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

Identificação de RNA não codificador utilizando redes neurais artificiais de treinamento não supervisionado

Tulio Conrado Campos da Silva

Brasília
2012

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

Identificação de RNA não codificador utilizando redes neurais artificiais de treinamento não supervisionado

Tulio Conrado Campos da Silva

Dissertação apresentada como requisito parcial
para conclusão do Mestrado em Computação

Orientador
Prof. Dr. Pedro de Azevedo Berger

Brasília
2012

Universidade de Brasília — UnB
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Mestrado em Computação

Coordenador: Prof. Dr. Maurício Ayala Rincón

Banca examinadora composta por:

Prof. Dr. Pedro de Azevedo Berger (Orientador) — CIC/UnB
Prof. Dr. André P. L. F. de Carvalho — ICMC/USP
Prof. Dr. Marcelo M. Brígido — IB/UnB

CIP — Catalogação Internacional na Publicação

da Silva, Tulio Conrado Campos.

Identificação de RNA não codificador utilizando redes neurais artificiais de treinamento não supervisionado / Tulio Conrado Campos da Silva. Brasília : UnB, 2012.
132 p. : il. ; 29,5 cm.

Tese (Mestrado) — Universidade de Brasília, Brasília, 2012.

1. ncRNA, 2. Inteligência Artificial, 3. Bioinformática, 4. SOM, 5. ART, 6. LVQ, 7. PCA

CDU 004.8

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro — Asa Norte
CEP 70910-900
Brasília–DF — Brasil

TÚLIO CONRADO CAMPOS DA SILVA

IDENTIFICAÇÃO DE RNA NÃO CODIFICADOR UTILIZANDO REDES NEURAIS ARTIFICIAIS DE TREINAMENTO NÃO SUPERVISIONADO

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre no Curso de Pós-graduação em Informática da Universidade de Brasília, pela Comissão formada pelos professores:

Orientador: Prof. Dr. Pedro de Azevedo Berger (CIC/UnB)

Prof. Dr. André Carlos Ponce de Leon Ferreira de Carvalho (USP)

Prof. Dr. Marcelo de Macedo Brígido (BIO/UnB)

Vista e permitida a impressão.
Brasília, 09 de março de 2012.

Prof. Dr. Mauricio Ayala Rincón - Coordenador
Programa de Pós-Graduação em Informática
Departamento de Ciência da Computação
Universidade de Brasília

Agradecimentos

Agradeço primeiramente à minha família, por todo o carinho, atenção e compreensão que me dão, sempre.
O empenho de meu orientador Pedro e de minha professora Maria Emília, depois de tantas reuniões, conseguimos obter tãos bons resultados! Agradeço com uma enorme felicidade, realmente. Também agradeço aos professores Marcelo Brígido, Tainá Raiol, Peter Stadler e Alexandre Zaghetto.
Também agradeço aos funcionários e pesquisadores da UnB e de outras instituições.
Especialmente aos meus colegas de mestrado e amigos do laboratório de Biologia Molecular da UnB. Pelas ótimas ideias, pelas confraternizações e pelo apoio.
Aos meus queridos amigos, que sempre me apoiaram e confiaram em mim.
A Deus.
Essa conquista é nossa!

Abstract

Several experiments conducted in the Molecular Biology field have shown that some types of RNA may control gene expression and phenotype by themselves, besides their traditional role of allowing protein synthesis. Roughly speaking, RNA can be divided into two classes: messenger RNA (mRNA), that are translated into proteins, and non-coding RNA (ncRNA), which play several important cellular roles besides protein coding. In recent years, many computational methods based on different theories and models have been proposed to distinguish mRNA from ncRNA. Among the newest methods, it is noteworthy the use of stochastic context free grammars, thermodynamical information, probabilistic theories and machine learning algorithms, which are very adaptive and low-complexity approaches. Particularly, machine learning methods that uses non-supervised learning artificial neural networks are a promising research field, for they are highly plastic and are able to classify ncRNA data using well established criteria. The present work extensively approaches the latter technique, particularly Self-Organizing Maps (SOM), Learning Vector Quantization (LVQ) and Adaptive Resonance Theory (ART) algorithms for distinguishing ncRNA from coding RNA in a given transcriptome. A test case was developed using biological data from 4 phylogenetically distant organisms. Using this test case, the trained networks achieved $\approx 98\%$ accuracy. The classification criteria used by the developed methods have been further optimized using Principal Components Analysis (PCA), reducing $\approx 32\%$ of the number of extracted numerical variables without reducing the assessed accuracy.

Keywords: ncRNA, Artificial Intelligence, Bioinformatics, SOM, ART, LVQ, PCA

Resumo

Experimentos diversos no campo da Biologia Molecular revelaram que alguns tipos de ácido ribonucleico (RNA) podem estar diretamente envolvidos na expressão gênica e do fenótipo, além de sua já conhecida função na síntese de proteínas. De modo geral, RNAs podem ser divididos em duas classes: RNA mensageiro (mRNA), que são traduzidos para proteínas, e RNA não codificador (ncRNA), que exerce papéis celulares importantes além de codificação de proteínas. Nos últimos anos, vários métodos computacionais baseados em diferentes teorias e modelos foram propostas para distinguir mRNA de ncRNA. Dentre os métodos mais atuais, destacam-se o uso de gramáticas estocásticas livres de contexto, informações termodinâmicas, teorias probabilísticas e algoritmos de aprendizado de máquina, sendo esses últimos abordagens muitos maleáveis e de menor complexidade. Particularmente, os métodos por aprendizado de máquina que utilizam redes neurais artificiais de treinamento não supervisionado constituem uma promissora linha de pesquisa, por sua grande plasticidade e capacidade de classificação do conjunto de dados de ncRNAs por critérios bem estabelecidos. Essa última técnica é extensivamente abordada no presente trabalho, mais precisamente utilizando Mapa Auto Organizável (SOM), *Learning Vector Quantization* (LVQ) e as redes Teoria da Ressonância Adaptativa (ART), para o problema de distinguir ncRNAs de mRNAs em um dado transcriptoma. As acurácias obtidas para as duas abordagens, em teste, ou estudo de caso, realizado com pequenos ncRNAs de 4 organismos filogeneticamente distantes atingiram $\approx 98\%$. Os critérios para classificação de ncRNA foram otimizados através da Análise de Componentes Principais (PCA), reduzindo o número de suas variáveis em $\approx 32\%$ sem reduzir a acurácia obtida no estudo de caso.

Palavras-chave: ncRNA, Inteligência Artificial, Bioinformática, SOM, ART, LVQ, PCA

Dedicatória

Aos meu queridos pais.

Sumário

Lista de Figuras

Lista de Tabelas

Glossário

AM Aprendizado de máquina. Classe de algoritmos de Inteligência Artificial. 2, 3, 17–19, 26, 42, 66, 71, 96

cDNA DNA complementar, formado a partir do processo inverso de transcrição aplicado a uma sequência de RNA mensageiro. Corresponde a um trecho de DNA sem material intrônico. 42, 49

HTS *High Throughput Sequencing*, Sequenciamento de Alto Desempenho, técnica de sequenciamento de DNA que produz grande volume de dados por execução, composto de pequenos fragmentos de DNA sequenciado ($\approx 200nt$). 3

métodos não supervisionados Classe de algoritmos de Aprendizado de Máquina cujo treinamento é realizado sem a necessidade de exemplares com nome de classe declarado. 2

métodos supervisionados Classe de algoritmos de Aprendizado de Máquina cujo treinamento é realizado com exemplares com nome da classe fornecido. 2

redes neurais artificiais Classe de algoritmos de Inteligência Artificial compostos por unidades de processamento (neurônios) interligadas. 2, 36

teste A fase de teste de um algoritmo de Aprendizado de Máquina estima a capacidade de generalização do conhecimento adquirido pelo algoritmo, normalmente utilizando dados diferentes dos dados de treinamento e validação. No escopo desse trabalho, a terminologia "estudo de caso" refere-se ao teste do algoritmo de Aprendizado de Máquina. vi, 4, 23, 63

treinamento Fase de aprendizado do algoritmo, ou aquisição de conhecimento, normalmente utilizando um conjunto de dados de exemplo. 3–5, 19–21, 23, 25, 29–31, 33, 35, 63–67, 71–73, 76, 77, 79–84, 87, 88, 91, 92, 95, 96

validação A etapa de validação de um algoritmo de Aprendizado de Máquina ocorre com a estimativa dos valores de desempenho da rede escolhidos, por exemplo acurácia e erro quadrático. 3, 5, 23, 26, 29, 63, 65, 67, 71, 74, 75, 84, 85, 92, 95–97

