRNA*.

2.2.1 Classificação de ncRNAs

A classificação de *ncRNAs* em diferentes famílias é feita normalmente pela função que uma molécula de RNA exerce em determinado organismo [58]. A função de um ncRNA está intimamente ligada à sua estrutura terciária. Como a estrutura terciária é determinada pela estrutura secundária, esta última é usada como uma aproximação no estudo de funções em *ncRNAs* [58]

Segundo Lima [57], as famílias de *ncRNAs* conhecidas até os anos 80 eram apenas as de *tRNAs* (responsáveis pelo transporte de um aminoácido usado na síntese de proteína) e as de *rRNA* (RNAs ribossomais responsáveis pela catálise de síntese protéica [12]). Atualmente o número de famílias conhecidas chega a 2.450, segundo dados disponíveis no banco de dados Rfam v12.0 [9]. Algumas das principais famílias de ncRNAs são descritas na tabela 2.2.

Tabela 2.2: Algumas famílias importantes de *ncRNAs* [57, 20, 70, 60].

Sigla	Nome	Função
tRNA	RNA transportador	Transporte de aminoácidos para a síntese de proteínas
rRNA	RNA ribossomal	Catálise de síntese de proteínas
snoRNA	*Small nucleolar RNA*	Modificações nos rRNAs, tRNAs e snRNAs
snRNA	*Small nuclear RNA*	Realização de excisão dos introns no processo de *splicing*
siRNA	*Small interfering RNA*	Interferência na tradução de proteínas, separando e promovendo a degradação de trechos de mRNAs
rasiRNA	*Repeat-associated siRNA*	Silenciamento da transcrição de genes via remodelagem da cromatina
snmRNA	*Small non-messenger RNA*	Pequenos *ncRNAs* com função regulatória
miRNA	*MicroRNA*	Família de genes reguladores da tradução
piRNA	*Piwi-interacting RNA*	Regulação de tradução e estabilidade de mRNA, dentre outras funções
stRNA	*Small temporal RNA*	Interrompção da tradução de mRNA
lncRNA	*ncRNAs* longos	Diversas funcionalidades, das quais muitas ainda são desconhecidas, como a regulação da expressão gênica a nível de remodelagem de cromatina (possuem mais de 200 nucleotídeos)

2.2.2 Predição de estrutura secundária de RNAs

A estrutura secundária é um importante meio de descrição de *ncRNAs*, pois frequentemente são bem conservadas na evolução. Por isso, a estrutura secundária predita é utilizada como um meio de se classificar um *ncRNA* em uma determinada família, sendo cruciais no estudo da função que um determinado *ncRNA* desempenha em um organismo [54, 58].

Formalmente uma estrutura secundária de RNA pode ser definida da seguinte forma:

Definição 2.2.1 *Seja* $x = x_1x_2...x_n$ *uma sequência de RNA, onde* $x_i \in \{A, C, G, U\}$ *para* $i = 1, ..., n$. *Uma estrutura secundária de* x *é um conjunto de pares de bases* $P = \{(i, j)|i < j\}$ *com as seguintes regras:*

1. *Se* $(i, j) \in P$ *então* $(x_i, x_j) \in \{(G, C), (C, G), (A, U), (U, A), (G, U), (U, G)\}$
2. *Se* $(i, j) \in P$ *e* $(i, l) \in P$, *então* $j = l$
3. *Se* $(i, j) \in P$ *e* $(k, j) \in P$, *então* $i = k$
4. *Se* $(i, j) \in P$ *então* $j - i < \theta$
5. *Se* $(i, j) \in P$ *e* $(k, l) \in P$ *e* $i < k < j$, *então* $i < k < l < j$

A partir dessas regras, são construídos diferentes componentes estruturais (figura 2.9):

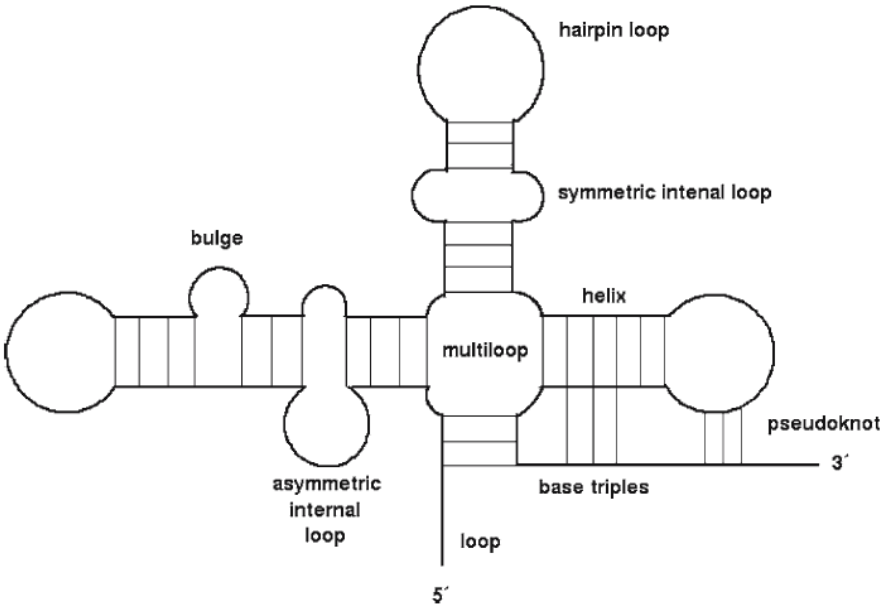


Figura 2.9: Principais componentes estruturais de um RNA [58].

- *Helix* ou *stem*: um empilhamento contíguo de pares de bases (por exemplo, (i, j) e $(i + 1, j + 1)$);
- *Loop*: uma região de bases não pareadas;
- *Hairpin-loop*: um *loop* delimitado por uma *stem*;
- *Multi-loop*: uma região de *loop* onde três ou mais *stems* são formados;
- *internal-loop*: um *loop* dentro de uma *stem*, podendo ser assimétrico, caso o número de nucleotídeos em cada lado da *stem* seja diferente, ou simétrico, caso contrário;
- *Bulge*: um *loop* dentro da *stem*, mas em apenas um lado dela.

Além destes componentes estruturais, alguns pareamentos podem ocorrer de duas estruturas diferentes: os *pseudoknots* e as *base triples*. Estas interações são consideradas parte da estrutura terciárias [58].

A tarefa de predizer a estrutura secundária a partir de uma sequência de RNA é bastante trabalhosa, visto que o número possível de estruturas secundárias cresce exponencialmente com o tamanho da sequência [58, 89]. Quando a estrutura secundária de uma única sequência de RNA necessita ser predita, apenas métodos *ab initio* podem ser utilizados. Caso um conjunto de RNAs homólogos esteja disponível, métodos comparativos como o *Infernal* [64] podem ser utilizados para predizer a estrutura consenso com mais acurácia [58].

2.2.3 Métodos para a identificação de *ncRNAs*

Devido à grande importância biológica dos *ncRNAs* e a grande dificuldade de identificação e classificação, diversos métodos computacionais foram e estão sendo desenvolvidos para a identificação e classificação de *ncRNAs*. Contudo, como já citado anteriormente, a função de um ncRNA está intimamente ligada à sua estrutura secundária e não à sua estrutura primária [51], dificultando a predição deste tipo de molécula com métodos normalmente projetados para identificação de genes codificadores de proteínas [57].

A tarefa de percorrer um genoma com a finalidade de detectar *ncRNAs* é um desafio em aberto na Bioinformática [58]. Estratégias normalmente utilizadas para solucionar este problema são a de identificar atributos e características específicas de famílias de *ncRNAs* [20], ou a de criar programas que podem ser treinados com objetivo de identificar características intrínsecas de uma família específica, a partir de uma sequência de entrada [58]. Essas estratégias normalmente seguem processos *ab initio*, isto é, não utilizam outros *ncRNAs* conhecidos a fim de identificar novos, ou métodos comparativos, onde a partir de um banco de dados contendo diversas amostras de *ncRNAs*, são anotados novos *ncRNAs*, classificados em determinada família de *ncRNAs*. Outra estratégia usada na identificação de *ncRNAs* em genomas se dá na busca de determinadas regiões do DNA ricas em determinada composição de nucleotídeos. Utilizando métodos estatísticos, é possível observar que *ncRNAs* possuem em geral uma média de conteúdo G+C de 50% [73]. Por exemplo, no genoma humano, que possui cerca de 42% de conteúdo GC, *miRNAs* e *snoRNAs* H/ACA box possuem 50% em média de conteúdo GC [88]. Outro fato interessante é o de que em organismos com genomas ricos em dinucleotídeos AT, a busca de *ncRNAs* em regiões ricas em GC é bastante satisfatória [46]. No snoReport [34] o conteúdo GC é utilizado como atributo para a identificação de *snoRNA*.

A seguir são apresentados três abordagens comumente utilizadas para a construção de métodos para a identificação de *ncRNAs*.

Abordagem termodinâmica

Métodos baseados em preditores termodinâmicos exploram a hipótese de que um RNA é dobrado em sua estrutura secundária mais termodinamicamente estável, ou seja, a estrutura secundária que possui menor valor de energia livre mínima (MFE) [58]. Uma abordagem direta seria a de listar todas as possíveis estruturas e então selecionar aquela com a menor MFE [71], contudo devido a complexidade de tempo exponencial, esta abordagem se torna impraticável em sequências muito grandes.

Para lidar com esse problema, os métodos atuais utilizam um método de programação dinâmica primeiramente proposto por Nussinov et al. [66] que reduz a complexidade para $O(n^3)$. Entretanto a estrutura secundária com menor MFE pode não ser a que realmente foi adotada pelo RNA. A estrutura correta pode estar entre estruturas com MFE sub-ótimas. Neste caso, a análise do espaço das possíveis estruturas pode dar pistas sobre a estrutura mais provável que um determinado RNA pode assumir [58].

O *Vienna RNA Package* [54] é um pacote de programas e bibliotecas de linguagem C para a predição e comparação de estruturas secundárias de RNAs que utiliza preditores termodinâmicos. O seu principal programa para predição de estrutura secundária é o *RNAfold*, que analisa as estruturas sub-ótimas com a finalidade de criar um modelo mais acurado de qual tipo de estrutura um determinado tipo de RNA está propenso a possuir [58]. Além disso, caso seja conhecida previamente a estrutura secundária esperada de uma sequência de RNA em estudo, é possível atribuir restrições aos nucleotídeos de forma a facilitar a predição correta de sua estrutura secundária.

Abordagem por homologia

A predição de *ncRNAs* é realizada através da comparação de genomas entre duas ou mais espécies. Essas comparações necessitam de bancos de dados curados, pois o quão melhores forem as anotações no banco, melhores serão as predições [58]. Dois genes são ditos homólogos se descendem de um ancestral comum, e possivelmente esses genes possuirão a mesma funcionalidade herdada [3].

O *Infernal ("INFErence of RNA ALignment")* [64] é uma ferramenta baseada em homologia de RNAs. O*Infernal* constrói perfis probabilísticos das sequências e das estruturas secundárias de uma família de RNAs (também chamados modelos de covariância) através de alinhamentos de estruturas secundárias de RNAs, ou a partir de estruturas primárias de sequências. A partir desses modelos de covariância, o *Infernal* possui duas aplicações principais: classificar RNAs em um conjunto de sequências (por exemplo, realizar anotação de RNAs em um genoma) e de criar alinhamentos múltiplos e alinhamentos baseados em estruturas de RNAs homólogos.

Abordagem utilizando Aprendizagem de Máquina

A Aprendizagem de Máquina vem se mostrando bastante eficaz na construção de programas para identificar *ncRNAs* [58]. Essa abordagem é normalmente utilizada construindo um classificador, a partir de um conjunto de treinamento, para identificar determinadas famílias de *ncRNA*. Este conjunto de treinamento normalmente é construído usando atributos de transcritos de ncRNA e mRNA, por exemplo tamanho da ORF, composição de nucleotídeos, estrutura secundária, dentre outros [20], extraídos de diversas amostras disponíveis nos bancos de dados de *ncRNAs*.

2.2.4 Bancos de dados de *ncRNAs*

Várias classes diferentes de ncRNAs vêm sendo descobertas recentemente [9], implicando no aumento da quantidade de dados sobre essas moléculas. Os bancos de dados de *ncRNAs* têm como objetivo organizar informações relevantes sobre os diversos tipos de *ncRNAs* existentes [81], sendo utilizados tanto para pesquisas por biólogos quanto para identificar novas sequências de ncRNAs em métodos computacionais como aprendizagem de máquina. Em seguida, listamos e comentamos brevemente bancos de dados de *ncRNA*.

ncRNAdb [81] (*noncoding RNA database*) tem por objetivo prover informação de sequências e funções dos *ncRNAs*. Atualmente esse banco de dados inclui sequências de *ncRNAs* provenientes de 99 espécies de bactérias, Archaea e eucariotos.

NONCODE [52] é um banco de dados de *ncRNAs* extraídos automaticamente da literatura e do GenBank [7], os quais foram manualmente curados. O NONCODE possui quase todos os tipos de *ncRNAs* com exceção de *tRNAs* e de *rRNAs* e todas as suas sequências e informações relacionadas (como função e localização celular, dentre outros) foram confirmadas manualmente. Mais de 80% de suas entradas são baseadas em dados experimentais.

RFAM [9] categoriza sequências primárias de *ncRNA* e estruturas secundárias, através do uso de alinhamento múltiplo, consenso de anotações de estruturas secundárias e modelos de covariância. Na sua primeira publicação, o RFAM v1.0 possuía 25 famílias de *ncRNAs* e na sua atual versão v12.0 possui 2.208 famílias.

RNAdb [67] é um banco de dados de *ncRNAs* de mamíferos contendo sequências de nucleotídeos e anotações de dezenas de milhares de *no-housekeeping ncRNAs*, incluindo uma variedade de microRNAs, snoRNAs e *ncRNAs* longos de tamanho similar ao de um mRNA.

fRNAdb [44] possui uma grande coleção de transcritos não codificadores de proteínas, incluindo sequências anotadas e não anotadas de outros bancos, tais como NONCODE e RNAdb.

miRbase [28] é um repositório de sequências, anotação e predições de microRNAs. Atualmente na sua décima versão, ele conta com 5.071 *loci* de *miRNAs* provenientes de 58 espécies, expressando 5.922 miRNAs maduros distintos e, provendo dessa forma, muitas informações importantes para estudos sobre *miRNA*.

2.3 Bioinformática de snoRNAs

Como discutido anteriormente, a maior parte do genoma de organismos mais complexos é transcrito em *ncRNAs* [81]. Alguns *ncRNAs* recém-gerados na transcrição sofrem um processamento posterior, gerando RNAs menores e metabolicamente estáveis possuindo diversas funções, como os *rRNAs, tRNAs* e *snRNAs*. Os *ncRNAs* gerados por esse pós-processamento são então modificados por um outro *ncRNA* denominado *snoRNA* [25].

Os *snoRNAs* são RNAs que possuem de 60 a 300 nucleotídeos que se acumulam no nucléolo. Os snoRNAs são classificados de acordo com determinadas características presentes em sua sequência secundária e possuem duas classes principais: C/D box *snoRNAs* e H/ACA box *snoRNAs*. Em humanos, estes RNAs são normalmente encontrados em regiões intrônicas. Após o processo de *splicing*, os introns são removidos em forma de laços que são abertos e subsequentemente degradados. Contudo, os *snoRNAs* escapam desta degradação participando de um complexo protéico [25].

Os C/D box *snoRNAs* (figura 2.10) são caracterizados pela presença de dois motivos conservados, o box C (RUGAUGA) e o box D (CUGA) encontrados perto das extremidades das fitas 5' e 3' da molécula, respectivamente. Um segundo par de boxes, C' e D', podem ser normalmente encontrados próximos do meio de um C/D box *snoRNA*, mas mostra baixa conservação de sequência quando comparados aos boxes C e D. A região guia (também chamada de box antisenso), trecho do C/D box *snoRNA* complementar a algum RNA alvo, é localizada no sentido 5'- 3' imediatamente antes do box D ou D' [77].

Muitos estudos têm mostrado que os *C/D box snoRNAs* devem possuir uma estrutura *kink turn* perfeita, a qual é formada pela interação entre os boxes C e D [5, 90, 93]. Algumas características da estrutura *kink-turn* em *C/D box snoRNAs* são: a interação de dinucleotídeos G•A entre os boxes C (RU**GA**UGA) e D (CU**GA**); ao menos uma uridina no par U-U (RUGA**U**GA e C**U**GA); e um par de bases *Watson-Crick* entre o sexto nucleotídeo do box C e o primeiro nucleotídeo do box D (RUGAU**G**A and **C**UGA) [5]. A figura 4.10 mostra a estrutura *kink-turn* do *C/D box snoRNA*.

Já os H/ACA box *snoRNAs* (figura 4.8) são caracterizados por duas estruturas imperfeitas de *hairpins* separados por uma fita contendo o box H (ANANNA) e seguida por uma pequena sequência contendo o motivo ACA, localizado 3 nucleotídeos antes do fim da fita 3' [49].

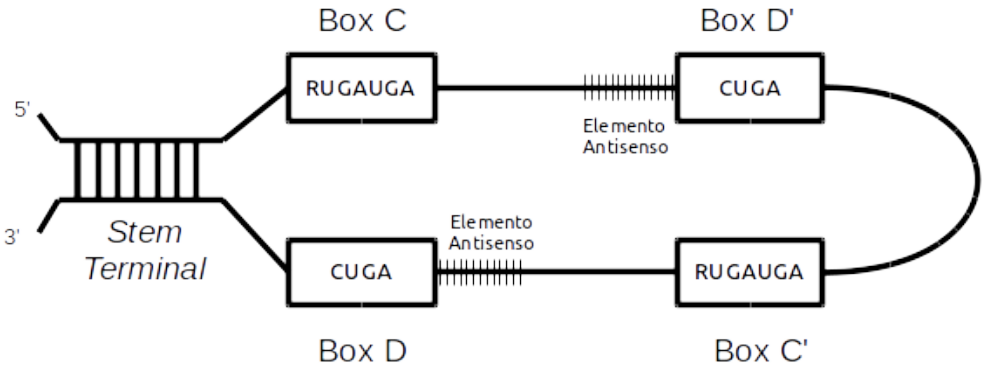


Figura 2.10: Estrutura secundária canônica de um *C/D box snoRNA*. [95]. Os boxes C e D são unidos por um pequeno talo (*stem*) terminal (4-5 pb) e toda região entre esses dois boxes permanecem não pareadas.

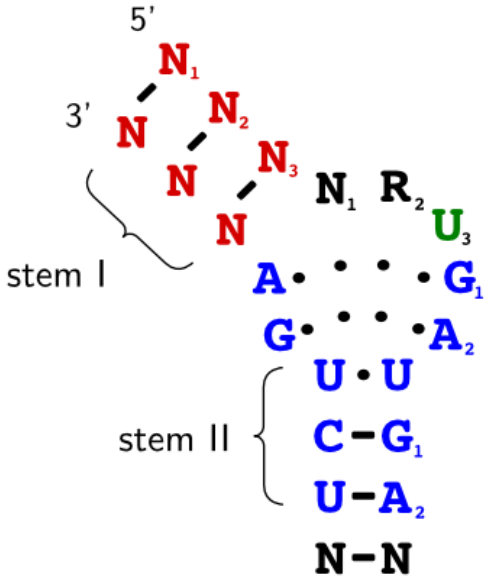


Figura 2.11: *Kink turn* do *C/D box snoRNA* [5].

Complexos protéicos contendo *snoRNAs*, denominadas *snRNPs (small nuclear ribonucleoprotein particles)*, contêm proteínas com atividades enzimáticas. No caso de *snoRNAs* do tipo C/D box, sua composição possui fibrilarina que promove a *2'-O-metilação* de seu RNA alvo. Já os snoRNAs do tipo H/ACA box, apresentam em sua composição *disquerina* que catalisa a conversão de uridina para pseudouridina [25].

2.3.1 Ferramentas de identificação e classificação de snoRNAs

A seguir serão apresentados cinco métodos utilizados para a identificação e classificação de snoRNAs, *snoReport*, *snoSeeker*, *snoGPS*, *snoScan* e *snoStrip*.

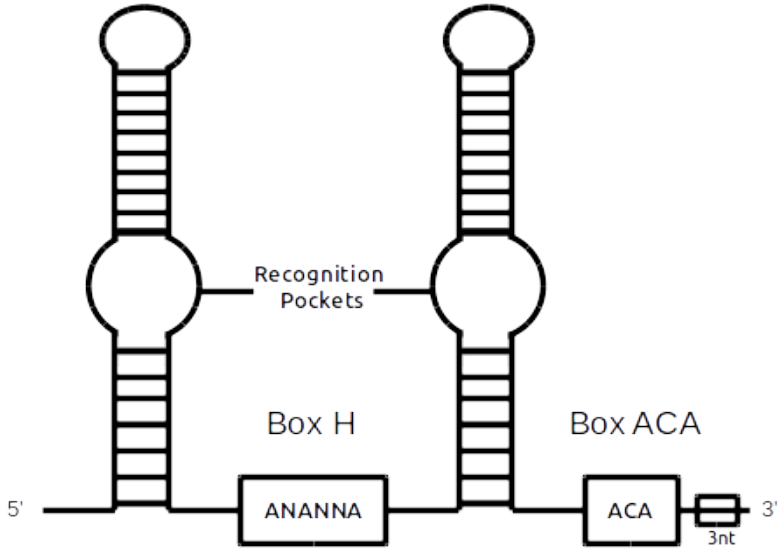


Figura 2.12: Estrutura secundária canônica de um *H/ACA box snoRNA* [95]. A estrutura secundária de um H/ACA box *snoRNA* consiste de dois *hairpins* e duas pequenas regiões de fita simples, contendo o box H e o box ACA. Os *hairpins* contêm *bulges*, ou *loops* de reconhecimento, que formam um complexo de *pseudoknot* com o RNA alvo. Pseudoknot é uma estrutura secundária de ácidos nucleicos contendo ao menos dois *hairpin-loops* (*stem-loops*), onde a metade de um *stem* é intercalada entre as duas metades do outro *stem*.

SnoReport

Um método computacional que vem mostrando bons resultados é o *snoReport* [34], que combina predição de estrutura secundária de *snoRNAs* com Máquina de Vetores de Suporte (SVM), que é capaz de reconhecer *snoRNAs* em sequências individuais sem incluir informação sobre o RNA alvo complementar ao bloco antisenso. Embora essa informação possa melhorar consideravelmente a sensibilidade e especificidade, um número crescente de *snoRNA* órfãos vem sendo descoberto, com a característica da falta de complementaridade do box antisenso com o RNA alvo [39, 38], como o caso de um subgrupo de *snoRNA* expresso no cérebro de mamíferos, que parece não estar envolvido no processo de modificação de *rRNAs* e *snRNAs* [87].

Na figura 2.13, é possível visualizar o *workflow* do *snoReport* utilizado para a identificação das duas principais classes de *snoRNAs*, C/D box *snoRNA* e H/ACA box *snoRNA*. É importante notar que, para cada classe de *snoRNA*, é realizado o mesmo *workflow*, mas com dados distintos e mudanças no vetor de características passado ao classificador SVM.

As sequências do conjunto positivo e negativo para ambas as classes de *snoRNA* (C/D box *snoRNA* e H/ACA box *snoRNA*) são retiradas dos bancos de dados snoRNAbase, Rfam e miRBase. Em seguida, o blastclust [2] é utilizado para retirar sequências redun-

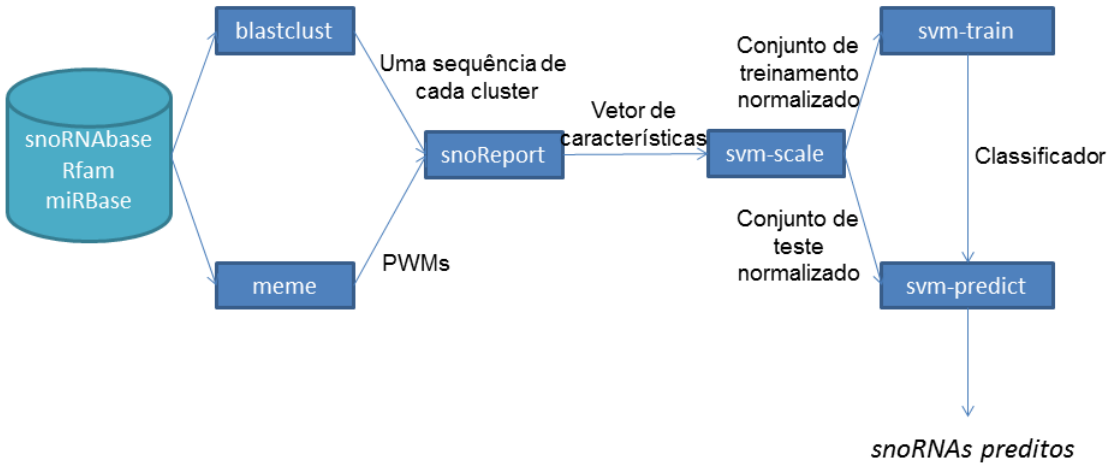


Figura 2.13: *Workflow* usado pelo snoReport.

dantes e, por meio da ferramenta meme [4] são obtidas as Matrizes de Pesos para Posições Específicas (PWMs), utilizadas para identificar os boxes das sequências candidatas. O *snoReport* recebe então as sequências não redundantes e as PWMs e, após realizar a predição da estrutura secundária, retorna um vetor de características para cada sequência computada. Esses vetores de características contêm atributos característicos de snoRNAs e serão utilizados pela SVM. Depois disso, o vetor de características é normalizado pelo programa *svm-scale* e dividido em um conjunto de treinamento e de teste. O programa *svm-train* recebe como entrada o conjunto de treinamento normalizado, e é gerado um classificador utilizado pelo programa *svm-predict* para predizer *snoRNAs* a partir de um conjunto de teste.

O *snoReport* obteve bons resultados. Na fase de testes obteve 96% de sensibilidade e 91% de especificidade para o classificador de C/D box *snoRNA*. Para o classificador de H/ACA box *snoRNA*, obteve 78% de sensibilidade e 89% de especificidade . Em ambos os casos, os valores correspondem ao particionamento de 80% dos dados para teste e 20% para treinamento.

SnoSeeker

Assim como o snoReport, o pacote snoSeeker [95] é capaz de identificar tanto *snoRNAs* guias (aqueles que modificam *rRNAs* e *snRNAs*) quanto *snoRNAs* órfãos (não possuem complementaridade com *rRNAs* e *snRNAs*) [34, 95]. O pacote snoSeeker possui dois programas principais: o CDseeker e o ACAseeker [95], que têm as funções de detectar C/D box *snoRNAs* e H/ACA *snoRNAs*, respectivamente.

Utilizando o snoSeeker, Yang [95] realizou uma varredura no genoma humano em busca de *snoRNAs* utilizando sequências de alinhamentos (*Whole-genome alignment* (WGA)) entre o genoma humano e quatro espécies de mamíferos (humano/rato, humano/camundongo, humano/cachorro e humano/vaca). Com seu método, a maioria dos snoRNAs conhecidos até então da classe C/D box e H/ACA foram detectados.

O programa CDseeker procura o box C, o box D, a haste terminal pareada e o elemento antisenso passo a passo no consenso das sequências WGA, atribui um *score* aos elementos através de modelos probabilísticos e, então, os avalia baseado em um *cutoff score* padrão do conjunto de treinamento. Os candidatos passam para as próximas avaliações apenas se o elemento possuir um *score* maior que o *cutoff score*. O exame da região antisenso é um critério opcional no CDseeker. O programa atribui a um candidato: um *snoRNA* guia; ou um *snoRNA* órfão. Finalmente, para classificar os candidatos, o programa soma os *scores* dos motivos resultantes em um *score* final. O *cutoff score* padrão do conjunto de treinamento também é aplicado para a seleção dos candidados [95].

O programa ACAseeker busca os boxes H e ACA nas sequências conservadas do consenso WGA e atribui uma *pontuação* a eles usando modelos probabilísticos. Os candidatos que possuem uma pontuação maior que um *cutoff score* padrão vão para o próximo passo, que é uma avaliação da estrutura secundária usando um padrão observado de H/ACA conhecidos. Similar ao CDseeker, o exame final da região antisenso no ACAseeker é um critério opcional. O programa, então, atribui aos candidatos um *snoRNA* guia ou *snoRNA* órfão, através dessas avaliações.

Segundo Yang [95], os programas CDseeker e ACAseeker do pacote snoSeeker conseguiram identificar 120 *snoRNAs* órfãos e 200 *snoRNAs* guias do genoma humano. Contudo o snoSeeker, diferentemente do snoReport, identifica apenas *snoRNAs* homólogos, que possam ser alinhados usando por Blast/Multiz [34]. Outra diferença encontrada com relação ao snoReport é a de que o snoSeeker é desenvolvido para identificar candidatos a *snoRNAs* em alinhamentos entre genomas (WGA), enquanto o snoReport é usado para anotar sequências resultantes de outras ferramentas de predição de *ncRNAs* ou para identificar *snoRNAs* em genomas/cromossomos candidatos a *snoRNAs*.

SnoGPS

O *snoGPS* [75] é uma ferramenta utilizada na identificação de H/ACA box *snoRNAs* guias. O seu método é baseado na combinação de um algoritmo determinístico de varredura de sequências genômicas com um modelo probabilístico, que dá um escore para cada candidato a H/ACA box *snoRNA*. Primeiramente, testes determinísticos limitam o espaço de busca, enumerando todas as possíveis características de um dado candidato. Na segunda fase do programa, várias rotinas irão medir o quão similar uma característica

identificada no candidato é de um conjunto de H/ACA box *snoRNAs* conhecidos. Utilizando essas medidas em um modelo probabilístico, será gerado um escore final usado para classificar os candidatos.

O *snoGPS* foi testado na levedura *Saccharomyces cerevisiae* e identificou 6 novos *snoRNAs*, onde 5 deles foram verificados experimentalmente como guias para a formação de pseudouridina em específicas posições no *rRNA*. Além disso, este trabalho foi o primeiro a identificar e verificar experimentalmente *snoRNAs* que guiam modificação de pseudouridina em mais de dois locais de um RNA. Também nesse trabalho, 41 de 44 modificações de pseudouridina nos *rRNAs* do *S. cerevisiae* foram ligados a um *snoRNA* verificado experimentalmente.

SnoScan

O *snoScan* [55] é uma ferramenta utilizada na identificação de C/D box *snoRNAs* guias em leveduras. Primeiramente, o programa faz uma varredura de um candidato a C/D box *snoRNA* guia de 2'O-metilação em uma sequência genômica, utilizando um algoritmo guloso de busca. O programa então identifica seis componentes característicos de C/D box *snoRNA*: os boxes C, D e D'; a região antisenso ao *rRNA*, o *stem* e a posição da metilação predita dentro do *rRNA* guia através da região de complementaridade. Após isso, o candidato recebe um escore de um modelo probabilístico, que será usado para confirmar se o candidato é ou não um C/D box *snoRNA*.

Utilizando o *snoScan* no genoma da levedura *S. cerevisiae*, foi possível identificar 22 C/D box *snoRNAs* guias de metilação. Além disso, 51 de 55 locais de metilação em RNAs foram atribuídos a 41 diferentes C/D box *snoRNAs* guias.

SnoStrip

O *snoStrip* [5] é um *pipeline* para análise de sequências de *snoRNA* em genomas de fungos. Há dois modos de execução para o *snoStrip*: verificar experimentalmente sequências detectadas de *snoRNAs* através de conservação entre uma grande quantidade de genomas de fungos, ou retornar uma anotação completa de famílias de *snoRNAs* para novos genomas de fungos. A ferramenta executa um *pipeline* que realiza várias análises, retornando no final uma série de detalhes. Os passos do *pipeline* são:

1. **Busca de candidatos putativos de *snoRNAs***: Todas as sequências de *snoRNAs* de uma família específica são usadas como sequências de entradas, de forma a buscar ortólogos putativos em certos organismos. Essa busca baseada em homologia utiliza duas ferramentas de bioinformática: *Blast* [2] e *Infernal*;

2. **Identificação de boxes corretos de *snoRNAs***: Nesta fase, o *snoStrip* irá diferenciar os candidatos que possuem boxes conservados de famílias de *snoRNAs* dos falsos positivos;

3. **Extração de propriedades características de *snoRNA***: Nesta fase, o *snoStrip* irá analisar diversas propriedades características de *snoRNAs* nos candidatos à *snoRNAs*;

4. **Predição de regiões alvo (ou região antisenso) putativas de *snoRNAs***: Para encontrar regiões alvo dos candidados a *snoRNAs*, o *snoStrip* utiliza a ferramenta *Plexy* [42] (para *C/D box snoRNAs*) e o *RNAsnoop* [82] (para *H/ACA box snoRNA*). Além disso, nesta fase, é realizada uma busca de regiões alvo entre todos os *snoRNAs* em organismos de uma determinada família;

5. **Alinhamento de famílias de *snoRNAs***: Por fim, o *snoStrip* usa a ferramenta *MUSCLE* [22] para produzir um alinhamento múltiplo de todos os *snoRNAS* com relação a sua respectiva família.

A tabela 2.3 resume as ferramentas de identificação e classificação de snoRNAs.

2.3.2 Bancos de dados de snoRNAs

A seguir serão descritos alguns bancos de dados de *snoRNAs*:

sno/scaRNAbase [94] consiste de um banco de dados que possui cerca de 1.979 sno/scaRNA[3] (snoRNAs e *small Cajal body-specific RNA*) obtidos de 85 organismos.

snoRNA-LBME-db [49], ou *snoRNAbase*, é um banco de dados de *C/D box snoRNAs*, *H/ACA box snoRNAs* e *scaRNAs* de humanos. O banco de dados é composto por *snoRNAs* verificados experimentalmente, *snoRNAs* ortólogos de humanos encontrados em outras espécies de vertebrados e de *snoRNAs* putativos preditos em aplicações de bioinformática, ainda não comprovados experimentalmente.

Plant snoRNA Database [8]: provê informação de *snoRNAs* de *Arabidopsis* e outras 18 espécies de plantas, como a sua sequência, regiões antisenso, RNAs alvo, localização genômica, dentre outros. Na data de sua publicação, o *Plant snoRNA dabase* possuia cerca de 475 sequências de *snoRNAs*.

snoRNP Database [23] é uma coleção de sequências de *snoRNAs* e *snoRNAs* associados a proteínas (*snoRNPs*) de vários organismos. Em sua data de publicação, o *snoRNP database* possuia 8894 sequências de *snoRNAs* de bactérias, Archaea e eucariotos, além de 589 *snoRNPs*

[3] *scaRNAs* são *ncRNAs* que guiam modificações em RNAs transcritos pela RNA polimerase II e são normalmente compostos de domínios de *C/D box* e *H/ACA box snoRNAs*.

Tabela 2.3: Métodos computacionais para identificação e classificação de *snoRNAs*

Ferramenta	Descrição
SnoReport (2008)	Usa combinação entre predição de estrutura secundária e aprendizagem de máquina (SVM) para identificar *C/D box* e *H/ACA box snoRNAs*
SnoSeeker (2006)	Identifica *C/D box* e *H/ACA box snoRNAs* homólogos procurando, em alinhamentos de sequências conservadas de *snoRNAs*, regiões características de *snoRNAs*
SnoGPS (2004)	Usa combinação de um algoritmo determinístico de varredura de sequências genômicas com um modelo probabilístico para encontrar *H/ACA box snoRNAs*
SnoScan (1999)	Faz uma varredura em uma sequência genômica buscando *C/D box* snoRNAs guias de 2'O metilação e, após identificar um candidato, algumas características dessas sequências são submetidas a um modelo probabilistico, responsável por classificar se esse candidato é um *snoRNA*
SnoStrip (2014)	É um *pipeline* para análise de sequências de *snoRNAs* em genomas de fungos

snOPY [56] é um banco de dados que provê informações de *snoRNAs*, localização genômica de *snoRNAs* e RNAs alvo de diversos organimsos. Também contém sequências de ortólogos de vários organismos, permitindo ao usuário analizar a evolução dos *snoRNAs*. No total, possui 13.770 sequências de *snoRNAs*, 10.345 localizações genômicas e 133 RNAs alvos.

A tabela 2.4 sumariza os bancos de dados acima descritos.

Tabela 2.4: Bancos de dados de *snoRNAs*

Banco de dados	Descrição
sno/scaRNAbase (2007)	Banco de dados que possui 1.979 (*snoRNAs* e *scaRNAs*) obtidos de 85 organismos
snoRNA-LBME-db (2006)	Banco de dados de *C/D box snoRNAs*, *H/ACA box snoRNAs* e *scaRNAs* de humanos
Plant snoRNA Database (2003)	Provê informação de *snoRNAs* de *Arabidopsis* e outras 18 espécies de plantas, como sequência, regiões antisenso, RNAs alvo, localização genômica, dentre outros
snoRNP Database (2010)	Coleção de sequências de *snoRNAs* e *snoRNAs* associados a proteínas (*snoRNPs*) de vários organismos
snOPY (2013)	Provê informações de *snoRNAs*, localização genômica de *snoRNAs* e RNAs alvo de diversos organimsos, além de *snoRNAs* ortólogos

Capítulo 3

Aprendizagem de Máquina

Aprendizagem de Máquina é uma subárea da Inteligência Artificial, que tem como principal foco a questão de como construir programas de computadores que automaticamente aprimoram-se com a experiência [61]. Nos últimos anos, muitas aplicações de grande sucesso utilizando aprendizagem de máquina foram desenvolvidas, tais como programas de mineração de dados que aprendem a detectar transações fraudulentas de cartão de créditos, veículos autônomos que aprendem a dirigir em estradas sem a intervenção humana, reconhecimento de fala, identificação e classificação de RNAs não codificadores (como é o caso deste trabalho) e muitos outros [61].

Neste capítulo serão apresentados conceitos gerais sobre aprendizagem de máquina (seção 3.1), além de duas técnicas: Máquinas de Vetores de Suporte (*Support Vector Machine* - SVM) (seção 3.2) e o *Explicit Decomposition with Neighbourhoods* (EDeN) (seção 3.3).

3.1 Conceitos Gerais

A palavra aprendizagem, assim como a palavra inteligência, estão relacionadas a uma grande faixa de processos que são difíceis de definir precisamente [65]. Neste trabalho, focaremos na aprendizagem de máquina que, segundo Mitchell [61], diz-se que um programa (de computador) aprendeu a partir de uma experiência E através de alguma classe de tarefas T e uma medida de performance P, se sua performance para a tarefa T, medida em P, é aprimorada com a experiência E.

Em geral, essas três váriáveis: Tarefa T, performance P e experiência E, são utilizadas para definir um problema de aprendizagem de máquina. Por exemplo, em uma aplicação de bioinformática onde um programa de computador precisa aprender a classificar se uma determinada sequência é ou não um *ncRNA*, podemos definir as três variáveis da seguinte forma:

- Tarefa T: Verificar se uma sequência de DNA é ou não um *ncRNA*;
- Performance P: Percentual de sequências de *ncRNA* classificadas corretamente;
- Experiência E: Um banco de dados de sequências conhecidas de *ncRNAs* e de sequências que não são *ncRNAs*.

A experiência E normalmente é usada em algoritmos de aprendizagem de máquina, como um conjunto de treinamento $\tau = (X_1, X_2, ..., X_i, ..., X_m)$, com m amostras. Cada amostra X_i, sendo $i = 1, 2, .., m$, é um vetor da forma $X_i = (x_1, x_2, ..., x_i, ..., x_n)$, tal que $x_1, x_2, ..., x_i, ...x_n$ são características (*features*) que descrevem X_i. Nossa hipótese é a de que exista uma função de aprendizado h capaz de gerar, a partir de uma amostra X_i, uma saída $h(X_i)$, que reflete o objetivo do aprendizado. No nosso caso, por exemplo, precisamos saber se uma amostra é ou não um *ncRNA*. Logo um algoritmo de aprendizagem de máquina busca uma função f tal que esta seja h, ou suficientemente próxima de h [61, 65]. A figura 3.1 mostra uma representação geral do funcionamento de um algoritmo de aprendizagem de máquina.

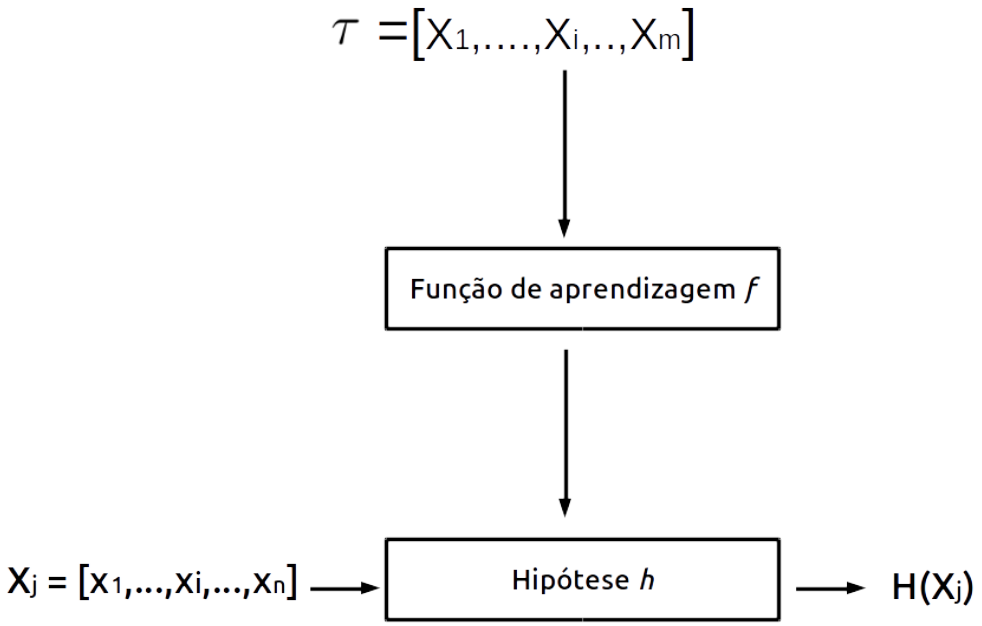


Figura 3.1: Representação geral do funcionamento de um algoritmo de aprendizagem de máquina, onde a partir de um conjunto de treinamento é buscada uma função de aprendizado que seja igual ou bastante próxima da função de hipótese, a qual a partir de uma amostra X_j retorna o valor esperado $h(X_j)$.

As amostras do conjunto de treinamento, também chamadas de vetor de *features*, possuem atributos, ou características, que podem ser classificadas em três tipos: números reais, números discretos, ou valores categóricos, como palavras ou valores booleanos

(verdadeiro, falso). Já a saída $h(x)$ pode ser: um valor real, neste caso h é chamado de estimador e a saída de estimação; ou pode retornar um valor categórico, podendo h ser chamado de classificador e a saída de classe ou categoria. No caso de algoritmos de classificação, uma entrada poderia ser formada por características de uma sequência de DNA e a saída retornaria em qual classe essa sequência de entrada se enquadra (por exemplo, uma proteína ou um *ncRNA*).

Atualmente, algoritmos de aprendizagem de máquina podem ser classificados em quatro tipos: supervisionada, não-supervisionada, semi-supervisionada e aprendizagem por reforço [61].

Na aprendizagem supervisionada, há um conjunto de treinamento, onde cada amostra possui uma característica ou classe já conhecida e com essas amostras, é então gerada uma função f igual a h (ou próxima), que será usada para descobrir as classes em que uma nova amostra pertence.

Já na aprendizagem não-supervisionada, há um conjunto de treinamento, contudo suas amostras não estão ainda classificadas ou caracterizadas completamente, tendo este aprendizado como objetivo o de encontrar alguma estrutura, ou padrão, escondido no conjunto de treinamento.

No caso da aprendizagem semi-supervisionada, nem todas as amostras possuem uma classe conhecida, apenas parte delas. A outra parte não tem amostras já classificadas. Isso é útil em casos quando são conhecidas poucas amostras já classificadas anteriormente.

Por fim, na aprendizagem por reforço, não há um conjunto de treinamento, e o sistema deve se adaptar a partir de um julgamento de suas ações, que podem ser positivas ou negativas e, a partir daí, tomar decisões que favoreçam determinado objetivo.

3.2 SVM

Nesta seção, será apresentada uma nova categoria de redes *feedfoward* conhecidas como Máquina de Vetores de Suporte (*Support Vector Machine* - SVM). As SVMs podem ser utilizadas tanto para classificação quanto para regressão [33]. Os problemas de classificação consistem em determinar qual a classe em que uma determinada amostra se encaixa, sendo que essas classes assumem valores discretos, diferentemente dos problemas de regressão, onde as classes assumem valores contínuos.

Uma SVM é uma máquina linear que tem como principal tarefa, no contextos de problemas de classificação de padrões, construir um hiperplano como superfície de decisão, de tal modo que a margem de separação entre amostras positivas e negativas é maximizada [33]. Então, na fase de treinamento, duas classes serão separadas a partir de uma

função, de forma que na fase de testes, onde dados, ainda não classificados são fornecidos a *SVM*, terão suas classes preditas. A figura 3.2 mostra um exemplo de *SVM*.

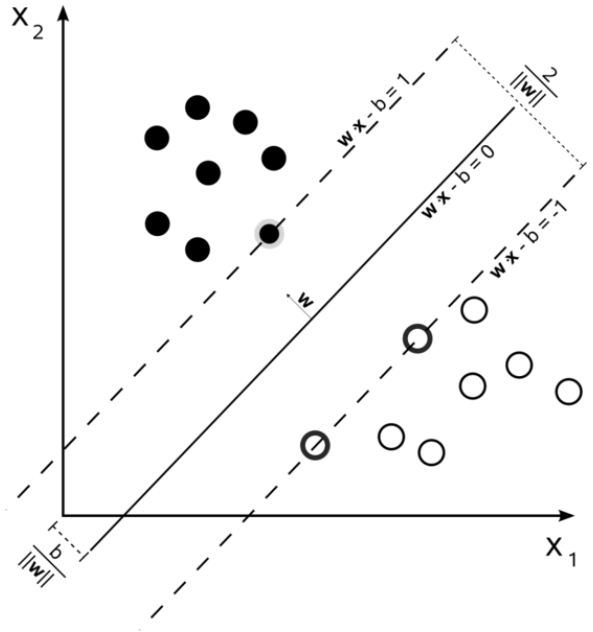


Figura 3.2: Exemplo de uma SVM [33].

Na figura 3.3, existe um conjunto de classificadores lineares (hiperplanos) separando duas classes distintas de amostras. Contudo apenas um classificador maximiza a margem de separação entre essas classes, que pode ser visto com mais detalhes na figura 3.4.