épocas Etapas de aprendizado de um algoritmo de Aprendizado de Máquina. Nesse trabalho, a conclusão de uma época pode ser definida cada vez que o conjunto de treinamento é totalmente apresentado ao algoritmo, no caso do algoritmo de Teoria

da Ressonância Adaptativa *ART-2* (Carpenter and Grossberg, 1987), ou cada vez que um exemplar de cada conjunto de treinamento é apresentado ao algoritmo, no caso das redes de Kohonen e do algoritmo *Learning Vector Quantization* (Kohonen et al., 1996b,a). 21, 30, 32–34, 54, 56, 57, 59, 60, 62–64, 72, 74, 92, 95, 96

Lista de Abreviaturas e Siglas

ART Teoria da Ressonância Adaptativa. vi, 2, 4, 5, 34–36, 40, 45, 48, 57–61, 63, 70, 71, 76, 82–85, 87–97

BIOMOL Laboratório de Biologia Molecular da Universidade de Brasília. 4

DNA ácido desoxirribonucleico. 1, 7–9, 11–13

HMM Modelo Oculto de Markov (*Hidden Markov Model*). 2, 42

IA Inteligência Artificial. 2, 5, 19, 26

LVQ *Learning Vector Quantization*. vi, 3, 5, 31–33, 40, 45, 48, 62, 63, 71, 79–82, 84, 85, 91, 93, 95, 96

mRNA RNA mensageiro. vi, 9, 11, 15, 16, 23, 37, 42, 45, 48, 56, 58, 80

ncRNA RNA não codificador. vi, x, 1, 4, 7, 12, 14, 16–18, 35, 37, 39, 42, 48, 53, 56, 74, 80, 96

ORF Janela ou fase aberta de leitura (*Open reading frame*). 12, 17, 37, 42–44, 51, 87

PCA Análise de Componentes Principais. vi, 3, 5, 26, 27, 40, 45, 48, 57, 60–62, 85–87, 96

RNA ácido ribonucleico. vi, 2, 3, 5, 7, 9, 11–14, 17, 18, 22, 39, 40, 42, 48, 50, 66, 69, 70, 80, 81, 96, 97

rRNA RNA ribossômico. 1, 12, 15

SCFG gramática estocástica livre de contexto. 2

SOM Mapa Auto Organizável. vi, 2–5, 23, 25, 30, 33–37, 39–41, 45, 48, 53, 56–58, 60–64, 67, 69–76, 80, 82–85, 87, 91–93, 95–97

SVM Máquina de Vetor de Suporte. 2, 4, 29, 31, 37, 66, 67, 70, 79

tRNA RNA transportador. 1, 15

Capítulo 1

Introdução

Nessa Seção, o trabalho é apresentado, em linhas gerais. Define-se o problema a ser abordado e sua motivação, bem como os objetivos e metas decorrentes, e o estado da arte sobre soluções para o problema em questão é comentado. Finalmente, a estrutura do trabalho é explicada.

1.1 Contextualização

A definição do Dogma Central da Biologia Molecular (descrito em detalhes no Capítulo 2) foi publicada na década de 50, por Watson e Crick (Watson and Crick, 1953). No trabalho, uma descrição detalhada do caminho da informação genética no organismo foi proposta, relacionando RNAs com papéis secundários de auxílio ao processo de síntese de proteínas. Essa definição inicial sofre um contínuo trabalho de aperfeiçoamento por meio de novas descobertas, primeiramente feitas, em sua maioria, ao acaso (Szymanski et al., 2007). O conhecimento cada vez mais amplo das funções e mecanismos genéticos, fisiológicos e metabólicos, entre outros, dos mais diversos organismos, além da criação de novas metodologias para descoberta dos processos microbiológicos, proporcionou uma melhor compreensão da atividade de codificação indireta do ácido desoxirribonucleico (DNA) em proteínas (Eddy, 2001), e inaugurou novas perpectivas de pesquisa direcionadas especificamente para os RNAs.

O ncRNA foi postulado pelo Dogma Central, desde sua criação, como sequências capazes de agir funcionalmente, regulatoriamente ou estruturalmente no organismo (Eddy, 2001), superando a visão simplificada de agente indireto da transcrição para entidade mais intrisecamente relacionada com o funcionamento celular. O clássico exemplo de RNAs que superavam uma função apositiva, de transporte da mensagem genômica a ser traduzida em sequência de peptídeos, foi dado pela descoberta das organelas ribossomais e do RNA ribossômico (rRNA) e do RNA transportador (tRNA) (Hoagland et al., 1958; Soll and RajBhandary, 1995). Estudos apontaram várias funcionalidades de ncRNAs, principalmente relacionadas à regulação da expressão gênica, e com funções estruturais, algumas relacionadas à transcrição e tradução de mRNAs. A contaminação de bancos de dados de sequências codificadoras, como o Swiss-Prot (Boeckmann et al., 2002), com material não codificante e a quantidade e relevância de tal material, antes considerado literalmente lixo intergênico (*junk DNA*) (Setubal and J. Meidanis, 2000), e constatado como composição intergênica majoritária em organismos eucariotos (Szymanski et al., 2007), também foram

importantes questões levantadas pelo estudo direcionado a ncRNAs. Para exemplificação, a quantidade de material genético codificante estimado em seres humanos é de apenas 2% (Szymanski et al., 2007), uma quantidade que, experimentalmente, acredita-se decrescer de acordo com a maior complexidade do organismo. Sabe-se também de inúmeras interações e relações entre controle da expressão de ncRNAs e surgimento ou supressão de diversas doenças, inclusive carcinomas em humanos (Gibb et al., 2011), bem como no controle da expressão em células musculares, como indicado pelo trabalho de Correia and Correia, 2007.

Motivados por tais descobertas iniciais, novos métodos computacionais dedicados especificamente para ncRNAs foram sucessivamente propostos. As primeiras abordagens seguiram a análise de alinhamentos utilizando ferramentas como o BLAST (Altschul et al., 1997) em busca de novos ncRNAs semelhantes a um banco de dados confiável de sequências não codificantes construído de forma empírica ou por intermédio de outros métodos computacionais. Tais abordagens tiveram sucesso limitado. Refinos sucessivos, adicionando informações específicas de determinados grupos ou classes ou famílias de ncRNAs melhoraram consideravelmente sua performance (Mount et al., 2007). Outros algoritmos, como o QRNA (Rivas and Eddy, 2001), analisam informação do alinhamento entre sequências de RNA utilizando gramática estocástica livre de contexto (SCFG) para identificar estruturas secundárias do RNA importantes para sua identificação funcional.

Devido ao forte caráter estrutural e composicional dos ncRNAs, relacionado diretamente à sua atuação na célula, os mais diversos algoritmos termodinâmicos foram propostos (Zuker et al., 1999; Hofacker et al., 2002). Inicialmente algoritmos que exigiam bastante recursos computacionais, foram sucessivamente melhorados e agora comportam avaliações de alinhamentos de sequências de RNA ou estimação de energias livres associadas a sequências individuais (Washietl et al., 2005). Outros métodos utilizam modelos de covariância e Modelo Oculto de Markov (*Hidden Markov Model*) (HMM) para identificar estruturas características de determinados tipos de ncRNA, atingindo desempenho considerável na classificação, principalmente para pequenos ncRNAs com estruturas bem conhecidas e definidas (Nawrocki et al., 2009).

A utilização de algoritmos de Inteligência Artificial (IA), particularmente métodos que utilizam AM, possibilitou, por sua vez, combinar vários algoritmos e dados diversos numa sinergia eficaz para identificação ou classificação de ncRNAs. Dentro de AM, enfatizou-se principalmente o uso de métodos supervisionados para o problema de identificação de ncRNAs, utilizando o algoritmo de Máquina de Vetor de Suporte (SVM) (Liu et al., 2005; Wang et al., 2006; Kong et al., 2007; Arrial et al., 2009). Essas abordagens obtiveram bons resultados, com acurácias em torno de 95%. Já os métodos não supervisionados encontram aplicação prática no campo da expressão gênica (Eisen et al., 1998) principalmente para análise de agrupamentos (*clusters*) de dados de expressão gênica e, mais recentemente, na análise de dados transcriptômicos para identificação de ncRNAs (Silva et al., 2009). Dada a sua capacidade geral de reconhecer padrões matemáticos complexos e, portanto, relações entre diferentes conjuntos de dados biológicos, métodos não supervisionados têm destaque em soluções que buscam identificar esses padrões complexos nos mais diferenciados problemas biológicos. Dentro da disciplina de métodos não supervisionados, tradicionalmente, redes neurais artificiais que utilizam o conceito de auto-organização (explicadas em detalhes no Capítulo 3) como SOM e ART são aplicadas para solução de problemas de reconhecimento de padrões e predição de padrões, como análise de imagens,

reconhecimento visual e sonoro, entre outros (Haykin, 1999; Kasabov, 1998). Tais padrões são traduzidos como agrupamentos de diferentes estímulos de treinamento em classes ou *clusters* com algum grau mensurável de semelhança entre si.

Como exemplo da capacidade de aplicação de redes não supervisionadas auto organizáveis para o problema de identificação e classificação, cita-se o recente método SOM-Portrait. Proposto por Silva et al., 2009, a ferramenta possibilita a identificação de sequências em três classes distintas: a classe de RNAs potencialmente codificadores (*Coding*), a classe de RNAs potencialmente não codificadores (*Noncoding*) e uma terceira classe hipotética, treinada com o propósito de confirmar a capacidade de categorização em classes de ncRNAs do método baseado em SOM. Esta classe é nomeada *Undefined*, e indícios experimentais foram obtidos para confirmar sua boa delimitação pela rede neural treinada, por meio de comparação com identificadores baseados em outras metodologias. A acurácia obtida na validação do treinamento do algoritmo, de $\approx 88\%$, motivou a atividade, nesse trabalho, de reformulação do referido método, melhorando consideravelmente sua acurácia e a percepção dos diferentes *clusters* de ncRNAs.

O algoritmo de AM necessita de conjuntos de dados biológicos que tenham bom grau de anotação de sua classe e função, de preferência manual, que servirão como matéria de aprendizado, ou experiência, do método, para que assim ele seja capaz de realizar uma determinada tarefa (Mitchell, 1997). Tais conjuntos de dados, chamados conjuntos de treinamento, para o problema de identificação de ncRNA, têm complexa elaboração utilizando bons representantes de ncRNAs já catalogados, por métodos pontuais, de bancada, ou por utilização de outros algoritmos de predição de ncRNAs. O problema de encontrar bons representantes para criar o conjunto de treinamento é um ponto revisitado nesse trabalho.

De posse do conjunto de treinamento, a seguinte etapa fundamental no correto aprendizado de um algoritmo AM é a escolha do tipo de informação a ser extraída das sequências de RNA, fonte da experiência adquirida em AM, que constitui os atributos extraídos. Normalmente, essa informação é fornecida ao algoritmo em forma de atributos numéricos (Kasabov, 1998). Vários trabalhos dissertam sobre bons atributos numéricos para a identificação de ncRNAs (Liu et al., 2005; Dinger et al., 2008; Arrial, 2008; Silva, 2009). Esse trabalho revisita essa escolha, utilizando PCA (Pearson, 1901) para criticar cada atributo previamente escolhido da literatura especializada a respeito.

Dentro da disciplina de AM, outros algoritmos utilizam análises hierárquicas, usando dados composicionais, termodinâmicos e de alinhamento oriundos da nova geração de HTS (Fasold et al., 2011). Apesar de abordagem interessante, nesse trabalho justifica-se não comportar utilização de dados de alto desempenho devido à relativa escassez de dados para treinamento recuperáveis nesse formato.

Abordagens híbridas de AM, com etapas não supervisionadas para agrupamentos dos dados seguidas de uma etapa supervisionada para identificação das categorias é uma prática apoiada (Kohonen, 2001) para algoritmos auto organizáveis. Tendo em vista essa característica, o presente trabalho propõe, além da reformulação do treinamento da rede SOM, a utilização de uma etapa posterior supervisionada utilizando o algoritmo LVQ (Kohonen et al., 1996a). Ainda partindo dos problemas encontrados durante a confecção do método SOM-Portrait, surge a necessidade de utilizar uma rede neural auto organizável que não dependa de um número inicial de neurônios na camada de processamento, como é o caso no algoritmo SOM. Tal dependência de um número arbitrariamente

selecionado pode ser diminuída utilizando análises estatísticas ou outros métodos, como K-médias. Porém, argumenta-se sobre a utilização, nesse trabalho, de uma rede baseada em ART (Carpenter and Grossberg, 1987), devido à sua capacidade de categorização condicionada a uma métrica controlável. Nesse trabalho, explora-se a capacidade dessa rede aplicada ao problema de identificação e classificação de ncRNAs, seus resultados comprovando também a validade da escolha anterior de classes para o algoritmo SOM.

O algoritmo SOM foi submetido a uma etapa de teste por meio de sequências conhecidas de ncRNAs do transcriptoma de vários diferentes organismos. A escolha desses organismos deve-se principalmente à confiabilidade dos dados de ncRNAs e à diversidade filogenética. De fato, escolheu-se dados de humano (*H. sapiens*), fungos (*S. cerevisiae*, *A. oryzae*, *C. immitis*, *P. brasiliensis*), planta (*A. thaliana*) e bactéria (*E. coli*). Comparação com outros métodos de identificação de ncRNA, como o PORTRAIT (Arrial et al., 2009) e Infernal (Nawrocki et al., 2009), também foram conduzidas para avaliar o treinamento obtido.

1.2 Definição do Problema

O presente trabalho aborda dois problemas relacionados:

1 Identificar sequências de RNAs não codificadores e também sua contrapartida codificadora, para qualquer tipo de organismo;

2 Validar o método para identificação e classificação de ncRNA chamado **SOM-Portrait**, que utiliza SOM, rede neural artificial de treinamento não supervisionado;

textbf3 Melhorar o método **SOM-Portrait**, e propor o método **ART-Portrait** como alternativa de aplicação de método não supervisionado ao problema de classificação de ncRNAs.

Os problemas têm por motivação o contínuo trabalho desempenhado entre 2008 e 2012 no Laboratório de Biologia Molecular da Universidade de Brasília (BIOMOL) (Brígido, 2012), onde primeiro surgiu a demanda por uma ferramenta de análise de ncRNAs especificamente criada para os projetos de sequenciamento e anotação vinculados ao BIOMOL. Através da experiência adquirida com a utilização da ferramenta PORTRAIT (Arrial et al., 2009) de identificação de ncRNAs por meio de SVM em duas classes distintas, codificante e não codificante, o problema de criar um método capaz de classificar em mais de duas classes de ncRNAs foi levantado. Por sua vez, tal questionamento tem por objetivo idealizar uma metodologia capaz de classificar os ncRNAs em classes ou conjuntos de classes com características comuns, para facilitar a anotação manual dos biólogos. Tal ferramenta se faz necessária devido ao caráter inédito de anotação de ncRNAs nos projetos em atividade no referido laboratório, tornando-se mister o domínio e aplicação das principais técnicas de identificação e análise de ncRNAs para auxiliar os biólogos na compreensão efetiva dos organismos estudados.

1.3 Objetivos

O principal objetivo do trabalho é o de **propor um método baseado em redes neurais de treinamento não supervisionado capaz de identificar e classificar ncRNAs dos mais diversos organismos, utilizando apenas informação da própria sequência de RNA fornecida**. A partir desse objetivo geral, 4 tarefas para o presente trabalho são definidas:

1 Propor o método SOM-Portrait, um classificador de ncRNAs baseado em SOM;

2 Propor uma etapa supervisionada para o classificador utilizando LVQ;

3 Propor o método ART-Portrait;

4 Realizar estudos de caso utilizando os organismos descritos anteriormente.

Em maior detalhe, o item **1** versa sobre a reimplementação do método SOM-Portrait, sem contemplar sua versão distribuída (Silva et al., 2009). A reimplementação envolve inclusive a reformulação do conjunto de treinamento, especialmente a parcela de sequências não codificantes, revisando o modo de seleção de candidatos para o treinamento. Uma avaliação detalhada usando PCA também é conduzida para auferir a relevância dos atributos numéricos extraídos das sequências de RNAs para o problema de identificação e classificação.

O item **2** é implementado como etapa subsequente ao treinamento da SOM, e avaliado usando-se os mesmos organismos descritos em detalhes no Capítulo 5. Já o item **3** não utiliza resultados da rede SOM. A rede ART treinada é uma proposta de um método paralelo ao SOM-Portrait, cujos resultados, entre outros, complementam e avaliam o treinamento da SOM.

O item **4** é aplicado a todos os métodos desenvolvidos, como forma de avaliar a capacidade de generalização do aprendizado das redes para identificar e classificar ncRNAs dos mais diversos organismos biológicos.