Neste contexto, temos uma SVM linear, um classificador que maximiza a margem de separação entre amostras positivas e negativas. Mas por que maximizar a margem de separação? Existem diversas explicações incluindo testes empíricos de performance [40]. Outra justificativa é a de que, caso houvesse um pequeno erro na localização das margens de separação, isto é, se ela não estivesse maximizada, haveria chances de erro de classificação, pois amostras de uma classe, por exemplo positiva, que estariam no interior da margem de separação maximizada no lado do hiperplano das amostras positivas, poderiam, em uma margem de separação menor, estar no lado das amostras negativas, causando um erro de classificação. Além disso, outra vantagem é a de diminuir a chance de cair em mínimos locais, o que pode levar a uma melhor classificação. [40].

O uso de uma SVM linear em problemas de classificação deve lidar com amostras do conjunto de treinamento formadas por vetores de características de alta dimensão. Isso dificulta a separação das amostras positivas e negativas que, em muitos casos, não podem ser divididas de forma linear. Para isso, é necessário fazer um mapeamento não linear de um vetor de entrada dentro de um espaço de características de alta dimensão [33]. Para isso, é necessário realizar uma operação chamada produto interno *kernel*, que segue do

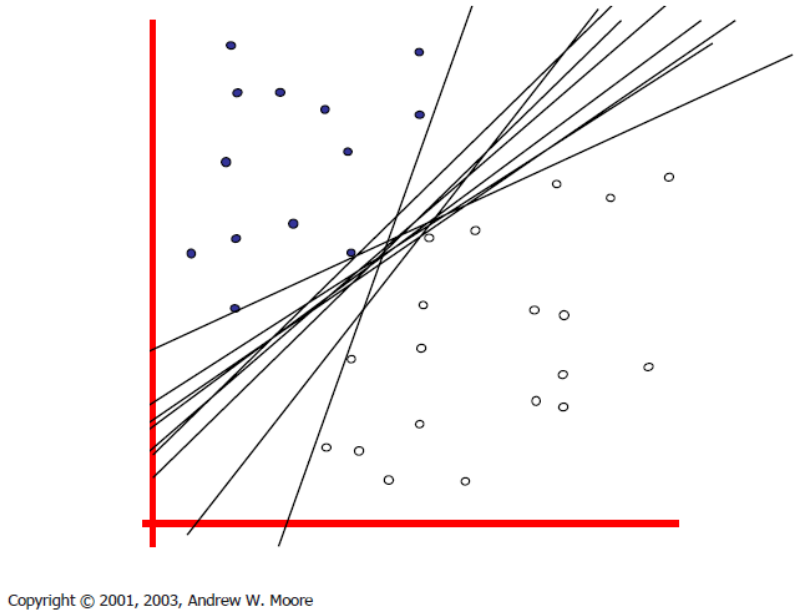


Figura 3.3: Conjunto de classificadores lineares (hiperplanos) separando duas classes distintas de amostras [62].

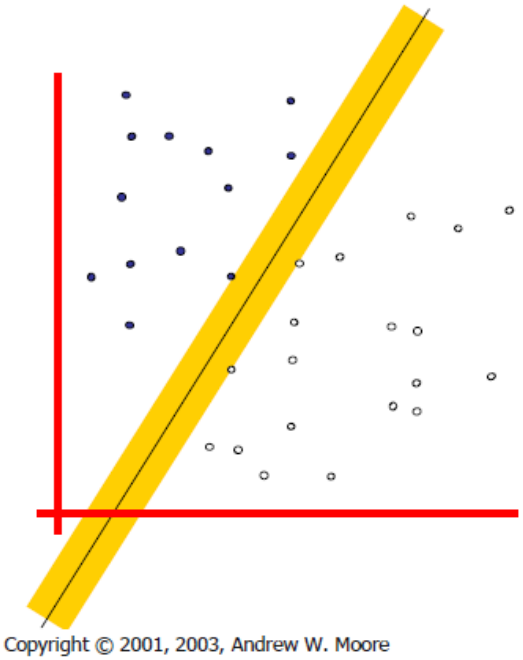


Figura 3.4: O classificador linear que maximiza a margem de separação entre as duas classes é ilustrado apresentando suas margens por um retângulo. Este é um exemplo de uma SVM linear, também chamada de LSVM [62]

teorema de Cover [17], que nos diz que, a partir de um espaço de entrada feito de padrões não linearmente separáveis, é possível transformá-lo em um novo espaço de características

onde os padrões são linearmente separáveis com alta probabilidade, se forem satisfeitas as seguintes condições: a transformação não é linear e a dimensão do espaço de característica é alta o suficiente [33]. A figura 3.6 exemplifica este mapeamento.

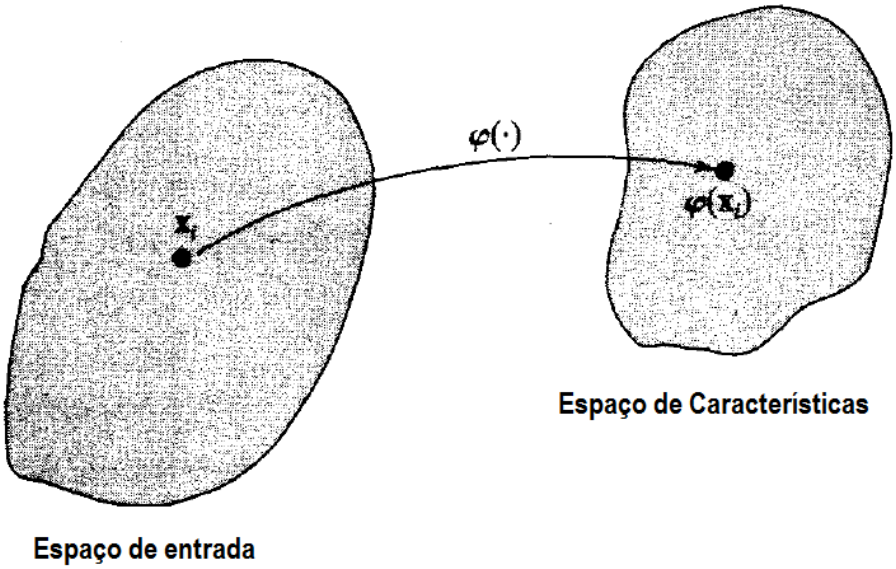


Figura 3.5: Mapeamento não linear $\varphi(.)$ do espaço de entrada para o espaço de características [33].

Um produto interno *kernel* é definido a seguir. Dado um conjunto X e uma função $K : XxX \rightarrow \Re$, K é um *kernel* em XxX se respeita as seguintes condições:

1. K é simétrico, ou seja, para cada x e $y \in X$, então $K(x, y) = K(y, x)$;
2. K é positivo semi-definido, isto é, para qualquer $N \geq 1$ e para qualquer $x_1, ...x_n \in X$, a matriz $K_{i,j} = K(x_i, x_j)$ é positiva semi-definida, ou seja, $\sum_{i,j} c_i c_j K_{ij} \geq 0$ para qualquer $c_1, .., c_n \in \Re$.

Alguns exemplos de produto interno *kernel* usados na SVM são apresentados na tabela 3.1.

3.2.1 libSVM

A *libSVM* [10] é um pacote de ferramentas e de funções que implementam a *SVM*, tanto para problemas de classificação quanto para problemas de regressão. A seguir são listadas algumas das principais ferramentas deste pacote:

- *grid.py*: identifica bons valores para os meta-parâmetros C e γ da SVM através de uma busca em grade que busca otimizar um critério de performance (acurácia por exemplo);

Tabela 3.1: Produto interno *kernel* para três tipos de SVMs [33].

Tipo de SVM	Produto interno *kernel* $K(x, x_i)$ para $i = 1, 2, ..., n$
Aprendizagem de máquina polinomial	$(x^T x_i + 1)^p$
Rede de função de base radial	$\exp(-\frac{1}{2\sigma^2}\|\|x - x_i\|\|^2)$
Perceptron de duas camadas	$tanh(\beta_0 x^T x_i + \beta_1)$

- *subset.py*: separa os dados de modo estratificado ou randômico em conjunto de treinamento e conjunto de testes;
- *svm-scale*: normaliza os vetores de características usados na SVM;
- *svm-train*: realiza a fase de treinamento e constrói um modelo usado na fase de testes e validação;
- *svm-predict*: Utiliza o modelo gerado na fase de treinamento para classificar determinada amostra no conjunto positivo ou negativo.

3.3 EDeN

Explicit Decomposition with Neighborhoods (EDeN) é um *kernel* decomposicional de grafos baseado no *Neighborhood Subgraph Pairwise Distance Kernel* (NSPDK) [15], que pode ser usado para a geração explicita de *features* a partir de grafos. Isto permite a adoção de algoritmos de aprendizagem de máquina para aprendizados supervisionados e não supervisionados.

A abordagem decomposicional, desde a criação dos *kernels* de convolução [32], tem sido o ponto de partida no desenvolvimento de *kernels* para objetos estruturados [15]. Um tipo de dado é tido como "estruturado" se é possível ser decomposto em partes, por exemplo *strings* e grafos, pois podem ser decompostas em *sub-strings* e subgrafos. A ideia principal dos *kernels* de convolução é a de definir um modo de comparar subcomponentes de um dado estruturado, por exemplo uma função de similaridade, aplicando *kernels* locais entre estes subcomponentes [15, 80].

Tais *kernels* seguem uma propriedade há muito tempo usada em aplicações de aprendizagem de máquina, que nos diz que é possível eficientemente computar *Kernels* de convolução, mesmo quando dados estruturados admitem um número exponencial de de-

composições [15]. Entretanto, embora a dimensão do espaço de *features* associado ao *kernel* torna-se exponencialmente grande, há uma probabilidade crescente que uma significante fração das dimensões do espaço de *features* será fracamente correlatada com a função de similaridade. Isto pode gerar um baixo nível de generalização em algoritmos de classificação [6, 15].

Contudo, algumas ações podem ser feitas para diminuir esse problema, como diminuir o peso de contribuição de subcomponentes grandes e (ou) limitar seu tamanho. Outra ação é a de encontrar um forte viés, relevante ao objeto em estudo, e considerar apenas um subconjunto de estruturas, limitando, então, a dimensão do espaço de *features* sem limitar a performance da predição [15]. Isso pode ser usado em preditores de alguma classe de *ncRNAs*, por exemplo, usando apenas uma região bastante conhecida da estrutura secundária de uma sequência, em vez de usar a sequência completa. O método EDeN utiliza essas ideias para aumentar a capacidade de generalização de uma aplicação de aprendizagem de máquina.

Antes de apresentar definição do NSPDK, é necessário definir alguns conceitos de grafos. Seja um **grafo** $G = (V, E)$ onde V é o conjunto de vértices e E o conjunto de arestas. A **distância entre dois vértices** u **e** v, denotada por $D(u, v)$, é o tamanho do menor caminho entre eles. **Uma vizinhança de raio** r **de um vértice** v é o conjunto de vértices a uma distância menor ou igual a r de v. Em um grafo G, um **subgrafo induzido no conjunto de vértices** $W = w_1, w_2, ...w_k$, é um grafo que tem W como seu conjunto de vértices e contém todos as arestas de G que incidem nos vértices de W. Um **subgrafo vizinho de raio** r **do vértice** v (também chamado de raiz) é um subgrafo induzido pela vizinhança de raio r do vértice v denotado por $N^v{}_r$.

O NSPDK é um *kernel* decomposicional de grafos. Neste método, um grafo é decomposto em todos os pares de subgrafos vizinhos de raio r, do qual as raízes estão a uma distância d de um dado grafo G (figura 3.6). Após isso é realizada uma contagem do número de pares idênticos (função de similaridade) dos grafos vizinhos de raio r em uma distância d, que pode ser denotada como $k_{r,d}$. O NSPDK pode ser definido então como:

$$K(G, G) = \sum_{r=0}^{r^*} \sum_{d=0}^{d^*} k_{r,d}(G, G') \tag{3.1}$$

Em outras palavras, o NSPDK decompõe um grafo em todos os pares de subgrafos vizinhos com raios $r = 0, .., *$ e distância entre as raízes dos subgrafos $d = 0, ..., *$ aplicando uma função de similaridade $k_{r,d}$ que conta o número de pares idênticos dos grafos vizinhos. Neste método, é imposto um limite superior nos valores de r e de d, limitando assim a soma dos $k_{r,d}$ *kernels* para fins de eficiência, sem prejudicar a performance nas predições.

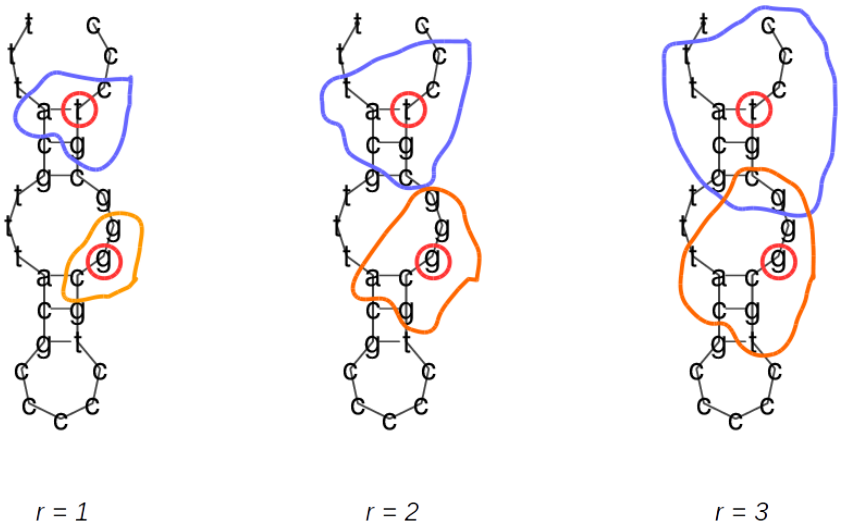


Figura 3.6: Ilustração de pares de subgrafos vizinhos de raios $r = 1, 2, 3$ e distância $d(t, g) = 5$. Note que pares de subgrafos vizinhos podem se sobrepor.

Como o número de subgrafos vizinhos cresce exponencialmente com o tamanho do raio, subgrafos muito grandes tendem a dominar o valor do *kernel* causando efeitos negativos na performance da generalização em sistemas de predição. Para isso, também é proposta neste método uma versão normalizada de $k_{r,d}$ para assegurar que a relação entre subgrafos vizinhos de todas as ordens de magnitude tenham pesos iguais, independentemente do tamanho de um subgrafo de um dado grafo [15].

O NSPDK é implementado a partir de uma solução aproximada do problema do isomorfismo de grafos (equivalente ao problema de *exact matching kernel*) [15]. Para isso foi produzido um eficiente algoritmo que produz uma codificação grafos invariantes em *strings*, e, para cada grafo invariante, é obtido um identificador através de uma função *hash* transformando essas *strings* em numeros naturais. A complexidade do algoritmo que implementa o NSPDK é $O(|V(G)||V(G_h)||E(G_h)|log|E(G_h)|)$, onde G_h é o conjunto dos grafos que possuem raizes. Logo, a complexidade do algoritmo é dominada pela computação repetitiva do grafo invariante para cada vértice de um grafo. Como este procedimento tem tempo constante para pequenos valores de d^* e r^*, pode-se concluir que o NSPDK tem complexidade linear na prática [15].

O EDeN usa o método NSPDK para a geração explicita de *features* de um dado grafo, viabilizando o uso de algoritmos de aprendizagem de máquina supervisionados, não supervisionados e semi-supervisionados. Além disso o EDeN tem a capacidade de processar grafos aninhados, com pesos e rótulos em seus vértices e arestas.

Devido a geração exponencial de *features* através da decomposição dos grafos em todos

os pares de subgrafos vizinhos com raios e distâncias crescentes, para que o método EDeN produza bons resultados, é necessário fazer uso de algoritmos de aprendizagem de máquina capazes de lidar com a grande quantidade de elementos do vetor de *features*. Algoritmos de aprendizagem como o Gradiente Descendente estocástico (SGD) são úteis para tal problema, pois conseguem ótimo desempenho em aplicações de larga escala contendo dados representados por vetores de *features* esparsos ou densos [98, 68].

3.3.1 Biblioteca EDeN

O método EDeN é implementado em linguagem Python e possui sua própria biblioteca. A biblioteca EDeN inclui diversas funcionalidades para Bioinformática e Aprendizagem de Máquina, dentre outros:

- **predição de estrutura secundária de RNAs**: inclui funções que executam e transformam estruturas secundárias, decorrentes de diversas ferramentas de predição de estrutura secundária, como RNAfold, RNAsubopt, RNAshapes, dentre outros, em representações em grafos a serem utilizadas no EDeN;
- **Manipulação de arquivos Fasta**: algumas funções são disponibilizadas para facilitar o uso de arquivos *fasta* em aplicações para o EDeN, tais como gerar uma lista de tuplas (cabeçalho, sequência) e transformar sequências fasta em grafos lineares;
- **Integração com a biblioteca scikit-learn [68]**: scikit-learn é uma poderosa biblioteca de aprendizagem de máquina para Python, incluindo inúmeros algoritmos de classificação, regressão, aglomeração, extração de *features*, dentre outros;
- **Otimização de meta-parâmetros**: na fase de treinamento dos algoritmos de aprendizagem de máquina, é possível gerar uma lista de diferentes valores de meta-parâmetros para o classificador a ser utilizado, além de variar os meta-parâmetros do *kernel* NSPDK e também parâmetros usados no pré-processamento responsável por transformar sequências em grafos (por exemplo, variar o nível de abstração da estrutura secundária no RNAshapes), aumentando, assim, a eficiência de um classificador;
- **Gerador de amostras negativas**: a partir de sequências positivas, é possível aplicar um embaralhamento de dinucleotídeos, gerando, então, um bom conjunto negativo, que pode ser usado em vários problemas de aprendizagem de máquina.

Capítulo 4

SnoReport 2.0

Neste capítulo, descreveremos o *snoReport* 2.0, submetido a *BMC Bioinformátics*.

4.1 Introduction

In recent years, methods to identify and classify non-coding RNA genes (ncRNA genes) have been continuosly improved. Many researches have shown that these ncRNAs play important roles in the cell, e.g., structural, catalytic and regulatory functions [58, 24]. It is well known that methods to study ncRNAs are challenging, due to difficulties to experimentally confirm functions performed by a ncRNA, and also to very distinct computational methods to identify and classify ncRNAs. One key problem is that ncRNA functions are closely associated to their spatial (secondary) structures, which prevent the use of methods to predict protein coding genes based only on their nucleotide sequences (primary structures). These computational methods focus on predicting candidates that have to be experimentally confirmed.

Identification of ncRNAs have been developed for a variety of organisms [31, 13, 96, 53], with the objective of constructing sets of different classes of ncRNAs. In particular, snoRNAs [25] are 60 to 300 nt ncRNAs, classified based on their characteristic sequence elements, called *boxes*, in two main classes: H/ACA box snoRNAs and C/D box snoRNAs. In humans [29], snoRNAs are usually found in intronic regions where, after splicing reaction, they escape from degradation by forming a protein complex [25]. Usually snoRNAs have a short stretch of sequence complementary with target RNAs, like rRNAs, tRNAs and snRNAs, performing chemical modifications on them. C/D box snoRNAs contains fibrillarin that promotes the 2'O-methylation on target RNAs, while H/ACA box snoRNAs contains dyskerin that catalyzes the conversion of uridine to pseudouridine [34, 25].

H/ACA box snoRNA and C/D box snoRNA have distinct secondary structures. H/ACA box snoRNAs are formed by a double hairpin loop structure with two short-single stran-

ded regions containing box H (ANANNA), located between the two hairpins loops, and box ACA (ACA) followed by 3 nt upstream the 3' end. The hairpin loops have bulges, or recognition loops, which form the antisense element for target RNAs. Normally the first unpaired nucleotide inside the recognition loop is an uridine located 13-16 nt before the H and ACA boxes [95, 25, 49]. Figure 5.1 shows a schematic secondary structure of H/ACA box snoRNA.

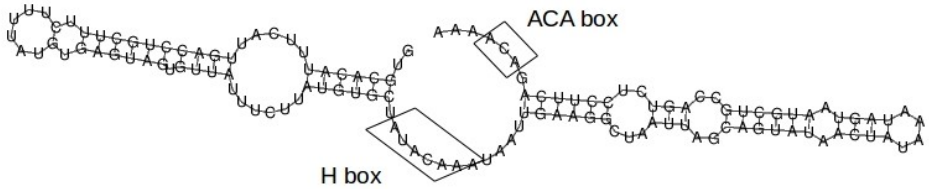


Figura 4.1: Example of H/ACA box snoRNA.

C/D box snoRNAs are formed by two conserved boxes C (RUGAUGA, where R is a purine) and D (CUGA) near their 5' and 3' end, separated by a short stem (3-10 nt). Inside the loop between C and D boxes, usually there is the presence of imperfect copies of C and D boxes, called C' and D'. Normally the antisense element is located 5 nt upstream D' and D' boxes. Figure 5.2 shows a schematic secondary structure of a C/D box snoRNA.

`snoReport` [34] is a tool that identifies the two main classes of snoRNAs in single sequences, using a combination of secondary structure prediction and machine learning. In contrast to previous methods for snoRNA identification (except snoSeeker [95]), `snoReport` prediction does not use information of putative target sites within ribosomal or spliceosomal RNA (this information can dramatically improve identification sensibility and specificity). However, many orphan snoRNAs have been discovered with the `snoReport` approach. The target(s) of orphan snoRNAs are not known, consequently such genes would be missed by target depending on the identification method [34, 43]. Beyond this, some snoRNAs are shown to target specific mRNAs, suggesting other functions, e.g., interference with A-to-I editing [87, 34, 43, 25, 45]. In order to identify C/D box and H/ACA box snoRNAs, `snoReport` uses position-specific weighted matrices (PWM's)

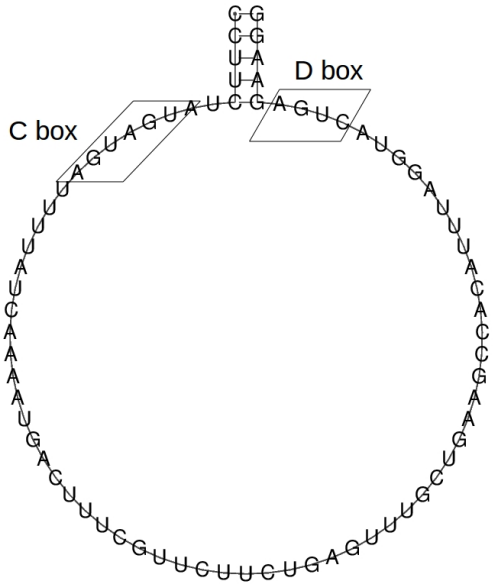


Figura 4.2: Example of C/D box snoRNA.

to identify boxes, together with a set of restrictions related to the secondary structure prediction, usually, restrictions about distance between regions of the secondary structure, and if it forms hairpins for H/ACA box snoRNAs, or forms the loop for a C/D box snoRNA.

`snoReport` produced good results. In the test phase, `snoReport` presented 96% of sensitivity and 91% of specificity for the C/D box snoRNA classification, while for H/ACA box snoRNAs, it has shown 78% of sensitivity and 89% of specificity. However, `snoReport` has been trained on almost exclusively mammalian sequences, having used some default parameters for the Support Vector Machine (SVM) classifier. To date, many new sequences of snoRNAs for different vertebrate organisms have been identified, and experimentally confirmed. Furthermore, many tools and databases used to build `snoReport` have been improved. This suggests that `snoReport` has to be updated, in order to use new data and refined machine learning techniques to improve its performance.