1.4 Organização do Trabalho

O presente trabalho se divide, para estas finalidades, em:

- Capítulo 2: detalhamento de conceitos fundamentais relativos a Biologia Molecular, Bioinformática detalhamento de conceitos fundamentais relativos a RNAs não codificadores;
- Capítulo 3: detalhamento de teoria e conceitos relativos a IA, redes neurais artificias, SOM, LVQ, ART e PCA;
- Capítulo 4: versa sobre métodos, ferramentas, bibliotecas e materiais utilizados para a confecção dos algoritmos e experimentos realizados, inclusive o conjunto de dados reais biológicos utilizados para validação dos métodos propostos. É um resumo técnico de todo o trabalho realizado para validar o método não supervisionado SOM-Portrait no contexto de classificação de RNA não codificador;

- Capítulo 5: descreve e discute os resultados obtidos dos experimentos descritos no Capítulo 4 à luz dos objetivos propostos;
- Capítulo 6: sintetiza a linha do tempo com as atividades realizadas, os principais resultados obtidos, e conclui o trabalho com as linhas de pesquisa e atividades futuras;
- Anexo I: descreve os parâmetros de ajuste e configuração das bibliotecas SOM_PAK, LVQ_PAK, ART-distance, ANGLE e CAST (Kohonen et al., 1996b,a; Promponas et al., 2000; Shimizu et al., 2006; Hudik and Zizka, 2011);
- Anexo II: descreve os procedimentos de configuração do ambiente de trabalho utilizado para execução das ferramentas desenvolvidas.

O presente trabalho contém arquivos e dados complementares em mídia digital, que acompanha o texto. Quando citados, são referenciados como "Material Complementar".

Capítulo 2

RNAs não codificadores

Nessa Seção os fundamentos de Biologia Molecular e Bioinformática utilizados no trabalho são explanados. O conceito de RNA não codificador (ncRNA) é definido em seguida, junto com bases de dados sobre ncRNAs disponíveis e ferramentas e métodos automatizados comumente utilizados para identificá-los e classificá-los.

2.1 Conceitos de Biologia Molecular e Bioinformática

Biologia Molecular é o ramo da Biologia responsável, basicamente, pelo estudo da estrutura de proteínas e ácidos nucleicos, processos relacionados e outros atores envolvidos, como organelas celulares e enzimas (Setubal and J. Meidanis, 2000). Tais estruturas são abordadas a seguir. Bioinformática é o ramo sinérgico entre Computação, Matemática e Biologia Molecular que contribui com modelos, análises estatísticas, algoritmos e sistemas de computação, entre outras contribuições teóricas e práticas à área de Biologia Molecular (Clote and Backofen, 2000). Assim, a Bioinformática é especialmente dedicada aos vários e complexos problemas que a Biologia Molecular oferece.

2.1.1 Proteínas e ácidos nucléicos

Ácidos desoxirribonucleico (DNA) e ribonucleico (RNA) são ácidos nucleicos, cuja função principal é a de armazenar informação necessária para criação de proteínas e possibilitar a transferência desta informação para outros organismos, por meio de processos de reprodução celular (Setubal and J. Meidanis, 2000).

Tanto o DNA como o RNA são compostos por cadeias de elementos menores. No caso de DNAs e RNAs, a composição de um grupo fosfato, um açúcar central e uma base nitrogenada, formam o **nucleotídeo** (Clote and Backofen, 2000; Setubal and J. Meidanis, 2000). A composição de cadeias de nucleotídeos forma uma sequência RNA ou DNA, dependendo do açúcar central deste nucleotídeo e de suas bases nitrogenadas. No caso do DNA, o açúcar é a desoxirribose, já no RNA o açúcar é a ribose. Tais açúcares compõem a estrutura central do nucleotídeo, portanto, neles se ligam as bases nitrogenadas e o grupo fosfato. Assim sendo, convencionou-se uma notação para detalhar a posição dessa ligação na molécula de açúcar. Tanto a desoxirribose quanto a ribose são pentoses, ou seja, açúcares compostos por 5 carbonos. Os grupos fosfatos se ligam nas posições 5' (5^o

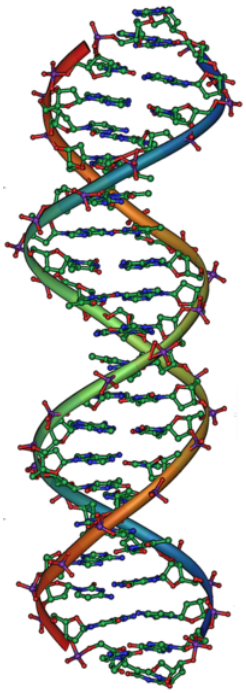


Figura 2.1: Representação da estrutura em dupla-hélice do DNA (RNA, 2011).

carbono) e 3' (3^o carbono) dessa pentose, uma notação muito utilizada para indicar o sentido da cadeia de nucleotídeos.

As bases nitrogenadas utilizadas no DNA são 4: a Adenina (A), Citosina (C), Guanina (G) e Timina (T). O modelo de Watson-Crick do DNA define que há uma afinidade entre as bases nitrogenadas, por causa de sua disposição espacial e afinidade eletrônica da molécula. Bases **purinas** (Adenina e Guanina) somente se ligam a bases **pirimidinas** (Timina e Citosina) (Watson and Crick, 1953), sendo a ligação de bases purina-pirimidina dita ligação de bases complementares. Pela característica das bases complementares, é possível extrair o complemento de uma faixa de DNA aplicando a seguinte regra:

$$Guanina \Longleftrightarrow Citosina \tag{2.1}$$

$$Adenina \Longleftrightarrow Timina \tag{2.2}$$

Da disposição espacial de uma fita DNA, indo da posição de ligação no açúcar principal 5' à posição 3', também se conclui que seu complemento é o exato oposto, indo do 3' ao 5'. Portanto, uma faixa é o exato **complemento reverso** da outra, possibilitando a duplicação de trechos do código DNA.

A Figura 2.1 ilustra a famosa estrutura de dupla hélice do DNA. O DNA pode ser encontrado na forma de cromossomos (aglomerado com proteínas para reduzir espaço), em forma circular (principalmente em organismos menos complexos, como bactérias) e em sua forma linear, assim como apresentado na Figura 2.1.

Grande parte do material genético encontrado em DNA não codifica para proteínas em organismos eucariotos, segundo observações extensas (Szymanski et al., 2003). Dá-se o nome de **genes** para as regiões delimitadas do DNA que codificam para proteínas ou

Figura 2.2: Representação da fita simples de RNA (RNA, 2011).

RNAs (Setubal and J. Meidanis, 2000), isso é, no ato de transcrição, o DNA é transcrito para um RNA funcional válido ou para um mRNA válido.

No RNA, a base nitrogenada Timina (T) é substituída pela base nitrogenada Uracila (U), também uma base pirimídica. Identicamente ao DNA, a orientação do RNA se dá do carbono 5' ao carbono 3' da ribose. Pode-se reescrever as regras de complementaridade das bases nitrogenadas descritas na Regra 2.2 simplesmente trocando-se a base Timina pela base Uracila. A estrutura de um filamento do RNA o torna mais vulnerável a danos e erros, e portanto menos apto a transportar informação genética (Clote and Backofen, 2000). Por essa característica, além da estrutura química mais simplificada tanto da base Uracila, quando confrontada com a base Timina, como da estrutura do RNA, existem várias teorias de que o RNA teria sido o primeiro ácido nucleico a ser usado como transportador de material genético (Eddy, 2001). A Figura 2.2 mostra a representação da fita simples de RNA.

Proteínas participam direta ou indiretamente de quase todas as atividades celulares de um organismo vivo (Setubal and J. Meidanis, 2000). A Figura 2.3 exibe uma representação visual da proteína *hemoglobina*, presente no sangue humano. São formadas a partir da ligação peptídica de vários aminoácidos, resultando num longo polipeptídeo, por vezes usado como sinônimo de proteína. Dessa forma, proteínas não são exatamente compostas de aminoácidos, mas sim do resíduo dessa ligação, os peptídeos (Clote and Backofen, 2000). Na natureza, são catalogados 22 aminoácidos conhecidos (Lesk, 2002), sendo 20 não-polares, comumente achados em proteínas, e 2 polares, mais raramente encontrados.

A Tabela 2.1 mostra o nome, abreviação e o código de uma letra usado para identificar o aminoácido. Em asterisco (*), os aminoácidos menos comumente encontrados em proteínas.

A função de uma proteína é grandemente caracterizada pela sua estrutura espacial (Clote and Backofen, 2000). A sequência de resíduos que forma a proteína é dita **estrutura primária** da proteína. Tal estrutura linear não caracteriza sua função (Setubal and J. Meidanis, 2000). A **estrutura secundária** de uma proteína é formada pelo

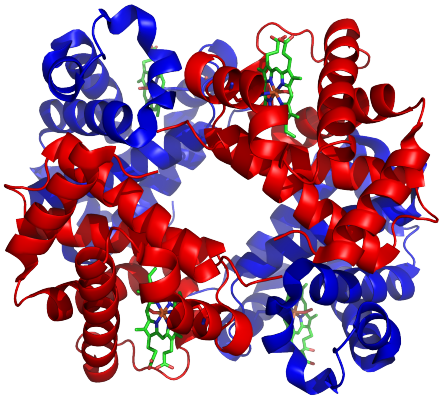


Figura 2.3: Estrutura quaternária da *hemoglobina*, proteína do sangue (Boaz and Shaanan, 1983).