Here, we present an update of `snoReport`, by extracting new features for both box C/D and H/ACA box snoRNAs, developing a more sophisticated technique in the SVM training phase (with recent data from vertebrate organisms and a different approach to refine the C and γ SVM parameters), and using new versions of the tools and data bases previously taken to build `snoReport`. To validate this new version of `snoReport`, we tested it in different organisms. These experiments have shown a very good performance.

This text is organized as follows. In the first section, we describe the methods used for building the new version of `snoReport`, particularly, data sources and the new workflow, besides the new features and details of the training phase. Next, we discuss the results

obtained by the new version of `snoReport` with different species of organisms. Finally, we conclude and suggest future work.

4.2 Implementation

First, data sources, software components, and the workflow used to build the new `snoReport` are described. Next, the new attributes for boxes H/ACA and C/D snoRNAs used in the SVM classifier are shown.

4.2.1 Data sources

Since `snoReport` uses a machine learning approach, data used for the training and test phases were divided in two sets: positive samples and negative samples, each having the two classes of snoRNAs. The positive sample set was composed of H/ACA box and C/D box snoRNAs, while the negative one was obtained from a dinucleotide shuffling procedure executed in the positive samples with the EDeN [15] library.

The positive sequences from each class of snoRNAs were divided in two datasets, to be used in the learning process. In order to avoid overfitting, these datasets were created such that very similar sequences would not be stored in different datasets. First, we clustered the sequences using ClustalW [84] with criterion *nucleotide similarity*, which generated 157 clusters for C/D box snoRNA and 101 clusters for H/ACA box snoRNA. After, 10 sequences from distinct vertebrates organisms were extracted from each cluster, noting that clusters containing less than 10 sequences were discarded. Therefore, a consensus sequence from each cluster was obtained with ClustalW and Cons (for EMBOSS [72]), and these sequences were used to generate a distance tree, with the neighbour-joining method [74] from ClustalW2 - phylogeny [59]. The next step was to divide this distance tree in two parts, which allowed to create the two datasets containing similar sequences. The generated trees of C/D box snoRNA and H/ACA box snoRNA clusters can be viewed on supplemental material.

Table 5.1 shows the number of sequences of each dataset.

Tabela 4.1: Number of sequences of Datasets 1 and 2 of both C/D box and H/ACA box snoRNAs.

	Dataset 1	Dataset 2
C/D box snoRNAs	750	520
H/ACA box snoRNAs	490	420

Position-specific weight matrices (PWMs) were used to represent each characteristic sequence motif of H/ACA box and C/D box snoRNAs. These PWMs were obtained by scanning the boxes from snoRNAs of vertebrates. A PWM shows the probability that each nucleotide can be found in a particular position of a box motif. These PWMs generate scores used to identify boxes in a candidate sequence. To create thresholds for each box, we scanned snoRNA sequences with a window size equal to the length of the corresponding box. The scanned candidate boxes that were not true boxes were classified as negative boxes. Thus, we generated a density plot to define the thresholds. Figures 4.3, 4.4, 4.5 and 4.6 show the generated density plots.

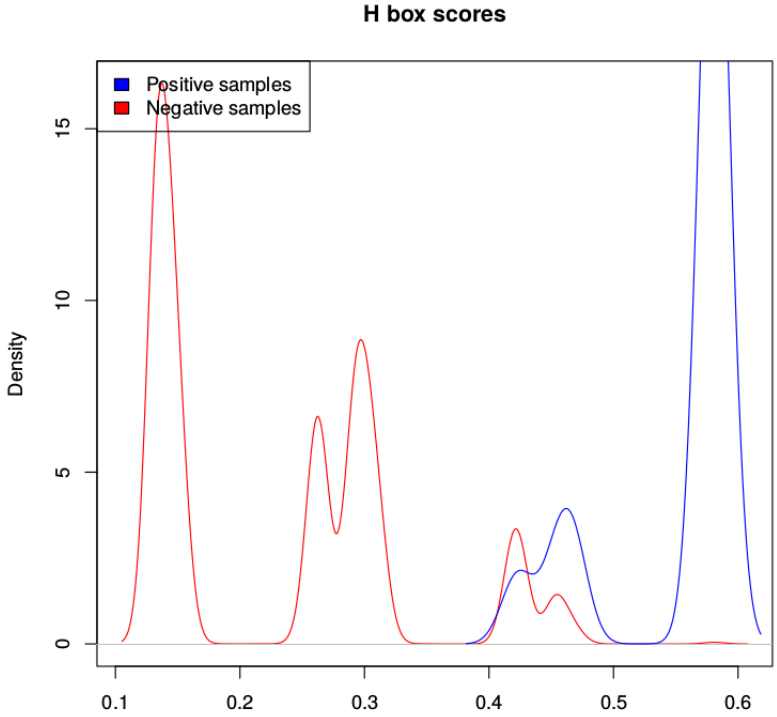


Figura 4.3: Density plot of H box PWM-based scores

In the validation phase, we used sets of predicted, and partially confirmed with experiments, snoRNAs from many organisms: human [95], nematodes [97], Drosophilids [38], chicken [79], platypus [76] and leishmania [50]. These sequences were manually extracted from additional files of each paper (originally in *pdf* format and *doc* format tables).

4.2.2 Software components

RNA secondary structure prediction was performed using Vienna RNA Package, current version 2.15, in particular RNAfold [36], RNAz [30] and RNALfold [35]. RNAfold predicts a secondary structure associated with the minimum free energy (MFE) of a single stranded RNA or DNA sequence. RNALfold computes locally stable RNA secondary structure

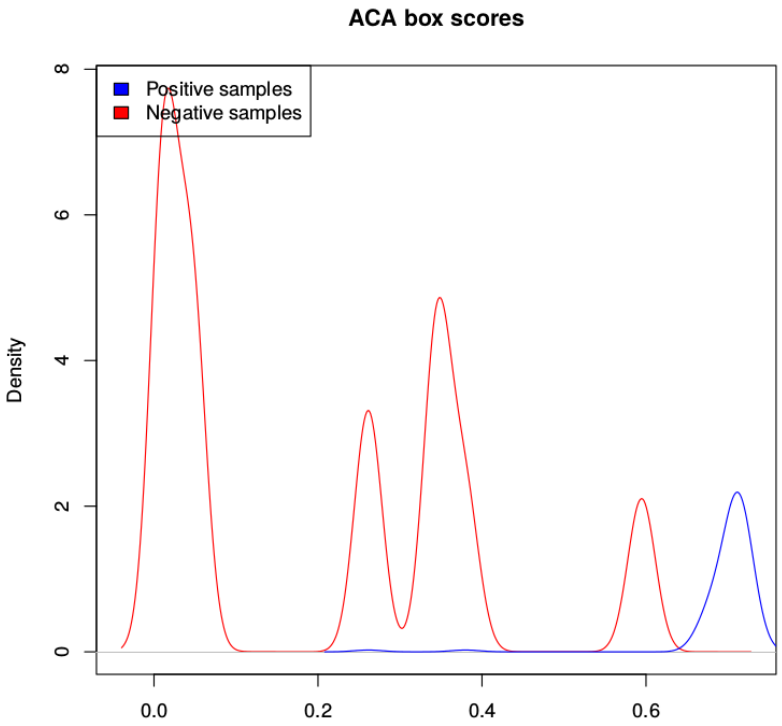


Figura 4.4: Density plot of ACA box PWM-based scores

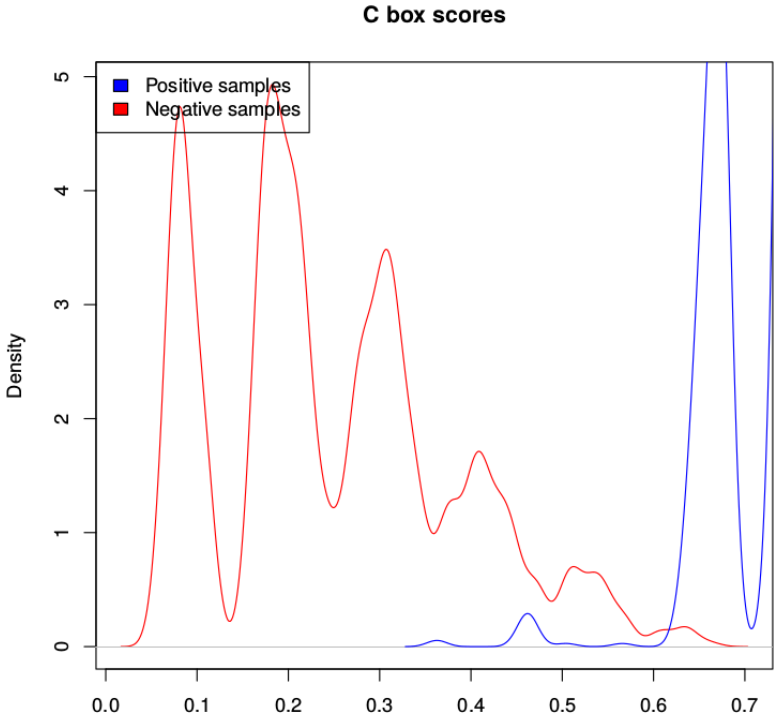


Figura 4.5: Density plot of C box PWM-based scores

with a maximal base pair span. It was used here in order to find the start position of a H/ACA box snoRNA candidate. RNAz was executed to calculate *zscore*, an attribute of

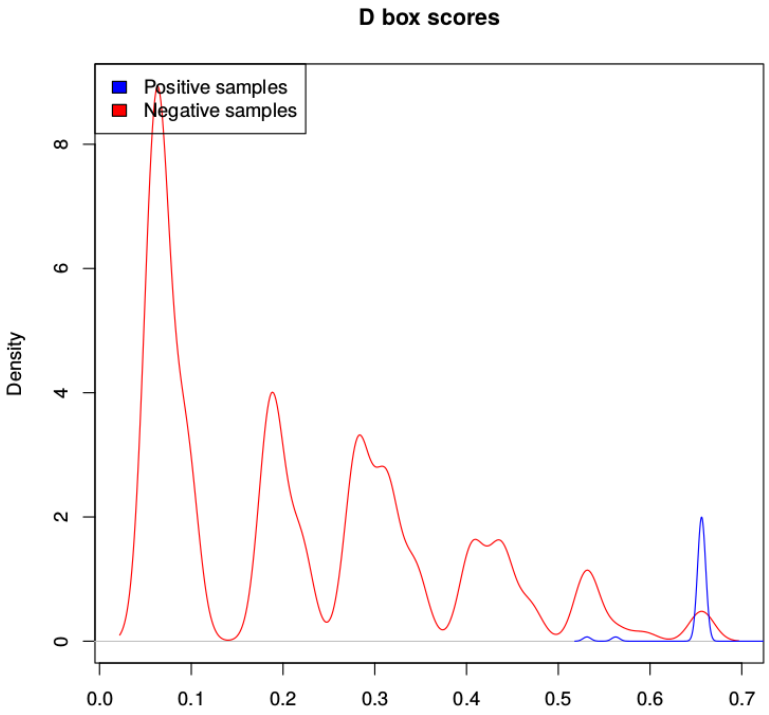


Figura 4.6: Density plot of D box PWM-based scores

the feature vector of H/ACA box snoRNA that represents the thermodynamic stability of a ncRNA secondary structure.

Many tools available in the libSVM version 3.20 [10] performed the classification of H/ACA box snoRNA and C/D box snoRNA:

- *grid.py*: to identify good values for C and γ SVM parameters;
- *svm-scale*: to scale the feature vector;
- *svm-train*: to perform training and build a model used for predicting new candidates in the *svm-predict* tool;
- *svm-predict*: to predict sequences not used in the training phase.

In order to measure different performance measures (not available in libSVM), we developed a script using scikit-learn library [68] to calculate Accuracy, F-score, Average Precision, ROC AUC score and Residual sum of squares (RSS). Using these software components, the `snoReport` 2.0 was entirely rewritten in the C language.

4.2.3 Identifying snoRNA candidates in genomic sequences

As said before, both classes of snoRNAs, H/ACA box and C/D box, can be distinguished by their characteristic *boxes*, and some specific secondary structure features. For this,

each class of snoRNA has a specific way to searching for candidates, described as follows.

Searching for H/ACA box snoRNAs in a genome sequence was performed with the following steps (Figure 4.7):

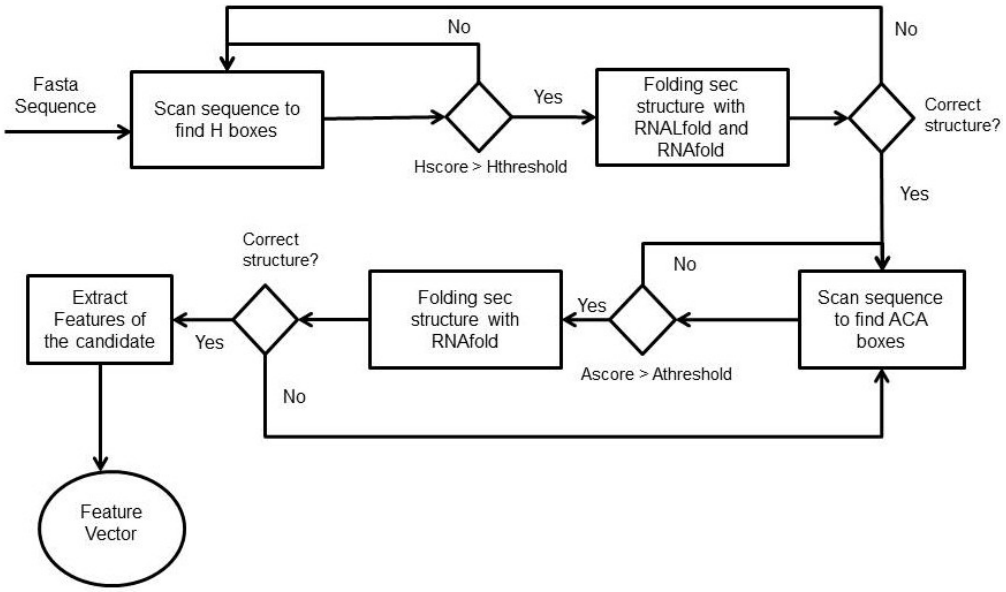


Figura 4.7: Workflow for H/ACA identification on `snoReport` 2.0.

1. The genome sequence is scanned in order to find potential H boxes with PWM-based scores above a certain threshold;
2. If one H box candidate has a good PWM-based score, we executed first RNALFold to find the start position of the H/ACA box snoRNA candidate, and then RNAfold with some constraints to predict its secondary structure;
3. If the sequence between the start position and the H box candidate has a correct secondary structure, we look for ACA box candidates with a maximum distance of 120 nts and presenting a PWM-based score above a certain threshold;
4. Finally, RNAfold is called for the sequence between H box and ACA box. If this sequence has the correct structure, features for this candidate were extracted.

Restrictions used to predict secondary structure are specific for each class of snoRNA. For the secondary structure of H/ACA box snoRNA, the region upstream of box H and the region between box H and ACA are used to fold into single stem loop structures. In the cell, snoRNA interacts with a set of different proteins that stabilize the large interior loop containing the target binding site. Without these proteins, standard MFE folding algorithms can predict base pairs within this loop. Therefore, to open the target

region, we constrained the 14^{th} base upstream of boxes H and ACA, and in most cases the complete interior loop turns out to be unpaired in the MFE structure. Figure 4.8 shows the canonical representation of H/ACA box snoRNAs.

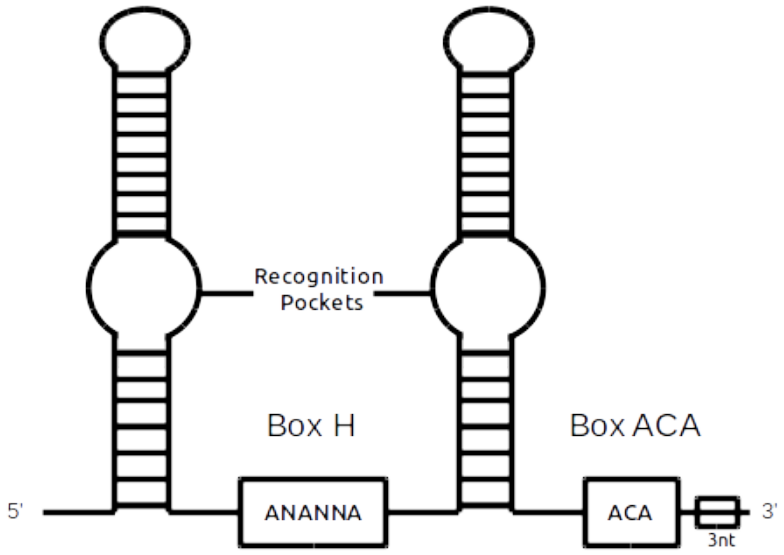


Figura 4.8: Canonical secondary structure of H/ACA box snoRNA, with two hairpins and two short-single stranded regions containing boxes H and ACA (located 3 nt upstream of the 3' end). The hairpin contains bulges, or recognition loops, which form complex pseudoknots with the target RNA, where the target uridine is the first unpaired base [95, 25].

Searching for C/D box snoRNAs in a genome sequence was performed with the following steps (Figure 4.9):

1. The genome sequence is scanned in order to find C boxes with PWM-based scores above a certain threshold;
2. If the C box candidate has a good PWM-based score, we look for D box candidates with a maximum distance of 200 nts with PWM-based score above a certain threshold;
3. The candidate has its kink-turn structure tested, and in case of having the correct one, RNAfold is called to predict its secondary structure;
4. If it has the correct secondary structure, features for this candidate are extracted.

For the secondary structure of C/D box snoRNA, the complete region from the start of box C to the end of box D has to remain unpaired. Many studies have shown that C/D box snoRNAs must have a perfect kink turn structure which boxes C and D [5, 90, 93]. For

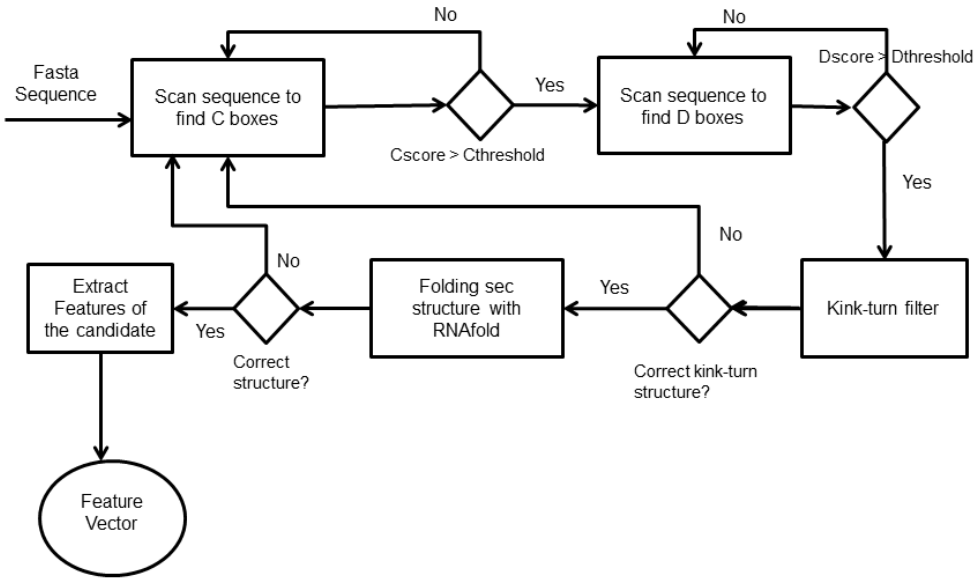


Figura 4.9: Workflow for C/D identification on `snoReport` 2.0.

this, *snoreport* 2.0 has a kink-turn structure test, where a C/D box snoRNA candidate must have: G•A dinucleotides in box C (RU**GA**UGA) and box D(CU**GA**); at least one uridine on the U-U pair (RUGA**U**GA and C**U**GA); and a Watson-Crick base pair between the 6th nt of C and the 1st nt of D box (RUGAU**G**A and **C**UGA). Figure 4.10 shows the kink turn structure of C/D box snoRNA, and Figure 4.11 shows the canonical representation of a C/D box snoRNA.

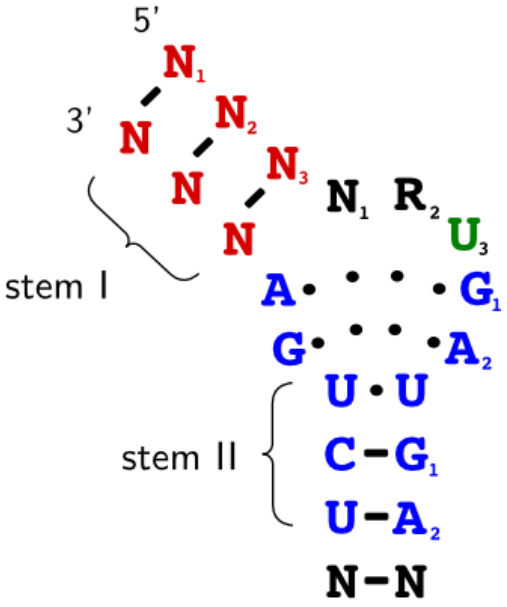


Figura 4.10: Kink turn structure of C/D box snoRNA [5]

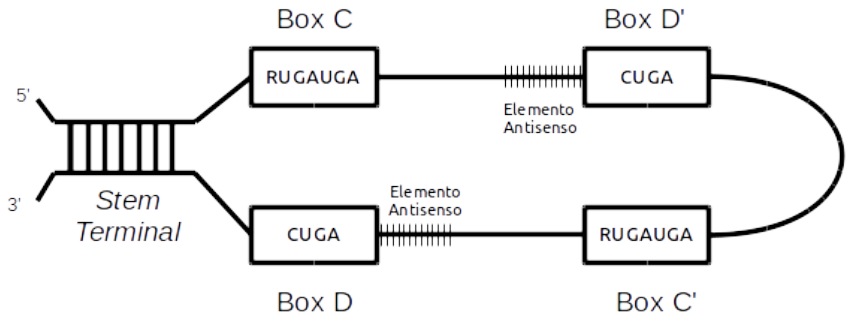


Figura 4.11: Canonical secondary structure of C/D box snoRNA [95]. Boxes C and D are located near 5' and 3' ends, noting that they are frequently folded together by a short stem. Normally, imperfect copies of C and D boxes, called D' and C', are located internally in the loop, ordered as C, D', C' and D. The target RNA is guided by antisense elements located upstream of D box or D' box.

Extraction of feature vectors

After a snoRNA candidate passes through all the previously described filters, and fold the secondary structure, `snoReport` 2.0 extracts some attributes from a H/ACA (C/D) box snoRNA candidate, in order to build a feature vector, which will be the input for the Support Vector Machine (SVM). Some changes in the feature vectors of both H/ACA box and C/D box snoRNA candidates were introduced, compared to the previous version of `snoReport`.

In the feature vector of H/ACA box snoRNA, the following new attributes were included: *AC*, *GU*, *zscore*, *Hscore*, *ACAscore*,*LloopSC*, *RloopSC*, *LloopYC*, *RloopYC*, *LloopSym* and *RloopSym*. Table 4.2shows all the attributes that have to be extracted from a H/ACA box snoRNA candidate.

The attribute $mfeC$ shows the MFE of folding with constraint nucleotides, providing the information of how much "effort"is needed to force the candidate sequence to fit the requested structure, or if the candidate is more stable in another structure. AC, GC and GU contents are used to distinguish ncRNAs from different RNAs. For example, the human genome has approximately 42% of GC content, but single sequences of miRNAs and H/ACA box snoRNAs have 50% of average GC content [88]. The *zscore* feature is obtained with RNAz [30], representing the thermodynamic stability of a ncRNA secondary structure. Values *Hscore* and *ACAscore* were computed using PWMs of H box and ACA box, respectively. Attributes *LseqSize*, *RseqSize*, *LloopSC*, *RloopSC*, *LloopYC*, *RloopYC*, *LloopSym* and *RloopSym* help to discriminate arbitrary double stem loop structures from H/ACA stem loop structures.