Tabela 2.1: Lista dos 22 aminoácidos encontrados na natureza.

	Nome	**Abreviação**	**Código**
1	Alanina	Ala	A
2	Arginina	Arg	R
3	Asparagina	Asn	N
4	Ácido Aspártico	Asp	D
5	Asparagina ou Ácido Aspártico *	Asx	B
6	Cisteína	Cys	C
7	Glutamina	Gln	Q
8	Ácido Glutâmico	Glu	E
9	Glutamina ou Ácido Glutâmico *	Glx	Z
10	Glicina	Gly	G
11	Histidina	His	H
12	Isoleucina	Ile	I
13	Leucina	Leu	L
14	Lisina	Lys	K
15	Metionina	Met	M
16	Fenilalanina	Phe	F
17	Prolina	Pro	P
18	Serina	Ser	S
19	Treonina	Thr	T
20	Triptofano	Trp	W
21	Tirosina	Tyr	Y
22	Valina	Val	V

alinhamento e dobramento da sequência de resíduos. **Estruturas terciárias** evidenciam o formato tridimensional de proteínas no organismo, sua estrutura nativa. Note que a estrutura nativa nem sempre é funcional no organismo. Diversos outros processos pós-tradução podem refinar e reformular a proteína até seu estado funcional. Esta estrutura

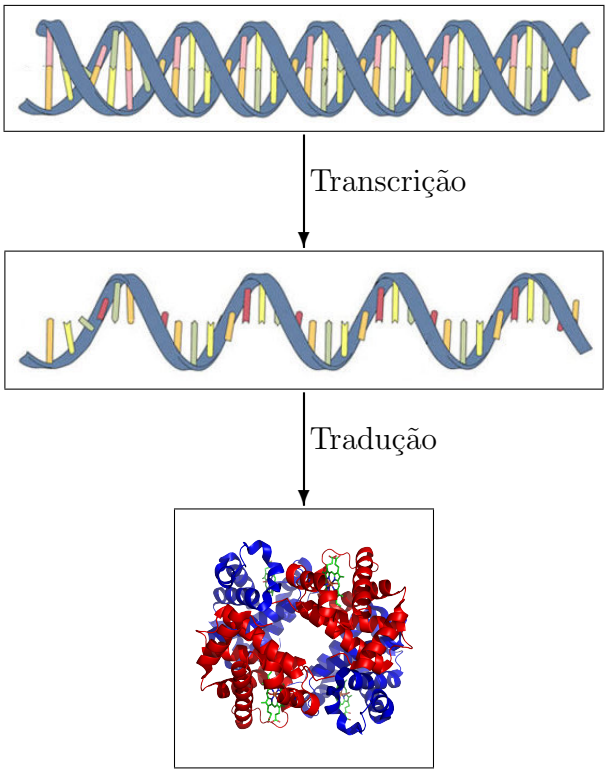


Figura 2.4: Representação gráfica do Dogma Central da Biologia Molecular.

nativa muitas vezes pode ser sintetizada em laboratório. Finalmente, a **estrutura quaternária** de uma proteína considera sua totalidade, em forma tridimensional nativa e terminada, isto é, sua forma ativa naturalmente encontrada no organismo. A Figura 2.3 representa a estrutura quaternária da *hemoglobina*.

2.1.2 O Dogma Central da Biologia Molecular

O Dogma Central da Biologia Molecular, determinado pelos estudos de Watson e Crick relacionados a ácidos nucleicos (Watson and Crick, 1953) relaciona os principais agentes da Biologia Molecular, ácidos nucleicos e proteínas, com atividades celulares muito importantes, o processo de **replicação** de trechos de DNA genômico, de **transcrição**, e de **tradução**. A Figura 2.4 resume o processo biológico de transmissão da informação contida no DNA até seu produto peptídico, frisando os principais atores envolvidos.

Transcrição envolve mecanismos e proteínas celulares com o objetivo de transformar genes do DNA em RNA. Já a **tradução** utiliza cadeias de RNA chamadas mensageiro - o **mRNA** - para traduzir sua sequência de bases nitrogenadas em aminoácidos, ligando-os com o auxílio de organelas celulares e outros tipos de RNA para formar proteínas (Setubal and J. Meidanis, 2000).

Durante a tradução, os nucleotídeos do mRNA são agrupados em códons (trincas de nucleotídeos), que são traduzidos no seu correspondente aminoácido, conforme a Tabela 2.2. Nota-se também a presença de sequências de parada ($STOP$) nessa tradução, bem como a sequência de início de uma proteína, o aminoácido Metionina (Met), representado pelo códon AUG (Clote and Backofen, 2000).

Tabela 2.2: Mapeamento de códons para aminoácidos e sequências de controle (Setubal and J. Meidanis, 2000).

	U	C	A	G	
U	Phe	Ser	Tyr	Cys	U
	Phe	Ser	Tyr	Cys	C
	Leu	Ser	STOP	STOP	A
	Leu	Ser	STOP	Trp	G
C	Leu	Pro	His	Arg	U
	Leu	Pro	His	Arg	C
	Leu	Pro	Gln	Arg	A
	Leu	Pro	Gln	Arg	G
G	Ile	Thr	Asn	Ser	U
	Ile	Thr	Asn	Ser	C
	Ile	Thr	Lys	Arg	A
	Met	Thr	Lys	Arg	G
A	Val	Ala	Asp	Gly	U
	Val	Ala	Asp	Gly	C
	Val	Ala	Glu	Gly	A
	Val	Ala	Glu	Gly	G

Um conceito importante para a tradução é o de **janela de leitura**. As sequências de bases nitrogenadas que formam cadeias de DNA e RNA podem ser agrupadas em códons (tripla de bases nitrogenadas) de diferentes maneiras, sempre obedecendo à ordem de leitura do carbono 5' ao 3'. Uma janela de leitura, em inglês *reading frame*, é um possível agrupamento em triplas da sequência, ao se adotar um determinado ponto da cadeia para começar o agrupamento (Clote and Backofen, 2000). Já uma Janela ou fase aberta de leitura (*Open reading frame*) (ORF) - é uma configuração em que a escolha de triplas resulta numa sequência contínua de triplas que representam exclusivamente aminoácidos, sem sequências de parada (sequências *STOP*), e que é múltipla de três, ou seja, não deixa bases residuais ao ser agrupado (Setubal and J. Meidanis, 2000).

2.2 Os RNAs não codificadores

RNAs não codificadores (*non coding RNAs* ou *ncRNAs*) são transcritos de genes que não expressam mRNAs codificadores de proteínas. Pelo contrário, agem diretamente na célula em funções *estruturais*, *catalíticos* ou *regulatórios* (Eddy, 2001). A Figura 2.5 dá exemplo da estrutura secundária de um micro RNA, que são pequenos ncRNAs, e ressalta a importância da conformação estrutural de tais entidades.

Exemplos de ncRNA com função estrutural são rRNA, pequenos RNAs nucleolares (snoRNAs), entre outros. As linhas de pesquisa atuais apontam relações mais extensas entre ncRNAs e os mais diversos processos de um organismo. Ainda muito pouco é conhecido, porém, principalmente pela grande dificuldade em verificar experimentalmente qual é exatamente a funcionalidade do determinado gene não codificador no or-

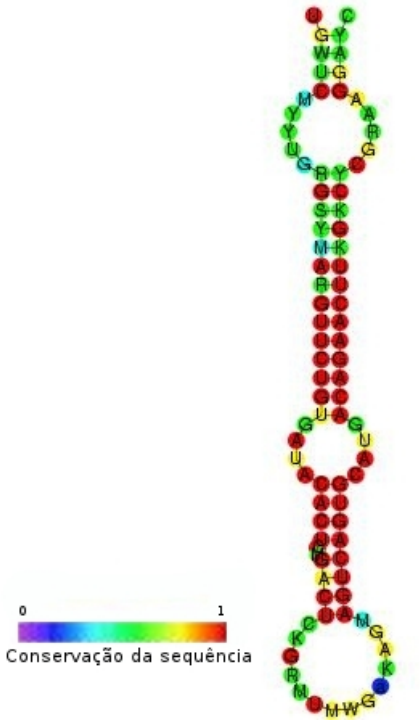


Figura 2.5: Exemplo de um RNA não codificador da família dos micro RNA (μRNA ou miRNA) mostrando sua característica estrutura em grampo de cabelo (*hairpin*) (Gardner et al., 2009).

ganismo (Machado-Lima et al., 2008). Existe uma relação intrínseca entre a quantidade de material não codificante e a crescente complexidade de um organismo. Sabe-se, por exemplo, que em bactérias, organismos eucariotos unicelulares, invertebrados e mamíferos, a porcentagem de sequências codificantes é de, respectivamente, $\approx 95\%$, 30%, 20% e $\approx 2\%$ (Szymanski et al., 2007).