In the feature vector of C/D box snoRNA, new attributes were also included: *zscore*,

Tabela 4.2: Attributes extracted from a H/ACA box snoRNA candidate.

$mfeC$	MFE of the secondary structure with restrictions in RNAfold
AC, GU, GC	AC, GU and GC content
$zscore$	zscore computed by RNAz
$Hscore$	Score of the H box
$ACAscore$	Score of the ACA box
$LseqSize$	Number of nucleotides before the H box
$RseqSize$	Number of nucleotides between H and ACA boxes
$LloopSC$	Lenght of the loop, where we find the pocket region containing the target region, near to the H box
$RloopSC$	Length of the loop, where we find the pocket region containing the target region, more close to the ACA box
$LloopYC$	Symmetry of the loop containing the pocket region near to the H box
$RloopYC$	Symmetry of the loop containing the pocket region near to the ACA box
$LloopSym$	Symmetry of all loops before H box
$RloopSym$	Symmetry of all loops before ACA box

bpStem, $lu5$, $lu3$, *stemUnpCbox*, *stemUnpDbox*. Table 4.3 shows the attributes that have to be extracted from a C/D box snoRNA candidate.

Attributes $mfeC$ and mfe are used to distinguish both RNAfold folding procedures, with and without restrictions, respectively. Attributes E_{avg} and E_{stdv} represent average and standard deviation of folding energy for random sequences with identical nucleotide frequency in RNAz. Values $Cscore$ and $Dscore$ were computed using PWMs of C box and D box, respectively. The other attributes (*bpStem*, $lu5$, $lu3$, *stemUnpCbox*, *stemUnpDbox*) allow to distinguish C/D box snoRNAs from other RNAs according to the stem found by the secondary structure prediction.

Training and test phases

Figure 4.12 shows the training and test phases workflow of `snoReport` 2.0.

Since we have two datasets for each class of snoRNA, two different training and test phases were performed, one with dataset 1 as training and dataset 2 as test, and vice versa. For each dataset, negative samples were generated with a dinucleotide shuffling procedure from EDeN. In order to more reliable measure the quality of the learning, we repeated the training and test phase 10 times for each dataset, generating on each time new negative samples. After creating the training and test dataset, the feature vector was scaled from -1 to 1 using *svm-scale* for a better SVM classification.

Tabela 4.3: Attributes extracted from a C/D box snoRNA candidate.

mfe	MFE of the secondary structure without restrictions in RNAfold
$mfeC$	MFE of the secondary structure with restrictions in RNAfold
E_{avg}	MFE average
E_{stdv}	MFE tandard deviation
ls	Length of the terminal stem
Dcd	Distance between C and D boxes
C_{score}	score of the C box
D_{score}	score of the D box
GC	GC content
$zscore$	zscore obtained by RNAz
$bpStem$	Number of base pairs on the terminal stem
$lu5$	Number of unpaired nucleotides inside the stem before C box
$lu3$	Number of unpaired nucleotides inside the stem after D box
$stemUnpCbox$	Number of unpaired nucleotides between the stem and the C box
$stemUnpDbox$	Number of unpaired nucleotides between the D box and the stem

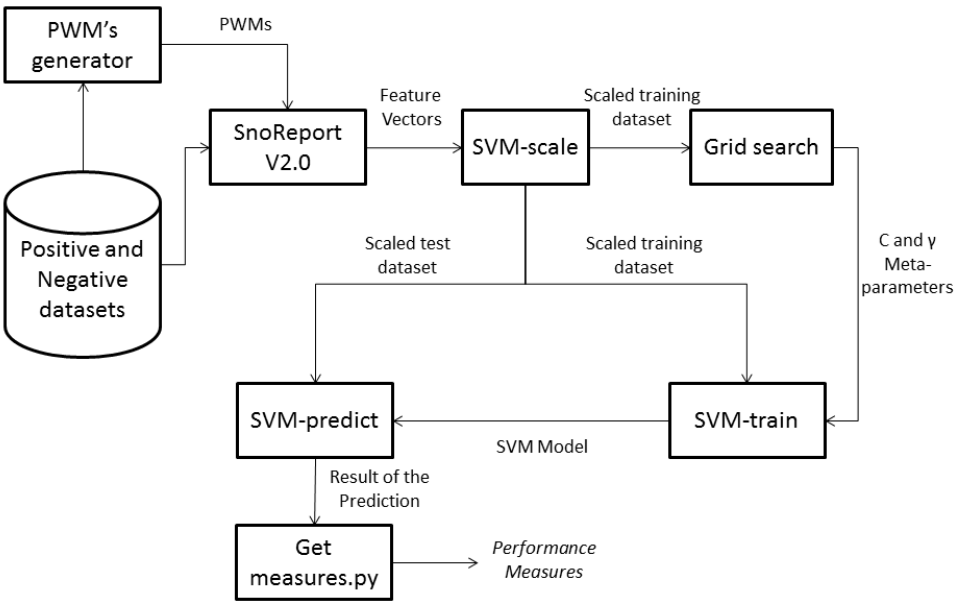


Figura 4.12: Workflow of the learning phase of `snoReport` 2.0.

The next step was to perform a grid search for the C and γ parameters, using *grid.py* (available in libSVM v3.20), a parameter selection tool for C-SVM classification that uses the RBF (radial basis function) kernel. It uses a cross validation technique (in our

case, 10-fold) to estimate the accuracy (another criteria could be used as well) of each combination of C and γ in the specified range, which allowed to choose the best values. Following Hsu [37], "a practical method to identify good parameters is to try exponentially growing sequences of C and γ". Therefore, we first investigated all the combinations of these two parameters ranging both from 2^{-15} to 2^{15}, shifting 2^1 for each step of the grid-search (for example, $2^{-15}, 2^{-14}, ...2^{15}$). Figure 4.13 shows an example of the performed grid search.

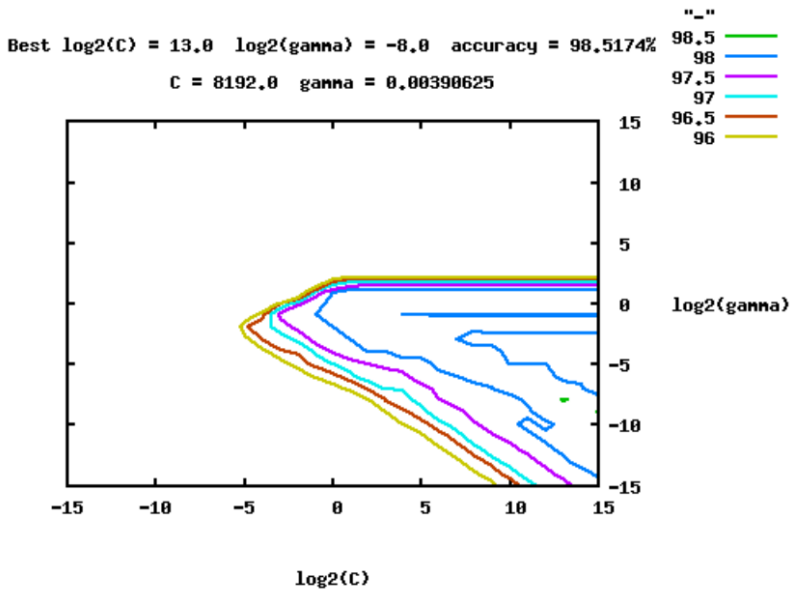


Figura 4.13: Grid search using accuracy as a criterion for C/D box snoRNA classification. Each line represents the accuracy obtained in the training phase, using parameters C and γ with 10-fold cross validation. Here, the green line represents 98.5% of accuracy using any point of this line.

After estimating parameters C and γ, the training phase was performed using *svm-train*, which used C-SVM with the RBF kernel and probabilities estimates enabled. After training, we obtained a classifier (called model) used as input in *svm-predict* to predict snoRNAs from sequences not used in the training phase.

For a more refined analysis, we used the scikit-learn library [68], which allowed to obtain three performance measures to better evaluate and compare the `snoReport` 2.0 with the previous snoReport:

- Fixed threshold (Accuracy and F-score): a sample is classified as positive if its score (or probability) is above a certain fixed threshold;
- Dynamic threshold (Average precision - APR - and Area Under the Curve - AUC): measure based on moving thresholds along the positive class. It returns the area under the precision-recall curve (APR) and the area under the ROC curve (AUC);

- Residual sum of squares: shows the discrepancy between data and an the estimator model.

4.3 Results

First, we present statistics of the performed tests. Next, we discuss the results obtained by executing `snoReport` 2.0 on real data of different organisms.

4.3.1 Statistics

To identify H/ACA box and C/D box snoRNAs, we built two different datasets for each class of snoRNAs. For the learning phases, we used one dataset as training and the other for test (vice and versa). Each training was repeated 10 times, and our results show the average of the obtained results and their corresponding standard deviation. Tables 4.4 and 4.5 show the test phase results of each snoRNA class obtained with `snoReport` 2.0.

Tabela 4.4: Test phase results for H/ACA box snoRNAs: accuracy (Acc), F-score (F-SC), Average Precision (APR), Area under the ROC curve (AUC) and Residual Sum of Squares (RSS). Dat1 and Dat2 means Dataset 1 and Dataset 2, respectively, and SD means *standard deviation*.

	Acc (%)	FSC (%)	APR (%)	AUC (%)	RSS
Dat1 average→Dat2	97.31	93.07	97.85	98.94	0.022
Standard deviation	0.24	0.60	0.20	0.20	0.002
Dat2 average→Dat1	97.43	94.71	98.66	99.33	98.66
Standard deviation	0.51	1.06	0.42	0.20	0.004
All trainings' average	97.37	93.89	98.25	99.14	0.021
All training' SD	0.39	1.19	0.53	0.28	0.003

In order to compare the results with the old version of `snoReport`, we ran the datasets used as test in `snoReport` 2.0 on `snoReport` 1.0. Tables 4.6 and 4.7 show the results.

Comparing these results, we can see that `snoReport` 2.0 presented a better performance to predict vertebrate data, with all the performance measures above 90%. For H/ACA box snoRNA, the F-score, which consider both precision and recall, `snoReport` 2.0 was 10.9% better, improving the old version. For C/D box, we again see an increase of 14,92% on F-score, and better performances on all measures. With that, `snoReport` 2.0 shows a considerable improvement in comparison with its old version.

Tabela 4.5: Test phase results for C/D box snoRNA. accuracy (Acc), F-score (F-SC), Average Precision (APR), Area under the ROC curve (AUC) and Residual Sum of Squares (RSS). Dat1 and Dat2 means Dataset 1 and Dataset 2, respectively, and SD means *standard deviation*.

	Acc (%)	FSC (%)	APR (%)	AUC (%)	RSS
Dat1 average→Dat2	94.37	93.67	98.43	98.82	0.044
Standard deviation	1.65	2.04	0.77	0.51	0.012
Dat2 average→Dat1	96.19	94.94	98.80	99.11	0.029
Standard deviation	0.90	1.25	0.53	0.63	0.007
All trainings' average	95.28	94.30	98.61	98.96	0.037
All trainings' SD	1.60	1.77	0.67	0.58	0.012

Tabela 4.6: Results of the old version of `snoReport` for H/ACA box snoRNAs using the same datasets used as test on the new version, where: accuracy (Acc), F-score (F-SC), Average Precision (APR), Area under the ROC curve (AUC) and Residual Sum of Squares (RSS). Dat1 and Dat2 means Dataset 1 and Dataset 2, respectively, and SD means *standard deviation*.

	Acc (%)	FSC (%)	APR (%)	AUC (%)	RSS
Dat2	92.71	80.62	94.42	96.33	94.42
Standard deviation	0.59	1.23	1.66	0.37	0.004
Dat1	93.31	85.36	95.61	97.37	0.054
Standard deviation	0.25	0.47	0.86	0.28	0.002
All trainings' average	93.02	82.99	95.01	96.85	0.055
All training' SD	0.53	2.61	1.42	0.63	0.003

4.3.2 Validation on real data

To verify the quality of prediction, a validation on real data was performed. We executed `snoReport` 2.0 with a set of previously predicted vertebrate and invertebrate sequences, some of them partially confirmed in experiments in humans, nematodes, drosophilids, platypus, chickens and leishmania. Tables 4.8 and 4.9 shows the summary of these results in vertebrates and invertebrates organisms, respectively.

Yang et al. [95] identified 54 snoRNAs, 21 C/D box and 32 H/ACA box in human, using snoSeeker, a method based on probabilistic models, pairwise whole-genome alignments of eukaryotes, in which the user can include information of the putative target region or not (to find orphan snoRNAs). The previous version of `snoReport` predicted 11 out of 21 C/D box snoRNAs and 23 out of 32 H/ACA box snoRNAs, while `snoReport` 2.0 predicted 21 C/D box snoRNAs and 28 H/ACA box snoRNAs.

Schmitz et al. [76] identified 166 individual snoRNAs in a platypus brain cDNA library, generated from small ncRNAs. After, using BLAST searches in platypus genomic

Tabela 4.7: Results of the old version of `snoReport` for C/D box snoRNAs using the same datasets used as test on the new version, where: accuracy (Acc), F-score (F-SC), Average Precision (APR), Area under the ROC curve (AUC) and Residual Sum of Squares (RSS). Dat1 and Dat2 means Dataset 1 and Dataset 2, respectively, and SD means *standard deviation*.

.

	Acc (%)	FSC (%)	APR (%)	AUC (%)	RSS
Dat2	90.81	78.27	92.36	96.38	0.076
Standard deviation	0.40	0.73	1.56	0.68	0.003
Dat1	88.67	80.49	96.61	97.79	0.088
Standard deviation	0.25	0.35	0.74	0.42	0.002
All trainings' average	89.74	79.38	94.49	97.09	0.082
All trainings' SD	1.15	1.27	2.48	0.91	0.007

Tabela 4.8: Results of executing `snoReport` 2.0 with snoRNA sequences of vertebrate organisms. The number of predicted candidates compared to the number of candidates identified in the cited references are shown.

Human		
Yang et al. [95]	C/D: 21/21	H/ACA: 28/32
Platypus		
Schmitz et al. [76]	C/D: 42/144	H/ACA: 45/73
Chicken		
Shao et al [79]	C/D: 112/132	H/ACA: 66/69

Tabela 4.9: Results of executing `snoReport` 2.0 with snoRNA sequences of invertebrate organisms. The number of predicted candidates compared to the number of candidates identified in the cited references are shown.

Nematodes		
Zemann et al. [97]	C/D: 32/108	H/ACA: 46/60
Drosophilids		
Huang et al. [38]	C/D: 2/63	H/ACA: 39/56
Leishmania		
Liang et al. [50]	C/D: 0/62	H/ACA *A-like*: 0/37

sequences, they found 51 more sequences of snoRNAs. Futhermore, they found cis- and trans-duplication distribution patterns for snoRNAs, which had not been described in other vertebrates, but only in nematodes. Snoreport 2.0 predicted 42 out of 144 C/D box snoRNAs, and 45 out of 73 H/ACA box snoRNAs.

Shao et al. [79] identified 132 C/D box snoRNAs in chicken using *CDseeker* and 69 H/ACA box snoRNAs using *ACAseeker* (both programs are used in snoSeeker [95]). We predicted, with `snoReport` 2.0, 112 out of 132 C/D box snoRNAs, and 66 out of 69 H/ACA box snoRNAs.

Zemann et al. [97] used a combination of high-throughput cDNA library screening and computational search strategies to find 121 snoRNAs (168 are shown in the supplementary material) in *Caernorhabditis elegans*. Our `snoReport` 2.0 predicted 32 out of 108 C/D box snoRNAs, and 46 out of 60 H/ACA box snoRNAs.

Huang et al. [38] performed a large-scale genome wide analysis to identify both classes of snoRNAs in *Drosophila melanogaster* using experimental and computational RNomics methods, having found 119 snoRNAs. Our `snoReport` 2.0 predicted 2 out of 63 C/D box snoRNAs, and 39 out of 56 H/ACA box snoRNAs.

Finally, Liang et al. [50] used a genome-wide screening approach to identify 62 C/D box snoRNAs and 37 H/ACA box snoRNAs of closely related pathogens of *Leishmania major*. We did not identify any C/D box or H/ACA box snoRNA. It is interesting to note that H/ACA box snoRNAs from *Leishmania major* are quite different from the canonical H/ACA box snoRNAs of yeast and vertebrate. For example, they lack a recognizable H box, presenting an AGA box instead of an ACA box [34]. Our `snoReport` 2.0 was designed to identify canonical snoRNAs from many different organisms, thus to predict H/ACA box snoRNAs from organisms that are different from the canonical model, we should use a different training set, together with a revision of the attributes of the feature vector.

4.4 Discussion

In this work, we refined the training phase of the SVM method, using different features in the characteristic vector, more data from different vertebrate organisms, and new versions of the tools and data bases used to build the first version of `snoReport`. We carefully chose good values for the C and γ SVM parameters using grid searches.

All these steps allowed us to improve the performance of `snoReport`, avoiding false positives and finding more snoRNAs. H/ACA box snoRNA classifier had an improvement of 10.9% regarding to F-score, with the same data, when compared to the first version of `snoReport`. Besides, the high score achieved from average precision,ROC AUC score and

RSS show us that the predictions have a high degree of reliability. The same could be observed for C/D box snoRNA classifier, which have an improvement of 14.92% regarding to F-score, and more than 90% of all performance measures presented, allowing us to have high rate of quality on each prediction.

The validation phase showed that `snoReport` 2.0 predicted 67.43% of sequences from vertebrates organisms, which shows that `snoReport` 2.0 can identify snoRNAs with significantly higher precision while maintaining recall. It is noteworthy that many sequences used for validation was not yet experimentally validated, and maybe some of them can be false positives, or are not representatives of the canonical snoRNAs (like the snoRNAs in leishmania). In this case, `snoReport` 2.0 could discard these candidates. Since snoReport was trained with vertebrate sequence, snoRNAs in invertebrates could not be detected efficiently by snoReport. To deal with some of this organisms, it is necessary to discover new features that describe those non standard snoRNAs and use particular datasets in machine learning tasks. However, We find 69,64% and 76,67% of H/ACA box snoRNAs of nematodes and drosophilids found in literature, which suggests that H/ACA box snoRNA predictor from snoReport can be used with high performance.

Therefore, `snoReport` 2.0 was improved and is now more efficient to identify both classes of snoRNAs, and can be used for many different organisms, even in some invertebrates, with high quality of prediction.

4.5 Conclusion

In this article, we presented `snoReport` 2.0, a reliable and efficient tool to predict the two main classes of snoRNAs in different organisms. This version is a refinement of a previous version of `snoReport`, obtained with extensive improvements in the SVM method, and the use of new versions of tools (specially those to predict secondary structures) and databases. In contrast to previous methods for snoRNA identification, `snoReport` 2.0 can identify both guide and orphan snoRNAs without using any information of putative target sites within ribosomal or spliceosomal RNA or using multiple alignments. Experiments with very different organisms have shown good performance, even in invertebrates organisms (for H/ACA box snoRNA), showing that `snoReport` 2.0 can be used to obtain reliable prediction of snoRNAs in a variety of organisms.

Future work include creating specific datasets for different kinds of organisms (e.g, for invertebrates), and studying at what extent different approaches to fold the sequences and different machine learning methods (e.g., using EDeN to transform the secondary structure of snoRNAs in a graph representation, that can be decomposed in a sparse vector, allowing us to discover intrinsic features or, even, discovery new snoRNAs). The

use of these techniques could affect the performance of `snoReport` 2.0. Our method could also be used to identify snoRNAs in specific species, e.g., fungi (*Paracoccidioides brasiliensis*, *Schizosaccharomyces pombe* and *Pichia pastoris*), or to find specific features and perform a SVM training to identify snoRNAs in Leishmania. Finally, a general method could be developed to allow SVM training with particular organisms, according to user needs.

4.6 Availability and requirements

- **Project Name**: SnoReport v2.0;
- **Project home page**: http://www.biomol.unb.br/snoreport;
- **Operation system(s)** Linux;
- **Programming language** C ansi;
- **Other requirements**: Vienna RNA Package v2.1.5 (particularly RNAfold, RNALFold and RNAz);
- **License**: GNU GPL
- **Any restriction to use by non-academics**: No restrictions

Capítulo 5

SnoRNA-EDeN

Neste capítulo, apresentamos o *snoRNA-EDeN*, a ser submetido a um periódico.

5.1 Introduction

Identifying and classifying non-coding RNA genes (ncRNA genes) are still challenging, since researchers have been continuously discovering new and important functions in the cell, e.g., structural, catalytic and regulatory functions [25, 34, 58]. It is difficult to experimentally confirm the functions performed by a ncRNA, as well as to propose computational methods to identify and classify ncRNAs. It is also known that ncRNA functions are closely associated to their spatial (secondary) structures, which can be predicted from their nucleotide sequences (primary structures). Computational methods aim to predict candidates which have to be experimentally confirmed.

In particular, snoRNAs are 60 to 300 nt ncRNAs that accumulate in the nucleolus. They are classified based on their characteristic sequence elements, called *boxes*, in two main classes: H/ACA box snoRNAs and C/D box snoRNAs. In humans they are usually found in intronic regions [25]. Figures 5.1 and 5.2 shows secondary structures of H/ACA box and C/D box snoRNAs.

Among others [95, 75, 55, 5], an important method used to identify both classes of snoRNAs in single sequences is `snoReport` [34]. This tool uses a combination of secondary structure prediction and machine learning algorithm (Support Vector Machine - SVM). In contrast to previous methods for snoRNA identification (except snoSeeker [95]), the prediction done by snoReport does not use information of putative target sites within ribosomal or spliceosomal RNA (this information can dramatically improve identification sensibility and specificity). However, many orphan snoRNAs have been discovered with the `snoReport` approach. The target(s) of orphan snoRNAs are not known, consequently such genes would be missed on identification methods using target information [34, 43].

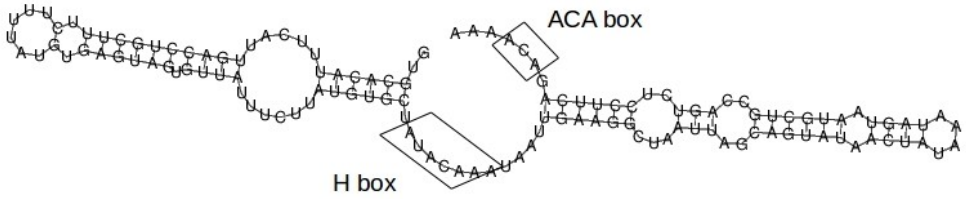


Figura 5.1: Secondary structure of H/ACA box snoRNA.

Beyond this, some of snoRNAs are shown to target specific mRNAs, suggesting other functions, e.g., interference with A-to-I editing [87, 34, 43, 25, 45].

Machine learning methods such as used on `snoReport` have been widely used on identification and classification of different families of ncRNAs [34, 95, 94, 86, 63]. Many of these methods are based on supervised learning, where some previous known attributes, called features, are collected from a sequence and then used in a classifier.