Sequências de ncRNAs têm fundamental papel no controle da expressão de genes em proteínas (Jossinet et al., 2007), bem como na diversidade epigenética de células em um organismo, regulação dos processos de transcrição, tradução, excisão, entre outros. Estudos indicam a relação entre o nível de expressão de certos ncRNAs regulatórios em humanos e o seu comportamento neural, problemas de desenvolvimento ou câncer (Szymanski et al., 2007). A Figura 2.6 inclui possíveis funções de ncRNAs no Dogma Central da Biologia Molecular.

Historicamente, a identificação de trechos de DNA que, transcritos, resultavam em RNAs não codificadores ocorreu na identificação de regiões intergenicas sem função aparente, presentes em grandes quantidades em organismos eucariotos complexos (Setubal and J. Meidanis, 2000). Estudos posteriores sobre o processo de transcrição e tradução realizados por Watson e Crick já postulavam, controversamente, a hipótese de "um gene, um ribossomo e uma proteína", dando caráter exclusivo para o RNA de mero sintetizador de proteínas a nível citoplasmático, mas também a provável existência de estruturas mais complexas de RNA como intermediadores de atividades de tradução, como os mais tarde identificados RNAs de transporte (tRNAs) vieram confirmar (Watson and Crick, 1953; Eddy, 2001).

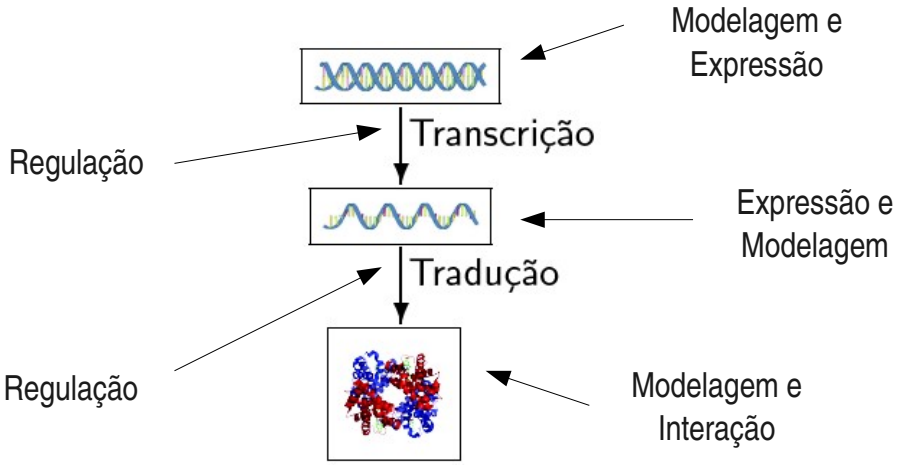


Figura 2.6: Exemplo de funções do ncRNA nas atividades de transcrição, tradução e excisão em eucariotos.

Antes chamados de DNAs lixo, ou *junk DNAs*, hoje as funcionalidades conhecidas para ncRNAs abrangem silenciamento de genes, replicação, regulação da expressão de genes, transcrição, estabilidade de cromossomos, estabilidade de proteínas, translocação, localização e modificação, processamento e estabilidade de RNA (Machado-Lima et al., 2008). Estudos nesta área têm por objetivo principal estabelecer os critérios para distinguir precisamente ncRNAs de mRNAs e possibilitar uma maior compreensão do mecanismo genético e seus produtos (Frith et al., 2006).

2.2.1 Famílias e classificações de ncRNAs

As classificações de ncRNAs variam conforme sua aparente funcionalidade visualizada no organismo. Ainda há muita discussão sobre a quantidade de ncRNAs e sobre como estes são divididos (Eddy, 2001). A Tabela 2.3 lista alguns tipos de ncRNAs e suas funções mais conhecidas em organismos.

Apesar de RNA funcional ser utilizado com a mesma acepção de ncRNA, fRNA é mais precisamente definido como o ncRNA com função catalítica ou regulatória (Szymanski et al., 2007). A classificação da Tabela 2.3 é determinada pela semelhança estrutural e funcional dos membros de cada família. A exceção são os snmRNAs e lncRNAs, agrupados somente por características estruturais, mas na verdade são uma reunião de várias famílias e subclasses de diversos ncRNAs. Particularmente, vários estudos recentes têm por alvo explorar as funcionalidades de diversos lncRNAs em eucariotos (Mercer et al., 2009; Gibb et al., 2011).

2.2.2 Bancos de dados de ncRNAs

Estudos sobre identificação de trechos não codificantes de genomas foram crescendo em importância. Inicialmente descobertos por meio de procedimentos empíricos e pontuais, e muitas vezes descobertos por acaso (Szymanski et al., 2007), foram gradualmente ga-

Tabela 2.3: Alguns tipos de RNAs não codificadores e suas funções conhecidas (Eddy, 2001; Lakshmi and Agrawal, 2007).

Sigla	Nome	Função
fRNA	RNA funcional	usualmente utilizado como sinônimo de RNA não codificador
miRNA	micro RNA	família putativa de genes reguladores da tradução. Pertence à classe dos ncRNAs estruturais
rRNA	RNA ribossômico	RNA constituinte do ribossomo
siRNA	RNA pequeno de interferência	moléculas ativas na interferência de RNA. Junto com o miRNA, constitui a classe dos ncRNAs estruturais
snRNA	Pequeno RNA nuclear	incluem RNAs relacionados ao processo de excisão
snmRNA	Pequeno não-mRNA	essencialmente pequenos RNAs não codificadores
snoRNA	Pequeno RNA nucleolar	envolvidos na modificação do rRNA
tRNA	RNA de transferência	envolvidos na tradução de mRNAs
rasiRNA	*Repeat-associated small interfering* RNA	Silenciamento da transcrição de genes via remodelagem da cromatina
lncRNAs	Longos ncRNAs ($> 200nt$)	Diversas funcionalidades, muitas das quais desconhecidas

nhando espaço e procedimentos específicos à medida em que novas funcionalidades foram relacionadas à expressão ou atividade de ncRNAs no organismo.

Devido a essa crescente atividade de identificação e descrição de funcionalidades de ncRNAs nos mais diversos organismos, a necessidade de se criar um banco de dados para organizá-los, análogo a bancos de dados de proteínas (PDB (Berman et al., 2000)), genomas (UCSC (Kent et al., 2002)) ou sequências diversas de mRNA e proteínas (Swiss-Prot e TrEMBL (Boeckmann et al., 2002)) já existentes.

O banco de dados **EMBL** (*EMBL Nucleotide Database*) (Cochrane et al., 2008) é constituído de diversos índices e integrações de teor colaborativo entre os mais diversos bancos de dados, com elevado volume de dados e número de anotações, tornando-o uma referência muito confiável para coleções de sequências de nucleotídeos.

O banco **Swiss-Prot** (Boeckmann et al., 2002) é especializado em sequências protéicas, concentrando-se na anotação de entradas do projeto de sequenciamento do genoma humano e de outros projetos de organismos modelo. Mantém, desta forma, anotações de boa qualidade.

O **RNAdb** (Pang et al., 2005) é composto por mais de 800 sequências de ncRNAs estudadas experimentalmente e especialmente selecionadas por sua associação com doenças e processos de crescimento e desenvolvimento em organismos. O banco não contém RNAs

de transferência ou RNAs ribossômicos, e também engloba várias sequências do genoma humano.

Incluindo RNAs estruturais não codificadores e regulatórios, o **Rfam** (Gardner et al., 2009) reúne diversas famílias de RNAs amplamente estudadas e anotadas, tais como RNAs de transferência e RNAs ribossômicos, como outras famílias de ncRNA com número mais limitado de anotações.

NONCODE (Liu et al., 2005) é um banco de dados integrado dedicado exclusivamente a catalogar e armazenar informações relativas a ncRNAs. Suas entradas são oriundas de dados obtidos de outros bancos de dados, notadamente o **GenBank**, e também de literaturas científicas relacionadas. Suas características composicionais incluem a ausência de RNAs de trasmissão e RNAs ribossômicos e a corroboração por meio de confronto com produções científicas relacionadas de mais de 80% de suas entradas. Em sua primeira versão, o banco conta com 5.339 sequências não redundantes dos mais variados organismos unicelulares.

O banco de dados **fRNAdb** (*functional RNA database*) (Kin et al., 2007) contém sequências de outros bancos de dados, inclusive o **NONCODE** e **RNAdb**. O banco contém também ferramentas para anotação automática de ncRNAs, utilizando abordagens termodinâmicas, estruturais e composicionais. Tais abordagens são explicadas na Subseção 2.2.3.

2.2.3 Abordagens para identificação de ncRNAs

Antes de prosseguir aos métodos e abordagens para identificação de ncRNAs, é importante frisar a diferença entre *classificar* e *identificar* ncRNAs. Métodos identificadores, atualmente, têm por objetivo separar sequências potencialmente não codificadoras de sequências codificadoras, de forma inequívoca e única. Já métodos classificatórios, mais avançados, são mais apurados em sua operação, discernindo sequências potenciais entre diversas classes de ncRNA. Métodos identificadores são comumente chamados de classificadores binários.

Não existe um consenso em métodos ou procedimentos para distinção precisa entre ncRNA e mRNA na Biologia Molecular contemporânea (Eddy, 2001; Frith et al., 2006). Esse fato gerou, por sua vez, a criação de diversos métodos computacionais para resolver o problema de identificação de ncRNAs. Estratégias consagradas para identificação e comparação de genes codificadores de proteínas falham ao serem aplicadas em transcritos não codificadores (Machado-Lima et al., 2008). Mesmo as sequências anotadas manualmente têm uma relevante porcentagem de erro: aproximadamente 10% das sequências manualmente traduzidas no banco de dados de proteínas Swiss-Prot são na verdade ncRNAs (Frith et al., 2006). A dificuldade em discriminar sequências genéticas como ncRNAs ou mRNAs é ainda maior quando aplicada a longos transcritos, com tamanho superior a 200 nucleotídeos (Dinger et al., 2008).

As estratégias computacionais para discriminar transcritos atualmente baseiam-se na identificação de atributos e características específicas de certas classes de ncRNAs ou na identificação menos restrita de atributos genéricos de famílias de ncRNAs em transcritos. Os atributos utilizados para discriminação podem ser extraídos diretamente da sequência, por processos *ab initio*, ou podem ser inferidos por comparação com um banco de dados estabelecido, realizando uma avaliação comparativa (Machado-Lima et al., 2008; Dinger

et al., 2008). Apesar do êxito experimental de vários métodos, a discriminação entre sequências de RNA capazes de atuar tanto funcionalmente na célula quanto gerar produtor proteicos, fenômeno pouco observado mas possivelmente muito comum, não é realizada por nenhum dos métodos atuais (Dinger et al., 2008). A seguir, as abordagens consideradas dão as linhas gerais de funcionamento de vários métodos baseados em uma ou mais dessas estratégias.

Avaliação termodinâmica

A composição e ordenação de nucleotídeos em uma molécula de RNA é responsável por sua conformação no espaço tridimensional. Uma investigação desta conformação, por sua vez, resulta num conhecimento aproximado sobre a organização da molécula e suas propriedades fisiológicas. A avaliação termodinâmica de moléculas de RNA pode ser utilizada em conjunto com várias regras estruturais e topológicas para inferir a estrutura secundária ativa da molécula de RNA (Zuker and Stiegler, 1981). RNAs com uma estrutura secundária bem definida têm energia livre associada menor do que sequências com mesma frequência de nucleotídeos, porém sem estrutura secundária definida (Machado-Lima et al., 2008). A partir da análise da mínima energia livre de uma molécula de RNA, é possível, portanto, inferir se a molécula tem uma conformação estável de sua estrutura secundária e se é possível sua atuação a nível funcional na célula.

Avaliação composicional

Há um forte indício em estudos e experimentos de que a ocorrência dos nucleotídeos G ou C é significativamente maior em transcritos de ncRNAs (Machado-Lima et al., 2008). Ocorrências do dinucleotídeo CG indicam, em estudos realizados com discriminadores utilizando máquinas de aprendizado supervisionado (Arrial, 2008), que sua conformação quimicamente mais estável do que a dupla AT ou TA possibilita a formação de uma estrutura secundária funcional.

Além de avaliações porcentuais de ocorrência, a avaliação por comprimento de fases de leitura ou do próprio transcrito são bastante utilizadas. Experimentalmente, várias classes e tipos de ncRNA contém de 15 a 300 nucleotídeos (nt). Aplicado a cadeias de proteínas putativas, a divisão orbita em torno de 100 peptídeos. Estes tipos de divisões, quando aplicadas a algoritmos discriminatórios, têm obtido bons resultados experimentais (Liu et al., 2006; Arrial, 2008).

Avaliação utilizando aprendizado de máquina

Métodos que utilizam AM mostraram um grau de acurácia elevado (Machado-Lima et al., 2008). CONC (Kong et al., 2007), CPC (Liu et al., 2006), PORTRAIT (Arrial, 2008) e SOM-PORTRAIT (Silva et al., 2009) utilizaram redes neurais artificiais treinadas com dois conjuntos: o positivo sendo constituído de características extraídas de trancritos de mRNAs e o negativo constituído de características extraídas de transcritos de ncRNAs. Os atributos extraídos foram, por exemplo, tamanho da ORF, composição de nucleotídeos, estrutura secundária, entre outros.

A aplicação de métodos por AM ao problema de detecção de ncRNAs, apesar de ter implementação mais simples, comparativamente, enfrenta problemas na etapa fundamen-

tal de construção do conjunto de treinamento de tais algoritmos. Como exemplo, cita-se métodos por AM aplicados ao problema de reconhecer, a partir de uma sequência de nucleotídeos de RNA, sua estrutura secundária conservada, propriedade fundamental em grande maioria dos ncRNAs. Poucos dados sobre conservação de estrutura secundária de ncRNAs existe ainda, e tais dados, quando existentes, concentram-se em determinadas famílias de ncRNAs, ou estão em frequente refino, impossibilitando a criação de um conjunto de treinamento com número suficiente de exemplares confiáveis para o algoritmo de aprendizado. Outro problema se refere a algoritmos que necessitam de um conjunto negativo de treinamento, isto é, exemplares que não são alvo de identificação, mas que também devem ser reconhecidos pelo algoritmo para evitar erros de identificação. No caso de estruturas secundárias, não existem quaisquer dados, até o momento, a respeito de exemplares utilizáveis para esse fim (Backofen et al., 2007).

Avaliação comparativa (homologia)

Por meio de comparação de genomas entre duas ou mais espécies, as regiões de similaridade comuns a todas as espécies comparadas é submetida a outras formas de avaliação, como a termodinâmica, para inferir regiões comuns de ncRNA. Como dependem previamente de boas bases de genomas para efetuar as comparações, estes métodos são pouco viáveis na prática, e portanto, são pouco utilizados atualmente (Machado-Lima et al., 2008). Além do mais, experimentos comparativos mostram que, mesmo aliados a outras formas de avaliação, informação de homologia pode criar um viés indesejável que descarta novas proteínas, sem comparativo semelhante presente em bases de genomas, como ncRNAs.

Capítulo 3

Aprendizado de Máquina

A Seção apresenta os conceitos e definições de IA utilizados no trabalho, especificamente relacionados a algoritmos de AM e métodos para avaliação do treinamento obtido por meio de tais algoritmos.

3.1 Inteligência Artificial e Redes Neurais Artificiais

O grande objetivo da disciplina de IA é o desenvolvimento de algoritmos e paradigmas que possibilitem a execução de tarefas cognitivas por máquinas. Para este fim, um sistema de IA deve ser capaz de realizar três atividades distintas (Haykin, 1999):

- armazenar conhecimento por meio de representação de dados;
- aplicar o conhecimento armazenado na resolução de problemas, uma forma primordial de raciocínio;
- aprendizado de novos conhecimentos por meio da experiência.