A recent approach in machine learning is described as follows. Given a region of interest of a sequence, the objective is to generate a sparse vector that can be used as micro-features in a specific machine learning algorithm, or it can be used to create powerful features (created from micro-features) on previous methods. One method that uses this approach is EDeN. Explicit Decomposition with Neighbourhoods (EDeN) is a decompositional graph kernel based on Neighborhood Subgraph Pairwise Distance Kernel (NSPDK) [15], which transforms one graph in a sparse vector, decomposing it into all pairs of neighborhood subgraphs of small radius at increasing distances. This sparse vector can be used as features, in both machine learning supervised and unsupervised learning tasks.

In this work, we present a new method based on EDeN to identify the two main classes of snoRNAs, C/D box and H/ACA box snoRNAs: transforming specific secondary structure regions of snoRNA in a graph representation, used to build sparse vectors that can be used on different machine learning algorithms, e.g., stochastic gradient descent (SGD).

This text is organized as follows. In the first section, we describe the methods used for building C/D box and H/ACA box finder using EDeN, particularly, data sources,

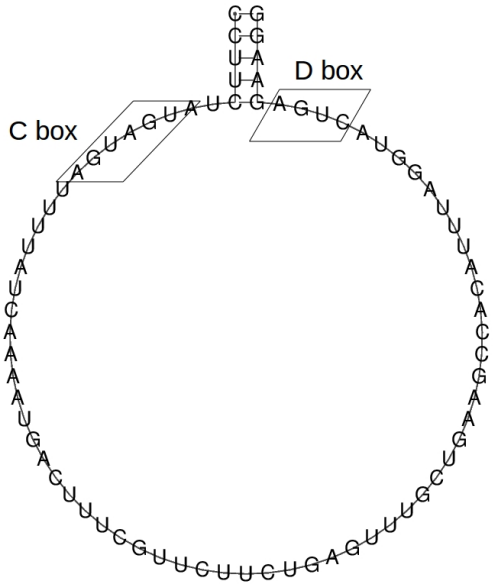


Figura 5.2: Secondary structure of C/D box snoRNA.

software components, workflows and details of the training phase. Next, we discuss the results obtained using these approach. Finally, we conclude and suggest future work.

5.2 Methods

First, data sources, as well as software components used to build C/D and H/ACA finder will be described. Next the workflows used to build them are shown.

5.2.1 Data sources

Since we are going to use a machine learning approach, data used for the training and test phases were divided in two sets: positive samples and negative samples, each having the two classes of snoRNAs. The positive sample set was composed of H/ACA box and C/D box snoRNAs, while the negative one was obtained from a dinucleotide shuffling procedure executed in the positive samples with the proper EDeN [15] library.

The positive sequences from each class of snoRNAs were divided in two datasets, to be used in the learning process. In order to avoid overfitting, these datasets were created such that very similar sequences would not be stored in different datasets (figure 5.3).

First, we clustered the sequences using ClustalW [84] with the criterion *nucleotide similarity*, which generated 157 clusters for C/D box snoRNA and 101 clusters for H/ACA box snoRNA. After, 10 sequences from distinct vertebrates organisms were extracted from each cluster, noting that clusters containing less than 10 sequences were discarded.

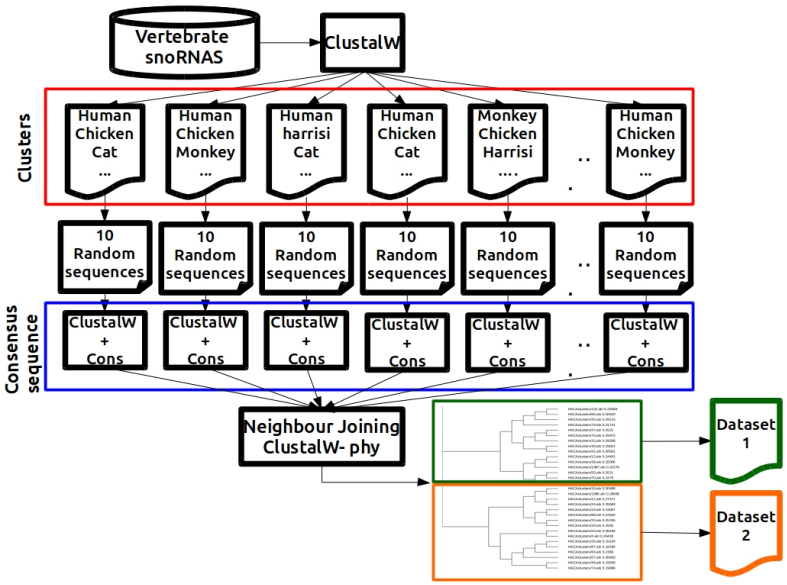


Figura 5.3: Pipeline used to create two snoRNA datasets such that very similar sequences are not stored in different datasets.

Therefore, a consensus sequence from each cluster was obtained with ClustalW and Cons (for EMBOSS [72]). After, these sequences were used to generate a distance tree, with the neighbour-joining method [74] from ClustalW2 - phylogeny [59]. The trees for H/ACA box and C/D box snoRNAs are shown in Annexes I and II, respectively. The next step was to divide this distance tree in two parts, which allowed to create the two datasets not containing similar sequences between them. To maintain proportionality between the number of sequences on each dataset, on C/D box snoRNA, the dataset 1 has only 7 sequences per cluster, and dataset 2 has 10 sequences per cluster. Table 5.1 shows the number of sequences of each dataset.

Tabela 5.1: Number of sequences of Datasets 1 and 2 of both C/D box and H/ACA box snoRNAs.

	Dataset 1	Dataset 2
C/D box snoRNAs	450	220
H/ACA box snoRNAs	490	420

5.2.2 Software Components

RNA secondary structure prediction was performed using Vienna RNA Package, current version 2.1.9, in particular RNAsubopt [35]. RNAsubopt calculates all the suboptimal

secondary structures from a sequence, within a user defined energy range above the minimum free energy (mfe). The input sequence, can contain a special character ' & ', which connects two specific regions of the sequence to form a complex.

LibSVM version 3.20 [10] implements the Support Vector Machine (SVM) used on both the training and test phases of HACA finder.

The EDeN library [15] is a Python package, extensively used in this work, since it includes many functionalities for bioinformatics and machine learning issues. These functionalities include:

- RNA visualization and secondary structure conversion: the secondary structure of a sequence is taken using many different secondary structure predictions methods, e.g., RNAfold and RNAsubopt, and it is transformed in a graph representation that can be visualized and used as samples in machine learning tasks;
- Fasta file manipulations: there are some functions that simplify the use of FASTA files;
- Integration with scikit-learn library [68]: scikit-learn is a powerful machine learning library, used in this work to perform the training and test phases using Stochastic Gradient Descent (SGD), and to calculate the performance measures: accuracy, F-score, average precision, ROC AUC score and Residual sum of squares (RSS);
- Meta-parameter optimizer: it has some functions that helps to find good values for meta-parameters in machine learning algorithms;
- Negative dataset generator: given a sequence, it performs a dinucleotide shuffling procedure on it, allowing to create our negative dataset, containing sequences with the same conservation of nucleotides found in snoRNAs;

Using these software components, the C/D finder and H/ACA finder were entirely written in the Python language.

5.2.3 Identifying snoRNA sequences

As said before, both classes of snoRNAs, H/ACA box and C/D box, can be distinguished by their characteristic boxes, and some specific secondary structure features. For this, each class of snoRNA has a specific way to searching for candidates. The main idea for both classes is to transform a particular region of the snoRNA secondary structure in a graph, use EDeN to build a sparse vector, and use it in a machine learning algorithm. We built two scripts, C/D finder for C/D box snoRNA prediction and H/ACA finder for H/ACA box snoRNA prediction. In order to identify the boxes C, D, H and ACA from snoRNAs, we developed C, D, H and ACA finder.

Box finder

Instead of using PWMs as used on snoReport 2.0, we used EDeN to predict the boxes C, D, H and ACA.

The dataset used for a box prediction consists at all the boxes on the snoRNA clusters, since we would like to predict all the boxes. If some false positive was introduced, the C/D and H/ACA finder should detect it by looking for other characteristics in the secondary structure.

Samples used to predict the boxes is formed by: *3nt before the box + dummy nt X + box sequence + dummy nt Y + 3nt after the box.* The samples were transformed in linear graphs, then transformed in a sparse vector with EDeN, and finally submitted to a SGD estimator. Figure 5.4 shows these samples.

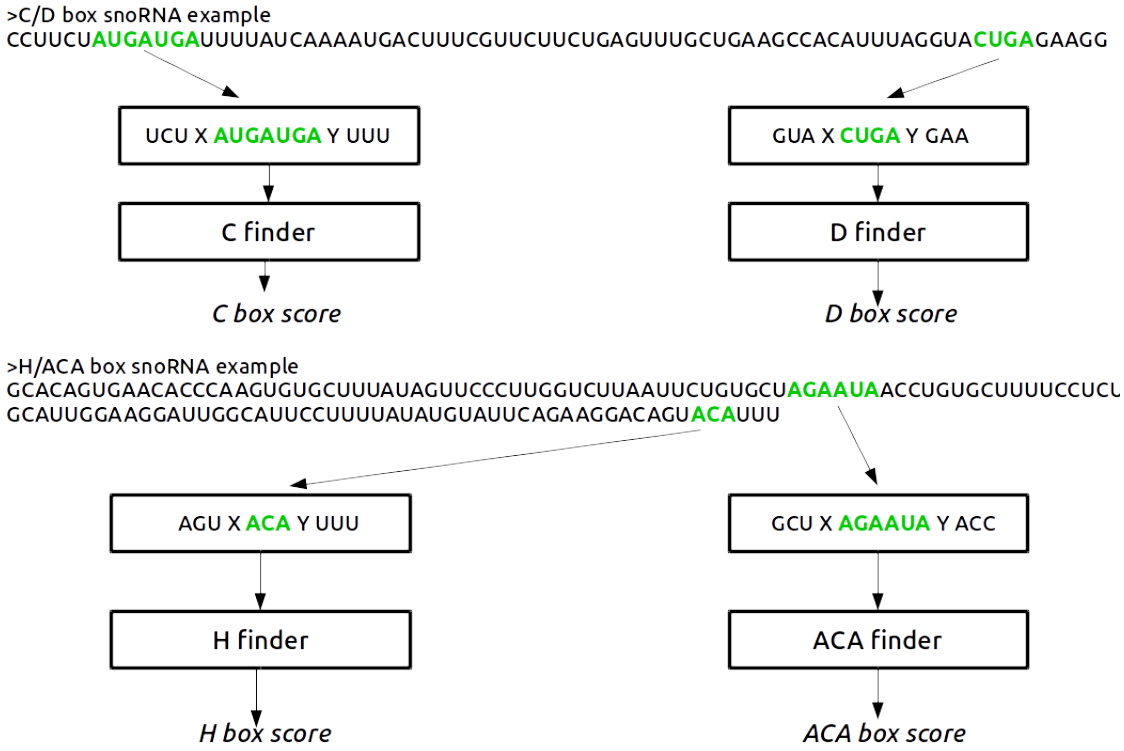


Figura 5.4: Example of samples used on box finder for C/D box and H/ACA box snoRNAs.

The dummy nucleotides X and Y were used to indicate the start and end position of a box, and it helps EDeN to better generate good *micro-features* to increase the estimator performance.

Data was randomly split on 70% as training dataset and 30% as test dataset. In order to improve the results, some meta-parameters from EDeN and SGD were optimized, using

a range of possible values for each meta-parameter in a proper function of the EDeN library.

C/D finder

The searching and prediction of C/D box snoRNAs in a genome sequence was performed with the following steps:

- All the combinations of C and D boxes with C and D finder is found in a single sequence with a maximal distance of 200 nt between then;
- For every pair of C and D boxes,a sample to be used on RNAsubopt is created. This sample consists at: *a maximum of 10 nt before the C box, the C box, a special character '&', the D box, and a maximum of 10 nt after D box*;
- The sample is submitted to RNAsubopt, and then transformed into a set of graphs (related with the number of suboptimal structures), using an EDeN routine called *RNAsubopt_to_eden*;
- The samples are sent to a model generated using Stochastic Gradient Descent (SGD), which decides if these sequence are (or not) a C/D box snoRNA.

The sample used as input to RNAsubopt contains a special character &, which connects the sequences that will form the short stem region of a C/D box. The pre-processor, responsible for running RNAsubopt and transform its outputs in graphs, was optimized using a function in the EDeN library that uses a different combination of parameter values on RNAsubopt. In this case, we used the following parameters:

- *Energy range*$(3\% - 10\%)$: computes suboptimal structures with energy in a certain range of the optimum secondary structure;
- *max_num_subopts* $(100 - 200)$: limits the maximum number of suboptimal secondary structures to be extracted from a sequence, using RNAsubopt;
- *max_num* $(3 - 8)$: from the suboptimal secondary structure, *max_num* structures were chosen with a maximum difference between them.

On the learning phase, we made two trainings and test phases, first, using dataset 1 as training and dataset 2 as test, and vice-versa. To make the negative dataset, we used a shuffled dinucleotide procedure on the positive dataset. In order to fix good meta-parameters for the SGD model, we used an EDeN routine responsible to optimize meta-parameters, informing a range of values for each meta-parameter. This routine itself discovers which combination of those meta-parameters is best for this model (using 10-fold cross validation).

H/ACA finder

The searching and prediction of H/ACA box snoRNAs candidates in a genome sequence was performed with the following steps:

- The genome sequence is scanned in order to find all the combinations of H and ACA boxes using H and ACA finder with a distance from 40 to 120 nt between then;
- 200 nt before a H box (hairpin 1 candidate) is submitted to RNAsubopt, and then transformed in a graph representation;
- Then, these graphs of hairpin 1 regions are transformed in a sparse vector, and used in a model generated by SGD (trained by using known hairpin 1 regions for snoRNAs), returning a score for each sample;
- After, the same is done for the region between H box and ACA box (hairpin 2) in order to obtain a score for each hairpin 2 sample;
- Finally, we used these 4 scores (H and ACA scores, hairpin 1 and hairpin2 scores) in a SVM model that returns if the candidate is (or not) a H/ACA box snoRNA .

In order to obtain scores for hairpin 1 and hairpin 2, two scripts were developed to train SGD models. We used the same ideas as used in the C/D finder, to obtain good meta-parameters, i.e, we tested different combinations of parameters from the pre-processor (RNAsubopt to graphs), EDeN and meta-parameters from the SGD. For these trainings, we also made one model from Dataset 1 as training and Dataset 2 as test, and vice versa.

In the final step, we used the previous generated models to create the feature vector containing four scores, to be used on the SVM to generate a model that predicts H/ACA box snoRNAs. For the SVM training, we performed a grid search for the C and γ parameters, using *grid.py* (available in libSVM v3.20), a parameter selection tool for C-SVM classification using the RBF (radial basis function) kernel. It uses a cross validation technique (in our case, 10-fold) to estimate the accuracy (another criteria could be used as well) of each combination of C and γ in the specified range, which allowed to choose the best values. Following Hsu [37], "a practical method to identify good parameters is to try exponentially growing sequences of C and γ". Therefore, we first investigated all the combinations of these two parameters ranging both from 2^{-15} to 2^{15}, shifting 2^1 for each step of the grid-search (for example, $2^{-15}, 2^{-14}, ...2^{15}$).

For a more refined analysis, we used the scikit-learn library [68], which allowed to obtain three types of performance measures to better evaluate and compare the our approach with snoReport 2.0:

- Fixed threshold (Accuracy and F-score): a sample is classified as positive if its score (or probability) is above a certain fixed threshold;
- Dynamic threshold (Average precision - APR - and Area Under the Curve - AUC): these measure is based on moving thresholds along the positive class. It returns the area under the precision-recall curve (APR) and the area under the ROC curve (AUC);
- Residual Sum of Squares: shows the discrepancy between data and an the estimator model.

5.3 Results and discussion

To identify H/ACA box and C/D box snoRNAs, we built two different datasets for each class of snoRNAs. For the learning phases, we used one dataset as training and the other for test (and vice-versa). Each training was repeated 10 times, and our results showed the average of the obtained results and their corresponding standard deviation. In this section, we show the results for C/D finder and H/ACA finder, compare then with snoReport 2.0 and apply C/D finder and H/ACA finder to a set of previously identified snoRNAs in literature.

5.3.1 C/D finder

Table 5.2 shows the results obtained for C/D finder test phase.

Tabela 5.2: Test phase results for C/D box snoRNA: Accuracy (Acc), F-score (F-SC), Average Precision (APR), Area under the ROC curve (AUC) and Residual Sum of Squares (RSS). Dat1 and Dat2 means Dataset 1 and Dataset 2, respectively.

	ACC (%)	FSC (%)	APR (%)	AUC (%)	RSS
Dat1→Dat2 average	97.77	94.2	99.14	99.79	0.019
Standard deviation	0.437	1.255	0.272	0.088	0.004
Dat2→Dat1 average	97.18	92.5	98.4	99.37	0.025
Standard deviation	0.934	2.632	0.494	0.134	0.008
All trainings' average	97.475	93.350	98.770	99.580	0.022
Standard deviation	0.772	2.188	0.543	0.242	0.007

In order to compare our approach with snoReport 2.0, we used the same datasets for training and test phases. Table 5.3 shows the results for C/D box snoRNA using snoReport 2.0.

Tabela 5.3: Test phase results for C/D box snoRNA identification with snoReport.

	Acc (%)	FSC (%)	APR (%)	AUC (%)	RSS
Dat1 average→Dat2	94.37	93.67	98.43	98.82	0.044
Standard deviation	1.65	2.04	0.77	0.51	0.012
Dat2 average→Dat1	96.19	94.94	98.80	99.11	0.029
Standard deviation	0.90	1.25	0.53	0.63	0.007
All trainings' average	95.28	94.30	98.61	98.96	0.037
All trainings' SD	1.60	1.77	0.67	0.58	0.012

We can see that C/D finder produced an equivalent result, when compared to snoReport 2.0, inducing micro-features from the terminal stem secondary structure region of a C/D box snoRNA, instead of using known features, like snoReport 2.0. This shows that EDeN generates a good set of features, only looking for the structure of the data. Futhermore, these results show that the terminal stem region of the C/D box is an important region of the secondary structure, which could be used in order to detect an entire C/D box snoRNA sequence.

5.3.2 H/ACA finder

Table 5.4 shows the results obtained in the H/ACA finder test phase.

Tabela 5.4: Test phase results for H/ACA box snoRNA prediction using EDeN: Accuracy (Acc), F-score (F-SC), Average Precision (APR), Area under the ROC curve (AUC) and Residual Sum of Squares (RSS). Dat1 and Dat2 means Dataset 1 and Dataset 2, respectively.

	Acc (%)	FSC (%)	APR (%)	AUC (%)	RSS
Dat1→Dat2 average	94.61	84.65	96.55	98.65	0.043
Standard deviation	1.104	3.718	0.938	0.510	0.011
Dat2→Dat1 average	94.90	85.59	96.70	98.68	0.041
Standard deviation	1.008	3.180	1.357	0.520	0.009
All trainings' average	94.75	85.12	96.62	98.67	0.042
Standard deviation	1.042	3.423	1.109	0.499	0.010

In order to compare our approach with snoReport 2.0, we used the same datasets for training and test phases. Table 5.5 shows the results for H/ACA box snoRNA using snoReport 2.0.

H/ACA finder produced good results on test phase, having a close performance with snoReport 2.0. The features have been extracted from hairpin secondary structure regions and regions where the box happens using EDeN and, then, used in a SVM model. This

Tabela 5.5: Test phase results for H/ACA box snoRNAs on snoReport 2.0: accuracy (Acc), F-score (F-SC), Average Precision (APR), Area under the ROC curve (AUC) and Residual Sum of Squares (RSS). Dat1 and Dat2 means Dataset 1 and Dataset 2, respectively, and SD means *standard deviation*.

	Acc (%)	FSC (%)	APR (%)	AUC (%)	RSS
Dat1 average→Dat2	97.31	93.07	97.85	98.94	0.022
Standard deviation	0.24	0.60	0.20	0.20	0.002
Dat2 average→Dat1	97.43	94.71	98.66	99.33	98.66
Standard deviation	0.51	1.06	0.42	0.20	0.004
All trainings' average	97.37	93.89	98.25	99.14	0.021
All training' SD	0.39	1.19	0.53	0.28	0.003

shows that EDeN is a good method to build macro-features from micro-features extracted from different parts of a molecule, that can be used on a set of different machine learning algorithms.

5.4 Validation on real data

In order to verify the capacity of generalization of our *snoRNA-EDeN*, a validation on real data was performed. We executed *snoRNA-EDeN* with a set of previously predicted vertebrate and invertebrate sequences, some of them partially confirmed in experiments in humans, nematodes, drosophilids, platypus, chickens and leishmania. Tables 5.6 and 5.7 shows the summary of these results in vertebrates and invertebrates organisms, respectively.

Tabela 5.6: Results of executing *snoRNA-EDeN* with snoRNA sequences of vertebrate organisms. The number of predicted candidates on each reference is shown in the following order: predicted with *snoRNA-EDeN*, predicted with snoReport 2.0, and predicted by the corresponding reference.

Human		
Yang et al. [95]	C/D: 21/21/21	H/ACA: 30/28/32
Platypus		
Schmitz et al. [76]	C/D: 133/42/144	H/ACA: 69/45/73
Chicken		
Shao et al [79]	C/D: 127/112/132	H/ACA: 67/66/69

Tabela 5.7: Results of executing *snoRNA-EDeN* with snoRNA sequences of invertebrate organisms. The number of predicted candidates on each reference is shown in the following order: predicted with *snoRNA-EDeN*, predicted with snoReport 2.0, and predicted by the corresponding reference.

Nematodes		
Zemann et al. [97]	C/D: 51/32/108	H/ACA: 32/46/60
Drosophilids		
Huang et al. [38]	C/D: 52/2/63	H/ACA: 35/39/56
Leishmania		
Liang et al. [50]	C/D: 45/0/62	H/ACA *A-like*: 0/0/37

Yang et al. [95] identified 54 snoRNAs, 21 C/D box and 32 H/ACA box in human, using snoSeeker, a method based on probabilistic models, pairwise whole-genome alignments (WGS) of eukaryotes, in which the user can include information of the putative target region or not (to find orphan snoRNAs). For C/D box snoRNA, both *snoRNA-EDeN* and snoReport 2.0 predicted all the C/D box snoRNAs cited on this reference.For H/ACA box snoRNA, *snoRNA-EDeN* find 30 out of 32 while snoReport 2.0 predicted 28 H/ACA box snoRNAs.

Schmitz et al. [76] identified 166 individual snoRNAs in a platypus brain cDNA library, generated from small non-protein-coding RNAs. After, using BLAST searches in platypus genomic sequences, they found 51 more sequences of snoRNA. Futhermore, they found cis- and trans-duplication distribution patterns for snoRNAs, which had not been described in other vertebrates, but only in nematodes. For C/D box snoRNAs, *snoRNA-EDeN* predicted 133 out from 144 C/D box snoRNAs (92.4%), while snoReport 2.0 only detected 42 C/D box snoRNAs (29.2%). For H/ACA box snoRNAs, *snoRNA-EDeN* predicted 69 out of 73 H/ACA box snoRNAs (94.5%), while snoReport 2.0 detected 45 H/ACA box snoRNAs (61.6%).

Shao et al. [79] identified 132 C/D box snoRNAs in chicken using *CDseeker* and 69 H/ACA box snoRNAs using *ACAseeker* (both programs are used in snoSeeker [95]). For C/D box snoRNA, we predicted 127 out of 132 C/D box snoRNAs (96,2%), while snoReport 2.0 predicted 112 C/D box snoRNAs (84.9%). For H/ACA box snoRNAs, we predicted 67 out of 69 H/ACA box snoRNAs (97.1%), while snoReport 2.0 predicted 66 H/ACA box snoRNAs (95.7%).

Zemann et al. [97] used a combination of high-throughput cDNA library screening and computational search strategies to find 121 snoRNAs (168 are shown in the supplementary material) in *Caernorhabditis elegans*. For C/D box snoRNAs, *snoRNA-EDeN* predicted

51 out of 108 C/D box snoRNAs (47.2%), while snoReport 2.0 predicted 32 C/D box snoRNAs (34.3%). For H/ACA box snoRNAs, *snoRNA-EDeN* predicted 32 out of 60 H/ACA box snoRNAs (53.3%), while snoReport predicted 46 H/ACA box snoRNAs (77.7%).

Huang et al. [38] performed a large-scale genome wide analysis (WGS) to identify both classes of snoRNAs in *Drosophila melanogaster* using experimental and computational RNomics methods, having found 119 snoRNAs. For C/D box snoRNAs, *snoRNA-EDeN* predicted 52 out of 63 C/D box snoRNAs (82.5%) while snoReport 2.0 predicted only 2 C/D box snoRNAs (3.2%). For H/ACA box snoRNA, *snoRNA-EDeN* predicted 35 out of 56 H/ACA box snoRNAs (62.5%), while snoReport 2.0 predicted 39 H/ACA box snoRNAs (69.6%).

Finally, Liang et al. [50] used a genome-wide screening approach to identify 62 C/D box snoRNAs and 37 H/ACA box snoRNAs of closely related pathogens of *Leishmania major*. For C/D box snoRNA, *snoRNA-EDeN* identified 45 out of 62 C/D box snoRNAs (72.6%) while snoReport 2.0 did not identified any C/D box snoRNAs. Like snoReport 2.0, *snoRNA-EDeN* did not identify any H/ACA box snoRNA. It is note worthy that H/ACA box snoRNAs from *Leishmania major* are quite different from the canonical H/ACA box snoRNAs of yeast and vertebrate. For example, they lack a recognizable H box, presenting an AGA box instead of an ACA box [34].

The validation phase performed on C/D finder showed that our new method to identify C/D box snoRNAs have a high capacity, of generalization,predicting 94.61% of all vertebrate C/D box snoRNAs and 63.52% of all invertebrate C/D box snoRNAs, a better result when compared with snoReport 2.0, which predicted only 52.92% of vertebrates and 14.6%. Probably, extracting features directly from the structure itself, than using known features, helped C/D finder to obtain better results on the validation phase, allowing our new method to not become too specific to a set of canonic features. It is interesting that this method allowed us to predict many drosophilids and leishmania C/D box snoRNAs, showing again, the high capacity of generalibilization of the EDeN method.

The validation phase performed on H/ACA box showed that our new method to identify H/ACA box snoRNA have a very a high prediction rate on vertebrate snoRNAs. Even presenting a lower F-score compared to snoReport 2.0, snoRNA-EDeN was capable to identify 95.4% of all the vertebrate snoRNAs, while snoReport 2.0 predicted 79.9%. For the invertebrate dataset, snoRNA-EDeN predicted 57.8% of Nematode and Drosophila sequences, while snoReport 2.0 predicted 73.3%. This suggests that snoRNA-EDeN, combined with snoReport 2.0, can be used to better validate putative H/ACA box snoRNAs in invertebrates.

5.5 Conclusion

In this work, we presented a new method, called *snoRNA-EDeN*, based on EDeN [15], to identify the two main classes of snoRNAs. The general idea of the method is to transform specific secondary structure regions of a snoRNA in a graph representation, used to build sparse vectors, which can be used on different machine learning algorithms, in our case, stochastic gradient descent (SGD).

For the C/D box snoRNA classifier, in the test phase, the results of snoRNA-EDeN showed equivalent results, when compared to snoReport 2.0. In the validation phase, it presented a strong capacity of generalization, allowing to effectively identify both vertebrate and invertebrate C/D box snoRNAs.

The H/ACA box snoRNA classifier presented close results when compared with snoReport 2.0. For vertebrate snoRNAs on the validation phase, *snoRNA-EDeN* showed a very good performance. For invertebrate snoRNAs, The results were close to snoReport 2.0, suggesting that both can be used together, in order to get more reliable putative H/ACA box snoRNAs.

Next steps include: identifying better regions on H/ACA box snoRNAs to be used on snoRNA-EDeN; using secondary structure prediction besides RNAsubopt, e.g., RNAshapes and RNALfold; and making *snoRNA-EDeN* available on Galaxy workflow [27].

Capítulo 6

Conclusão

Nesta dissertação, foram desenvolvidos dois métodos computacionais de identificação de *snoRNAs*, usando técnicas de aprendizado de máquina: snoReport 2.0 e o *snoRNA-EDeN*. No snoReport 2.0, foram extraídos novas *features* de ambas as classes de *snoRNAs* e utilizada uma técnica mais sofisticada na fase de treinamento, adotando uma nova abordagem para encontrar bons meta-parâmetros da SVM, além de usar dados recentes de vertebrados. No *snoRNA-EDeN*, regiões específicas de estruturas secundárias de *snoRNAs* foram identificadas e representadas em grafos, que foram transformados em vetores esparsos e usados no algoritmo de gradiente descendente estocástico (SGD) para gerar: um classificador de *C/D box snoRNAs*; e *features* poderosas de *H/ACA box snoRNAs*, usadas para gerar um classificador SVM de *H/ACA box snoRNAs*.

Ambos os métodos foram bastante eficazes na identificação de *snoRNAs*, tanto na fase de testes, quanto na fase de validação. O snoReport 2.0 é uma ferramenta com grande potencial para a identificação de *snoRNAs* que possuem características canônicas, especialmente em organismos vertebrados. Entretanto, seu resultado não foi tão satisfatório em organismos invertebrados, possivelmente devido características diferentes das canônicas, encontradas nesses *snoRNAs*. Além disso, o snoReport ainda obteve bons resultados na predição de *H/ACA box snoRNAs* de nematóides e drosófilídeos, podendo então ser usado na predição de *snoRNAs* nesses organismos.

O *snoRNA-EDeN* mostrou uma grande eficácia na identificação de *C/D box snoRNAs*, tanto canônicos, quanto não canônicos, o que possivelmente pode ser explicado pela geração de *micro-features* a partir da estrutura de dados utilizada (no caso, representação em grafos da estrutura secundária). Portanto, a predição de *C/D box snoRNAs* com o *snoRNA-EDeN* teve um desempenho bastante significativo, tanto na fase de testes, quanto na validação, o que envidenciou que a região da haste terminal (*terminal stem*) é bastante importante na identificação desse tipo de molécula, visto que não foram utilizadas características da sequência do loop interno (a não ser dos próprios boxes). Com relação

a *H/ACA box snoRNAs*, o snoRNA-EDeN apresentou ótimos resultados para organismos vertebrados, enquanto que, para organismos invertebrados, o snoRNA-EDeN identificou 57.8% de nematóides e drosófilas, comparados aos 73.3% identificados pelo snoReport 2.0.

6.1 Contribuições

Este trabalho teve como contribuições:

- Construção do snoReport 2.0. Esta nova versão foi inteiramente reescrita em C, apresentando mudanças significativas em todo processo de busca de candidatos, filtros de estruturas secundárias e terciárias, extração de *features* e na fase de aprendizagem de máquina. Além disso foram usadas sequências de vertebrados, da qual foram cuidadosamente selecionados de forma a não causar viés na fase de treinamento;
- Construção do snoRNA-EDeN, um poderoso programa capaz de extrair explicitamente *micro-features* de regiões de uma sequência de *snoRNA* através do EDeN, que foram usadas para identificar *C/D box snoRNAs* e formar *macro-features* para a identificação de *H/ACA box snoRNAs* usando SVM. Além disso, para *C/D box snoRNA* revelou a possibilidade de se identificá-los apenas usando a informação do terminal *stem*;
- Apresentação oral no ISCB-LA 2014 do artigo: *New features and refined SVM improve snoRNA identification in snoReport*;
- Apresentação oral no X-meeting+BSB 2015 (atualmente em processo de análise para publicação na *BMC Bioinformatics*) do artigo: *SnoReport 2.0: new features and a refined Support Vector Machine improve snoRNA identification*;
- Apresentação de poster no X-meeting+BSB 2015 do trabalho: *Identification of snoRNAs using EDeN*, que recebeu menção honrosa (dada aos três melhores trabalhos dos eventos). O artigo será submetido para revista científica neste ano.

6.2 Trabalhos Futuros

As perspectivas deste trabalho são:

- Para o snoReport 2.0:
  - Desenvolver versão paralela;

  - Identificar melhores *features*, a fim de melhorar a capacidade de generalização do método;
  - Desenvolver classificadores para organismos (ou grupos de organismos) específicos, como a *Leishmania e Platypus*;

- Para o snoRNA-EDeN:
  - Diminuir consumo de memória e tempo;
  - Identificar melhores regiões, ou melhores algoritmos de predição de estrutura secundária, a serem usados para aprimorar o desempenho na identificação de *H/ACA box snoRNAs*;
  - Disponibilizar o snoRNA-EDeN no *Galaxy* [27], um *workflow* para aplicações de bioinformática.

Referências

[1] O código genético. SO Biologia. http://www.sobiologia.com.br/conteudos/Citologia2/AcNucleico6.php Acessado em 02/12/2015. xii, 15

[2] S. F. Altschul et al. Basic Local Alignment Search Tool. *Journal of Molecular Biology*, 215(3):403–410, 1990. 24, 27

[3] W. C. Arruda. *ncRNA-Agents: Anotação de RNAs não-codificadores baseado em Sistemas Multiagente*. Tese de Doutorado em Informática. Departamento de Ciência da Computação. Universidade de Brasília, 2015. 20

[4] T. Bailey e C. Elkan. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. In *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, pages 28–36. AAAI Press, 1994. 25

[5] S. Bartschat et al. snoStrip: a snoRNA annotation pipeline. *Bioinformatics*, 30(1):115–116, 2014. xii, xiii, 2, 22, 23, 27, 49, 50, 61

[6] S. Ben-david et al. Limitations of learning via embeddings in euclidean half spaces. *Journal of Machine Learning Research*, 3:441–461, 2002. 38

[7] D. A. Benson et al. Genbank. *Nucleic acids research*, 37(Database issue):D26–31, 2009. 21

[8] J. Brown et al. Plant snoRNA database. *Nucleic Acids Research*, 31(1):432–435, 2003. 28

[9] S. W Burge et al. Rfam 11.0: 10 years of RNA families. *Nucleic Acids Research*, 41(Database-Issue):226–232, 2013. 17, 21

[10] C. C. Chang e C. J. Lin. LIBSVM: a library for Support Vector Machines, 2001. Software available at: http://www.csie.ntu.edu.tw/ cjlin/libsvm/. 4, 36, 47, 65

[11] P. Clote e R. Backofen. *Computational Molecular Biology: An Introduction*. John Willey & sons Ltd, 2000. 1, 7, 12, 14

[12] J. R. Cole. et al. The ribosomal database project (rdp-ii): previewing a new autoaligner that allows regular updates and the new prokaryotic taxonomy. *Nucleic Acids Research*, 31(1):442–443, 2003. 17

[13] L. J. Collins. Characterizing ncRNAs in human pathogenic protists using high-throughput sequencing technology. *Frontiers in Genetics*, 2(96), 2011. 2, 41

[14] G. M. Cooper. *The Cell - A Molecular Approach 2nd Edition*. Sunderland (MA): Sinauer Associates, 2000. 2

[15] F. Costa e K. D. Grave. Fast neighborhood subgraph pairwise distance kernel. In Stefan Wrobel, Johannes Fürnkranz, e Thorsten Joachims, editors, *Proceedings of the 26th International Conference on Machine Learning*, pages 255–262, 2010. 4, 37, 38, 39, 44, 62, 63, 65, 74

[16] R. Cotter. Bacterial genetics and growth. http://www.pc.maricopa.edu/Biology/rcotter/BIO%20205/LessonBuilders/Chapter%209%20LB/Ch9b_print.html. Acessado em 02/12/2013. xii, 13

[17] T. M. Cover. Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition. *Electronic Computers, IEEE Transactions on*, EC-14(3):326–334, 1965. 35

[18] M. M. Cox, J. A. Doudna, e M. O'Donnell. *Biologia Molecular. Princípios e Técnicas*. Artmed, 2013. 2

[19] F. Crick. Central Dogma of Molecular Biology. *Nature*, 227(5258):561–563, 1970. 1

[20] T. C. C. da Silva. *SOM-PORTRAIT: um método para identificar RNA não codificador utilizando Mapas Auto Organizáveis*. Monografia de Graduação. Departamento de Ciência da Computação. Universidade de Brasília, 2009. xiv, 17, 19, 21

[21] S. R. Eddy. Non-coding RNA genes and the modern RNA world. *Nature Reviews Genetics*, 2:919–929, 2001. 15

[22] R.C. Edgar. Muscle: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research*, 32(5):1792–1797, 2004. 28

[23] J. C. Ellis, D. D. Brown, e J. W. Brown. The small nucleolar ribonucleoprotein (snoRNP) database. *RNA*, 2010. 28

[24] Manel Esteller. Non-coding RNAs in human disease. *Nature Reviews Genetics*, 12:861–874, 2011. 41

[25] M. Falaleeva e S. Stamm. Processing of snoRNAs as a new source of regulatory non-coding RNAs. *BioEssays*, 35(1):46–54, 2013. 2, 3, 4, 13, 15, 22, 23, 41, 42, 49, 61, 62

[26] P. Gardner. Metazoan u3 secondary structure. http://en.wikipedia.org/wiki/File:Metazoan_U3_secondary_structure.png. Acessado em 02/12/2013. xii, 16

[27] J. Goecks et al. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biology*, 11(8):R86+, 2010. 74, 77

[28] S. Griffiths-Jones et al. mirbase: tools for microRNA genomics. *Nucleic Acids Research*, 36(Database-Issue):154–158, 2008. 22

[29] A. S. Grigory et al. Regulatory Role of Small Nucleolar RNAs in Human Diseases. *BioMed Research International*, Article ID 206849:1–10, 2015. 41

[30] A. R. Gruber et al. RNAz 2.0: Improved Noncoding RNA Detection. In *Pacific Symposium on Biocomputing*, pages 69–79, 2010. 45, 51

[31] M. Guttman, I. Amit, M. Garber, C. French, M.F. Lin, D. Feldser, M. Huarte, O. Zuk, B.W. Carey, J.P. Cassady, M.N. Cabili, R. Jaenisch, T.S. Mikkelsen, T. Jacks, N. Hacohen, B.E. Bernstein, M. Kellis, A. Regev, J.L. Rinn, e E.S. Lander. Chromatin signature reveals over a thousand highly conserved large non-coding RNAs in mammals. *Nature*, 458:223–227, 2009. 41

[32] D. Haussler. Convolution kernels on discrete structures. Technical Report UCS-CRL-99-10, University of California at Santa Cruz, 1999. 37

[33] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1999. xii, xiv, 33, 34, 36, 37

[34] J. Hertel, I. L. Hofacker, e P. F. Stadler. SnoReport: computational identification of snoRNAs with unknown targets. *Bioinformatics*, 24(2):158–164, 2008. 2, 3, 15, 19, 24, 25, 26, 41, 42, 58, 61, 62, 73

[35] I. L. Hofacker. Vienna RNA secondary structure server. *Nucleic Acids Research*, 31(13):3429–3431, 2003. 4, 45, 64

[36] I. L. Hofacker et al. Fast Folding and Comparison of RNA Secondary Structures. *Monatsh Chem*, 125:167–188, 1994. 45

[37] C. Hsu, C. Chang, e C. Lin. A practical guide to support vector classification, 2003. 54, 68

[38] Z. P. Huang, H. Zhou, H. L. He, C. L. Chen, D. Liang, e L. H. Qu. Genome-wide analyses of two families of snoRNA genes from Drosophila melanogaster, demonstrating the extensive utilization of introns for coding of snoRNAs. *RNA*, 11(8):1303–1316, 2005. 24, 45, 57, 58, 72, 73

[39] A. Hüttenhofer, M. Kiefmann, S. Meier-Ewert, J. O'Brien, H. Lehrach, J. P. Bachellerie, e J. Brosius. RNomics: an experimental approach that identifies 201 candidates for novel, small, non-messenger RNAs in mouse. *EMBO J*, 20(11):2943–2953, 2001. 24

[40] V. Jakkula. Tutorial on Support Vector Machine (SVM). School of EECS. Washington State University, 2006. 34

[41] W. E. F. Júnior e W. Francisco. Proteinas: Hidrolise, precipitacao e um tema para o ensino de quimica. http://qnint.sbq.org.br/qni/ visualizarConceito.php?idConceito=21 Acessado em 02/12/2013. xii, 12

[42] S. Kehr et al. PLEXY: efficient target prediction for box C/D snoRNAs. *Bioinformatics*, 27(2):279–280, 2011. 28

[43] S. H. Kim et al. Plant U13 orthologues and orphan snoRNAs identified by RNomics of RNA from Arabidopsis nucleoli. *Nucleic Acids Research*, 38(9):3054–3067, 2010. 3, 42, 61, 62

[44] T. Kin et al. fRNAdb: a platform for mining/annotating functional RNA candidates from non-coding RNA sequences. *Nucleic Acids Research*, 35(Database-Issue):145–148, 2007. 21

[45] S. Kishore e S. Stamm. The snoRNA HBII-52 regulates alternative splicing of the serotonin receptor 2C. *Science*, 311(5758):230–232, 2006. 3, 42, 62

[46] R. Klein, Z. Misulovin, e S. Eddy. Noncoding RNA genes identified in AT-rich hyperthermophiles. *Proc. Natl Acad. Sci.*, 56(99):7542–7547, 2002. 19

[47] E. S. Lander et al. Initial sequencing and analysis of the human genome. *Nature*, 409:860–921, 2001. 1, 15

[48] D. Leja. Transfer RNA (tRNA). http://www.genome.gov/dmd/img.cfm?node=Photos/Graphics&id=85250. Acessado em 02/12/2013. xii, 14

[49] L. Lestrade e M. J. Weber. snoRNA-LBME-db, a comprehensive database of human H/ACA and C/D box snoRNAs. *Nucleic Acids Research*, 34(suppl 1):D158–D162, 2006. 22, 28, 42

[50] X. H. Liang et al. A genome-wide analysis of C/D and H/ACA-like small nucleolar RNAs in *Leishmania major* indicates conservation among trypanosomatids in repertoire and in their rRNA targets. *Eukaryot. Cell*, 6:361–377, 2007. 45, 57, 58, 72, 73

[51] A. M. Lima, V. F. Onuchic, e A. M. Durham. Procura de padrões estruturais em RNA utilizando grafos. In *Curso de Verão 2011 - Bioinformática - USP*, pages 315–318. USP, 2011. 19

[52] C. Liu et al. NONCODE: an integrated knowledge database of non-coding RNAs. *Nucleic Acids Research*, 33(Database-Issue):112–115, 2005. 21

[53] M. Lluch et al. Defining a minimal cell: essentiality of small ORFs and ncRNAs in a genome-reduced bacterium. *Molecular Systems Biology*, 11(1):780, 2015. 41

[54] R. Lorenz et al. ViennaRNA Package 2.0. *Algorithms for Molecular Biology*, 6(1):1–14, 2011. 17, 20

[55] T. M. Lowe e S. R. Eddy. A Computational Screen for Methylation Guide snoRNAs in Yeast. *Science*, 283:1168–1171, 1999. 27, 61

[56] A. Nakao M. Yoshihama e N. Kenmochi. snopy: a small nucleolar rna orthological gene database. *BMC Research Notes*, 6(1):1–5, 2013. 29

[57] A. Machado-Lima. *Predição de RNAs não codificantes e sua aplicação na busca do componente RNA da telomerase*. Tese de Doutorado em Bioinformática. Universidade de São Paulo, 2006. xiv, 17, 19

[58] A. Machado-Lima et al. Computational methods in noncoding RNA research. *Journal of Mathematical Biology*, 56(1-2):15–49, 2008. xii, 2, 16, 17, 18, 19, 20, 21, 41, 61

[59] H. McWilliam et al. Analysis Tool Web Services from the EMBL-EBI. *Nucleic Acids Research*, 41(Web Server issue):W597–W600, 2013. 44, 64

[60] T. R. Mercer, M. E. Dinger, e J. S. Mattick. Long non-coding RNAs: insights into functions. *Nature reviews. Genetics*, 10(3):155–159, 2009. xiv, 17

[61] T. M. Mitchell. *Machine Learning*. McGraw-Hill ScienceEngineeringMath, 1997. 31, 32, 33

[62] A. W. Moore. Suport Vector Machines. Tutorial slides. School of Computer Science. Carnegie Mellon University. Available at: www.cs.cmu.edu/ awm, 2003. xii, 35

[63] E. P. Nawrocki e S. R. Eddy. Infernal 1.1: 100-fold faster RNA homology searches. *Bioinformatics*, 29(22):2933–2935, 2013. 2, 3, 62

[64] E. P. Nawrocki e S. R. Eddy. Infernal 1.1: 100-fold faster RNA homology searches. *Bioinformatics*, 29(22):2933–2935, 2013. 19, 20

[65] N. J. Nilsson. Introduction to machine learning: An early draft of a proposed textbook. pages 175-188. http://robotics.stanford.edu/people/nilsson/mlbook.ht ml, 1998. 31, 32

[66] Ruth Nussinov e Ann B. Jacobson. Fast Algorithm for Predicting the Secondary Structure of Single-Stranded RNA. *Proceedings of the National Academy of Sciences of the United States of America*, 77(11):6309–6313, 1980. 20

[67] K. C. Pang et al. RNAdb 2.0 - an expanded database of mammalian non-coding RNAs. *Nucleic Acids Research*, 35(Database-Issue):178–182, 2007. 21

[68] F. Pedregosa et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 40, 47, 54, 65, 68

[69] J. Pevsner. *Bioinformatics and Functional Genomics*. John Wiley & sons, inc, 2009. 7

[70] T. Phillips. sirna. biotech. http://biotech.about.com/od/glossary/g/siRNA.htm Acessado em 14/02/2014. xiv, 17

[71] J. M. Pipas e J. E. McMahon. Method for predicting RNA secondary structure. In *Proceedings of the National Academy of Sciences of the United States of America*, pages 2017–2021, 1975. 20

[72] P. Rice, I. Longden, e A. Bleasby. EMBOSS: the European Molecular Biology Open Software Suite. *Trends in genetics: TIG*, 16(6):276–277, 2000. 44, 64

[73] E. Rivas e S. R. Eddy. Secondary structure alone is generally not statistically significant for the detection of noncoding RNAs. *Bioinformatics*, 16(7):583–605, 2000. 19

[74] N. Saitou e M. Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4(4):406–425, 1987. 44, 64

[75] P. Schattner et al. Genome-wide searching for pseudouridylation guide snoRNAs: analysis of the *Saccharomyces cerevisiae* genome. *Nucleic Acids Research*, 32(14):4281–4296, 2004. 26, 61

[76] J. Schmitz et al. Retroposed SNOfall - A mammalian-wide comparison of platypus snoRNAs. *Genome Research*, 18:1005–1010, 2008. 45, 56, 57, 71, 72

[77] M. S. Scott et al. Human box C/D snoRNA processing conservation across multiple cell types. *Nucleic Acids Reseach*, 40(8):3676–3688, 2012. 22

[78] J. C. Setubal e J. Meidanis. *Introduction to Computational Molecular Biology*. PWS Publishing Company, Boston, MA, 2000. 1, 7, 8, 9, 10, 12, 14, 15

[79] P. Shao et al. Genome-wide analysis of chicken snoRNAs provides unique implications for the evolution of vertebrate snoRNAs. *BMC Genomics*, 10:86+, 2009. 45, 57, 58, 71, 72

[80] J. Shawe-Taylor e N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004. 37

[81] M. Szymanski, J. Barciszewski, e V. A. Erdman. *Noncoding RNAs: Molecular Biology and Molecular Medicine, chapter Riboregulators*. Springer, 2003. 1, 2, 15, 21, 22

[82] H. Tafer et al. RNAsnoop: efficient target prediction for H/ACA snoRNAs. *Bioinformatics*, 26(5):610–616, 2010. 28

[83] R. Taft et al. Small RNAs derived from snoRNAs. *RNA*, 15(7):1233–1240, 2009. 4

[84] J. D. Thompson, D. G. G. Higgins, e T. J. Gibson. Clustalw: improving the sensitivity if progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22:4673–4680, 1994. 44, 63

[85] J. C. Venter et al. The sequence of the human genome. *Science*, 291(5507):1304–1351, 2001. 1, 15

[86] P. Videm et al. BlockClust: efficient clustering and classification of non-coding RNAs from short read RNA-seq profiles. *Bioinformatics*, 30(12):i274–i282, 2014. 3, 62

[87] P. Vitali, E. Basyuk, E. Le Meur, E. Bertrand, F. Muscatelli, J. Cavaillé, e A. Huttenhofer. ADAR2-mediated editing of RNA substrates in the nucleolus is inhibited by C/D small nucleolar RNAs. *J Cell Biol*, 169(5):745–753, 2005. 3, 24, 42, 62

[88] S. Washietl et al. Structured RNAs in the ENCODE selected regions of the human genome. *Genome Res*, 17(6):852–864, 2007. 19, 51

[89] M. S. Waterman e T. F. Smith. RNA secondary structure: A complete mathematical analysis. *Math. Biosc*, 42:257–266, 1978. 19

[90] N. J. Watkins et al. A Common Core RNP Structure Shared between the Small Nucleolar Box C/D RNPs and the Spliceosomal U4 snRNP. *Cell*, 103(3):457 – 466, 2000. 22, 49

[91] J. D. Watson e F. H. C. Crick. Molecular struture of nucleic acids. *Nature*, 171(4356):737–738, 1953. 1

[92] Wikimedia. Difference DNA RNA. http://commons.wikimedia.org/wiki/File:Difference_DNA_RNA-EN.svg Acessado em 02/12/2013. xii, 9

[93] L. Xia, N. J. Watkins, e E. S. Maxwell. Identification of specific nucleotide sequences and structural elements required for intronic U14 snoRNA processing. *RNA*, 3(1):17–26, 1997. 22, 49

[94] J. Xie et al. Sno/scaRNAbase: a curated database for small nucleolar RNAs and Cajal body-specific RNAs. *Nucleic Acids Research*, 35(Database-Issue):183–187, 2007. 3, 28, 62

[95] J. Yang et al. snoSeeker: an advanced computational package for screening of guide and orphan snoRNA genes in the human genome. *Nucleic Acids Research*, 34(18):5112–5123, 2006. xii, xiii, 2, 3, 23, 24, 25, 26, 42, 45, 49, 51, 56, 57, 58, 61, 62, 71, 72

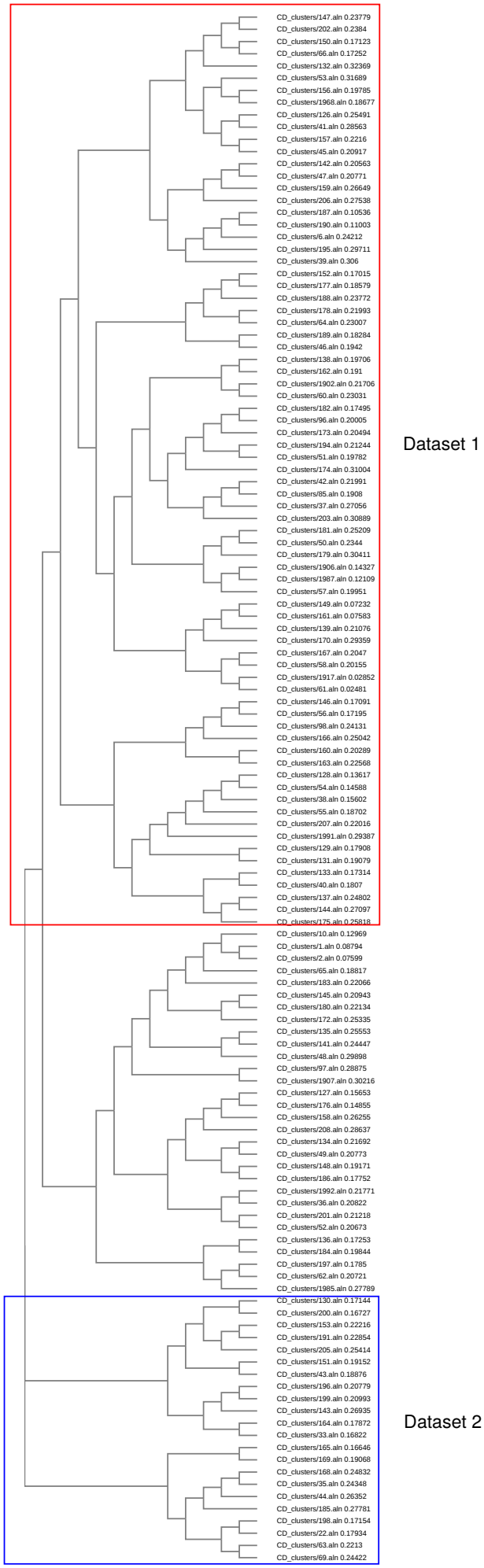
[96] L. Yongsheng et al. Genome-wide DNA methylome analysis reveals epigenetically dysregulated non-coding RNAs in human breast cancer. *Scientific Reports*, 5(8790):1–12, 2015. 41

[97] A. Zemann et al. Evolution of small nucleolar RNAs in nematodes. *Nucleic Acids Research*, 34(9):2676–2685, 2006. 45, 57, 58, 72

[98] T. Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *ICML 2004: Proceedings of the twenty-first International Conference on Machine Learning. OMNIPress*, pages 919–926, 2004. 40

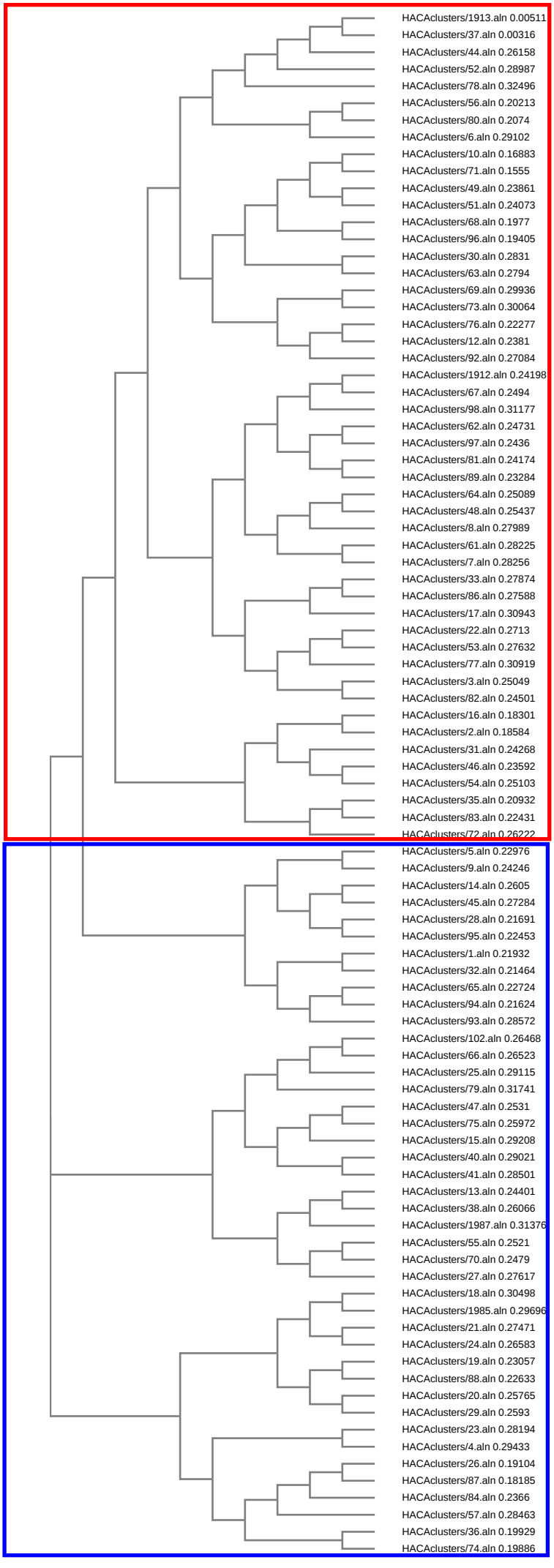
Anexo I

Distance tree for C/D box snoRNA clusters


CD_clusters/147.aln 0.23779
CD_clusters/202.aln 0.2384
CD_clusters/150.aln 0.17123
CD_clusters/66.aln 0.17252
CD_clusters/132.aln 0.32369
CD_clusters/53.aln 0.31689
CD_clusters/156.aln 0.19785
CD_clusters/1968.aln 0.18677
CD_clusters/126.aln 0.25491
CD_clusters/41.aln 0.28563
CD_clusters/157.aln 0.2216
CD_clusters/45.aln 0.20917
CD_clusters/142.aln 0.20563
CD_clusters/47.aln 0.20771
CD_clusters/159.aln 0.26649
CD_clusters/206.aln 0.27538
CD_clusters/187.aln 0.10536
CD_clusters/190.aln 0.11003
CD_clusters/6.aln 0.24212
CD_clusters/195.aln 0.29711
CD_clusters/39.aln 0.306
CD_clusters/152.aln 0.17015
CD_clusters/177.aln 0.18579
CD_clusters/188.aln 0.23772
CD_clusters/178.aln 0.21993
CD_clusters/64.aln 0.23007
CD_clusters/189.aln 0.18284
CD_clusters/46.aln 0.1942
CD_clusters/138.aln 0.19706
CD_clusters/162.aln 0.191
CD_clusters/1902.aln 0.21706
CD_clusters/60.aln 0.23031
CD_clusters/182.aln 0.17495
CD_clusters/96.aln 0.20005
CD_clusters/173.aln 0.20494
CD_clusters/194.aln 0.21244
CD_clusters/51.aln 0.19782
CD_clusters/174.aln 0.31004
CD_clusters/42.aln 0.21991
CD_clusters/85.aln 0.1908
CD_clusters/37.aln 0.27056
CD_clusters/203.aln 0.30889
CD_clusters/181.aln 0.25209
CD_clusters/50.aln 0.2344
CD_clusters/179.aln 0.30411
CD_clusters/1906.aln 0.14327
CD_clusters/1987.aln 0.12109
CD_clusters/57.aln 0.19951
CD_clusters/149.aln 0.07232
CD_clusters/161.aln 0.07583
CD_clusters/139.aln 0.21076
CD_clusters/170.aln 0.29359
CD_clusters/167.aln 0.2047
CD_clusters/58.aln 0.20155
CD_clusters/1917.aln 0.02852
CD_clusters/61.aln 0.02481
CD_clusters/146.aln 0.17091
CD_clusters/56.aln 0.17195
CD_clusters/98.aln 0.24131
CD_clusters/166.aln 0.25042
CD_clusters/160.aln 0.20289
CD_clusters/163.aln 0.22568
CD_clusters/128.aln 0.13617
CD_clusters/54.aln 0.14588
CD_clusters/38.aln 0.15602
CD_clusters/55.aln 0.18702
CD_clusters/207.aln 0.22016
CD_clusters/1991.aln 0.29387
CD_clusters/129.aln 0.17908
CD_clusters/131.aln 0.19079
CD_clusters/133.aln 0.17314
CD_clusters/40.aln 0.1807
CD_clusters/137.aln 0.24802
CD_clusters/144.aln 0.27097
CD_clusters/175.aln 0.25818
CD_clusters/10.aln 0.12969
CD_clusters/1.aln 0.08794
CD_clusters/2.aln 0.07599
CD_clusters/65.aln 0.18817
CD_clusters/183.aln 0.22066
CD_clusters/145.aln 0.20943
CD_clusters/180.aln 0.22134
CD_clusters/172.aln 0.25335
CD_clusters/135.aln 0.25553
CD_clusters/141.aln 0.24447
CD_clusters/48.aln 0.29898
CD_clusters/97.aln 0.28875
CD_clusters/1907.aln 0.30216
CD_clusters/127.aln 0.15653
CD_clusters/176.aln 0.14855
CD_clusters/158.aln 0.26255
CD_clusters/208.aln 0.28637
CD_clusters/134.aln 0.21692
CD_clusters/49.aln 0.20773
CD_clusters/148.aln 0.19171
CD_clusters/186.aln 0.17752
CD_clusters/1992.aln 0.21771
CD_clusters/36.aln 0.20822
CD_clusters/201.aln 0.21218
CD_clusters/52.aln 0.20673
CD_clusters/136.aln 0.17253
CD_clusters/184.aln 0.19844
CD_clusters/197.aln 0.1785
CD_clusters/62.aln 0.20721
CD_clusters/1985.aln 0.27789
CD_clusters/130.aln 0.17144
CD_clusters/200.aln 0.16727
CD_clusters/153.aln 0.22216
CD_clusters/191.aln 0.22854
CD_clusters/205.aln 0.25414
CD_clusters/151.aln 0.19152
CD_clusters/43.aln 0.18876
CD_clusters/196.aln 0.20779
CD_clusters/199.aln 0.20993
CD_clusters/143.aln 0.26935
CD_clusters/164.aln 0.17872
CD_clusters/33.aln 0.16822
CD_clusters/165.aln 0.16646
CD_clusters/169.aln 0.19068
CD_clusters/168.aln 0.24832
CD_clusters/35.aln 0.24348
CD_clusters/44.aln 0.26352
CD_clusters/185.aln 0.27781
CD_clusters/198.aln 0.17154
CD_clusters/22.aln 0.17934
CD_clusters/63.aln 0.2213
CD_clusters/69.aln 0.24422
Dataset 1
Dataset 2

Anexo II

Distance tree for H/ACA box snoRNA clusters



Dataset 1

Dataset 2

Anexo III

Poster apresentado no X-Meeting + BSB 2015: Identification of snoRNAs using EDeN

Identification of snoRNAs using EDeN

João Victor de Araujo Oliveira[1], Fabrizio Costa[2], Maria Emília M. T. Walter[1] and Rolf Backofen[2]

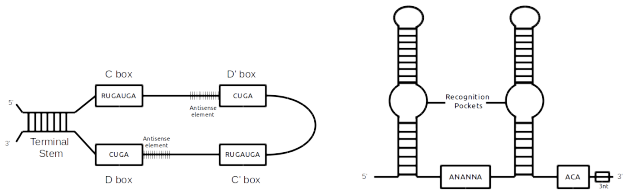
[1] Department of Computer Science, University of Brasilia,
[2] Bioinformatics Group, Department of Computer Science, Albert-Ludwigs-University Freiburg,

Introduction

Machine learning methods have been widely used on identification and classification of different families of non-coding RNAs, e.g., snoReport [1]. Many of these methods are based on supervised learning where some previous known attributes, called features, are extracted from a sequence, and then used in a classifier. Instead of using known features from a sequence (difficult to find in general) to identify ncRNAs, a recent approach in machine learning is described as follows. Given a region of interest of a sequence, the objective is to generate a sparse vector that can be used as micro-features in a specific machine learning algorithm, or it can be used to create powerful features on previous methods. One method that uses this approach is EDeN (Explicit Decomposition with Neighbourhoods). EDeN is a decompositional graph kernel based on Neighborhood Subgraph Pairwise Distance Kernel (NSPDK), that transforms one graph in a sparse vector decomposing it in all pairs of neighborhood subgraphs of small radius at increasing distances.

In this work, we present a new method based on EDeN to identify the two main classes of snoRNAs, C/D box and H/ACA box snoRNAs: transforming specific secondary structure regions of snoRNA in a graph representation used to build sparse vectors that can be used on different machine learning algorithms, e.g., stochastic gradient descent (SGD).

METHODS



C/D box snoRNA H/ACA box snoRNA

Figure 1: Canonical secondary structure of H/ACA and C/D box snoRNA

Creating two datasets for learning phase

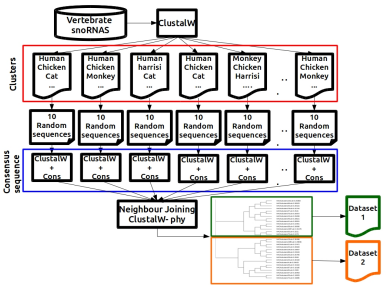


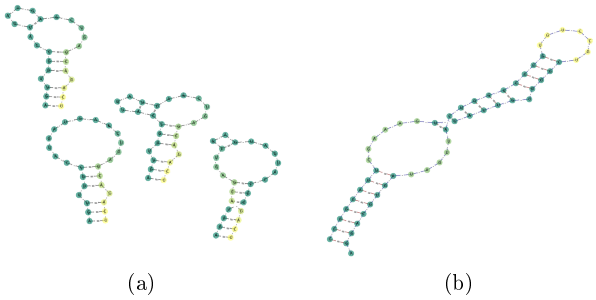
Figure 2: Creating two datasets for the learning phase. In order to avoid overfitting, First, we clustered the sequences using ClustalW [2], then 10 sequences from distinct vertebrates organisms were extracted from each cluster. Therefore, a consensus sequence from each cluster was obtained with ClustalW and Cons (for EMBOSS [3]), and these sequences were used to generate a distance tree, with the neighbour-joining method [4] from ClustalW2 - phylogeny [5].

EDeN CD finder

1. Find all combinations of C and D boxes with PWM-based scores above a certain threshold
2. Create a sample consisting on: a maximum of 10 nt before the C box + the C box + a special character '&' + the D box + and a maximum of 10 nt after D box (E.g NNNNNN**RUGAUGA&CUGA**NNNNNN)
3. Submit the sample to RNAsubopt [6] and transform its output into a graph using a EDeN routine
4. Transform the graphs into sparse vectors and send then to a Stochastic Gradient Descent (SGD) model that decides if these sequence are or not a C/D box snoRNA.

EDeN HACA finder

1. Find all combinations of H and ACA box predicted using H finder and ACA finder (We generated models that predict H and ACA boxes using a linear graph of the following sample: 3 nt before the box + x +the box + y + 3 nt after the box - e.g NNNx**ACA**yNNN)
2. Get 200 nt before a H box (hairpin 1), and take its secondary structure with RNAfold [6] with constraints
3. Transform the secondary structure in a annotated graph
4. Transform the annotated graph in a sparse vector and use it in a SGD model to obtain hairpin1 score
5. Do the same for the region between H box and ACA box (hairpin 2) to obtain hairpin2 score
6. Finally, use these 4 scores (H and ACA scores, hairpin 1 and hairpin2 scores) in a SVM model that returns if the candidate is or not a H/ACA box snoRNA



(a) (b)

Figure 3: Examples of graphs used in EDeN CD and HACA finder. (a) Example of graphs created using a C/D box snoRNA sequence in RNAsubopt. (b) Example of annotated graph of a secondary structure of the first hairpin loop in a H/ACA box snoRNA, where the internal loops have different weights compared to the stems, which could help us to identify good features in the vectorizer procedure.

Preliminary Results

Test Phase Results

Table 1: Test phase results for C/D and H/ACA box snoRNAs on EDeN CD/HACA finder compared with `snoReport` 2.0: Accuracy (Acc), F-score (F-SC), Average Precision (APR), Area under the ROC curve (AUC) and Residual Sum of Squares (RSS).

	ACC (%)	FSC (%)	APR (%)	AUC (%)	RSS
EDeN C/D finder	91.50	89.50	93.00	97.50	0.075
snoReport 2.0 (C/D)	95.28	94.30	98.61	98.96	0.037
EDeN HACA finder	88.95	62.06	96.98	99.01	0.085
snoReport 2.0 (H/ACA)	97.37	93.89	98.25	99.14	0.021

Discussion

At first analysis, snoReport 2.0 showed a better performance to identify C/D box snoRNAs. However, We observed that snoReport 2.0 discards 13% more positive samples on the pre-processing phase when compared with CD finder. This happens due to its restrictive approach, where a sample is submitted to a several secondary structure filters and it is checked if the candidate have some canonical properties, like a almost perfect kink turn structure. CD finder did not use any of these filters, because at this work we would like to generalize more our models of predictions. With that, we believe that CD finder have more power to identify non-canonical C/D box snoRNAs, as observed in some organism, like *Leishmania major* and other invertebrate organisms [7, 1].

EDeN HACA finder shows promising results. Further changes in the creation of the annotated graphs will be done in order to better characterize the secondary structure of a H/ACA box snoRNA, using a powerful routine in EDeN to optimize parameters used in the annotation (for example, setting a range of values of weights and other graph attributes in different regions of the secondary structure and making the EDeN itself choose the best configuration).

Conclusion and Perspectives

In this work, we presented a new method based on EDeN to identify the two main classes of snoRNAs, C/D box and H/ACA box snoRNA. Preliminary results on C/D box snoRNAs classifier showed F-score of 89.5%, Average Precision of 93%, and AUC of 97.5%. Furthermore, this new method discarded 13% less positive samples in the pre-processing phase, when compared to snoReport 2.0, allowing to discover a diversity of C/D box snoRNAs quite different to canonical ones. For H/ACA box snoRNA, further modifications will be done in the annotation of the graphs in order to improve its performance.
Next steps include: improve hairpin region annotation, using different techniques available on EDeN. training CD and HACA finder more times in order to guarantee that the classifiers in fact have a great performance; Validation on real data in several vertebrates and invertebrates organisms; and made EDeN CD and HACA finder available on Galaxy workflow [8].

ACKNOWLEDGEMENTS

J.V.A. Oliveira has been supported by CAPES scholarship. M.E.M.T. Walter has been continuously supported by productivity fellowship from CNPq (project 308509/2012-9).

References

[1] Hertel, J., Hofacker, I.L., Stadler, P.F.: SnoReport: computational identification of snoRNAs with unknown targets. Bioinformatics **24** (2008) 158–164

[2] Thopmson, J., Higgins, D., Gibson, T.: Clustalw: improving the sensitivity if progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice. Nucleic Acids Research **22** (1994) 4673–4680

[3] Rice, P., Longden, I., Bleasby, A.: EMBOSS: the European Molecular Biology Open Software Suite. Trends in genetics : TIG **16** (2000) 276–277

[4] Saitou, N., Nei, M.: The neighbor-joining method: a new method for reconstructing phylogenetic trees. Molecular Biology and Evolution **4** (1987) 406–425

[5] McWilliam, H., et al.: Analysis Tool Web Services from the EMBL-EBI. Nucleic Acids Research **41** (2013) W597–W600

[6] Hofacker, I.L.: Vienna RNA secondary structure server. Nucleic Acids Research **31** (2003) 3429–3431

[7] Liang, X.H., et al.: A genome-wide analysis of C/D and H/ACA-like small nucleolar RNAs in *Leishmania major* indicates conservation among trypanosomatids in repertoire and in their rRNA targets. Eukaryot. Cell **6** (2007) 361–377

[8] Goecks, J., et al.: Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. Genome Biology **11** (2010) R86+