Uma *rede neural artificial* é um modelo computacional inspirado no funcionamento do cérebro (Kasabov, 1998). O modelo neuronal humano pode ser visto como um sistema em três estágios: a recepção da informação, sua identificação e a decisão apropriada (Haykin, 1999). Uma unidade neuronal é representada na Figura 3.1, sendo constituída de:

- conexões de entrada (*inputs*) $x_1..x_m$;
- pesos das conexões de entrada $w_1..w_m$;
- conexão de entrada fixa w_0: uma conexão de entrada especial, com valor constante C, também chamada de peso ou *bias*;
- função de entrada $\sum$: calcula o valor agregado de entrada $u = f(x_i, w_i)$, onde x_i são as entradas e w_i seus respectivos pesos. A função basicamente efetua o somatório $u = \sum_{i=1}^{n} x_i.w_i$;
- um sinal (função) de ativação s: calcula o nível de ativação do neurônio $a = s(u)$. Esta função de ativação pode ser do tipo limiar, linear, sigmóide, hiperbólica e gaussiana (Kasabov, 1998; Mendes and Oliveira, 2009);

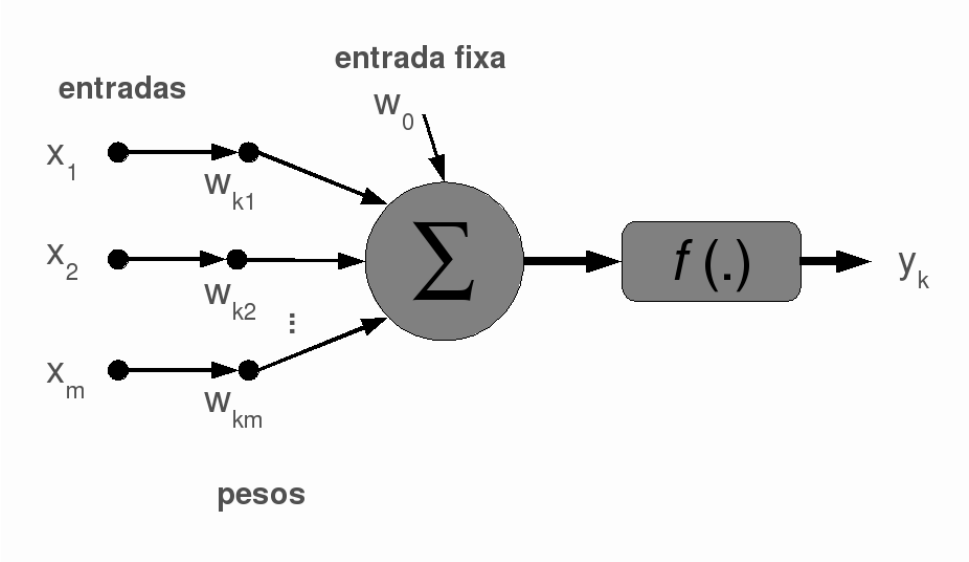


Figura 3.1: Representação de um neurônio artificial.

- uma função de saída $f()$: calcula o sinal de saída emitido pelo neurônio no seu terminal de saída $y_k = f(a)$. O sinal de saída é comumente definido pelo nível de ativação do neurônio, isto é, $y_k = a$.

Além das unidades neuronais, uma rede neural artificial é caracterizada por sua topologia (as conexões entre os neurônios), por um algoritmo de treinamento, responsável pelo aprendizado da rede, e por um algoritmo de processamento, responsável pela avaliação de novos dados tendo por base o conhecimento adquirido durante o treinamento (Kasabov, 1998).

3.1.1 Características de uma rede neural

Redes neurais associam seus neurônios artificiais refletindo a associação neuronal biológica. Alcançam, a seu modo, os critérios de inteligência identificados para uma rede neural biológica (Kasabov, 1998):

- Aprendizado e adaptação: uma rede neural é capaz de reter nova informação, moldando seu circuito por meio de supressões e estímulos a determinadas entradas do seu conjunto x_i de entradas, atingindo um estado de estabilidade. Também é capaz de adaptar-se a novas informações, alterando estes valores indefinidamente
- Generalização: redes neurais generalizam dados recebidos para incorporar uma série de características que os identificam. Essas características, mais ou menos generalizadas a critério da rede, são utilizadas para identificar novos dados
- Paralelismo massivo: Assim como o cérebro, as redes neurais são constituídas de milhares de ligações entre neurônios, e funcionalidades redundantes para neurônios. Assim sendo, o mesmo processamento pode ser realizado concomitantemente por várias unidades neuronais ligadas em paralelo

- Plasticidade: Caracteriza a robustez da rede, isto é, sua capacidade de se moldar a novas configurações espaciais por retirada ou adição de ligações ou unidades neuronais. É um resultado indireto do enorme paralelismo da rede
- Armazenamento associativo de informação: A informação é armazenada de forma a relacionar-se com outros dados já presentes na rede. Assim, ao ser exposto a uma entrada identificada qualquer, a rede é capaz de, além de recuperar a informação relativa à entrada, associá-la a outros dados e informações já presentes. Um exemplo biológico para o armazenamento associativo de informação é a capacidade de associar determinados estímulos sensoriais a eventos
- Processamento espaço-temporal de informação: Além da percepção de estímulos capturados pelas conexões de entrada dos neurônios, a rede neural pode processar informação por meio da associação com informação relativa à posição espacial de um referido dado, e acompanhar sua mudança com relação ao tempo. O comparativo ao cérebro humano é a capacidade de criar uma sucessão de eventos localizados precisamente no espaço e no tempo, um evento histórico ou identificar uma pessoa apesar de não tê-la visto há muito tempo

3.1.2 Aprendizado de máquina

Aprendizado é a capacidade de um algoritmo alterar seu comportamento frente aos estímulos recebidos do ambiente (Kasabov, 1998). Também podemos definir aprendizado de máquina como a propriedade de um dado *software* de aprender com uma experiência E relacionada a uma determinada classe de tarefas T e medida de desempenho D, se a performance de T, medida por P, melhorar com a experiência E (Mitchell, 1997).

O processo de aprendizado em uma rede neural artificial consiste no treinamento desta rede com um conjunto de entradas selecionado de forma a fazê-la aprender as características relevantes destes dados. Este conjunto especial é chamado conjunto de treinamento (Kasabov, 1998). Numa etapa de treinamento supervisionado, a rede pode ser apresentada a um conjunto de pares x_i de entrada e y_i de respostas esperadas. A rede adaptará sua função $f(a)$ para atingir estes valores. Já numa etapa de treinamento não supervisionado, a rede é apresentada somente ao conjunto de entradas x_i.

3.1.3 Conjuntos de dados de treinamento

O conjunto de entrada utilizado para o treinamento de uma rede neural ou, mais genericamente, de um algoritmo de aprendizado deve ser cuidadosamente escolhido. Utilizando a definição de Mitchell, 1997, deve-se escolher um conjunto de treinamento tal que a experiência E seja ótima para que o algoritmo aprenda a fazer T com uma performance P aceitável.

Para alcançar esse objetivo, normalmente os algoritmos de aprendizado utilizam um processo de minimização de uma função de erro $e\colon \mathbb{R}^m \to \mathbb{R}$ normalmente dependente dos atributos numéricos do conjunto de treinamento. O treinamento é satisfeito se $e < \delta$, sendo $\delta \in \mathbb{R}$ (Bishop, 1995). Outros algoritmos implementam a condição de parada do treinamento baseada somente em épocas, principalmente redes baseadas no conceito de auto organização (Kohonen, 2001).

Tabela 3.1: Conjunto de treinamento para um algoritmo de reconhecimento bayesiano de mensagens de *spam* (Thrun and Norvig, 2011).

SPAM	**Normal**
Olá fulano	Olá amigo
Hoje é dia	O dia é hoje
Imperdível compre já	Amigo, não compre isso!
Veja vídeo já	Olá meu caro

Conjuntos de treinamento são usualmente arquivos com um exemplar x_i por linha. Em dados para treinamento supervisionado, a classe à qual x_i pertence é também apresentada de alguma forma ao algoritmo. Para algoritmos não supervisionados, a informação de x_i é suficiente. É importante manter os conjuntos de treinamento balanceados, para que situações de *overfitting* ou super ajustamento (Souto et al., 2003) da rede, em que ela se especializa em reconhecer somente os dados de treinamento, não ocorram.

A importância de um conjunto de treinamento e sua utilização para aprendizado de uma dada tarefa T pode ser exemplificada por meio de um algoritmo simples de filtro de mensagens de *email* indesejadas, ou *spam* (Thrun and Norvig, 2011). Nesse filtro, um modelo M treinado para identificar palavras p de *spam* dentro de uma mensagem é treinado de forma supervisionada com o conjunto de treinamento descrito na Tabela 3.1.

As frases são igualmente distribuídas, ou seja, existem 4 exemplares de frases em mensagens de *spam* e 4 exemplares de frases em mensagens normais. O algoritmo, idealmente, deve ser capaz de responder à pergunta: "dado que uma palavra p foi identificada no texto, qual é a probabilidade de que seja SPAM ou Normal?" Para construir os modelos de probabilidade do modelo M, um sistema de contagem simples é utilizado. Por exemplo, para a palavra $p =$ "vídeo", a probabilidade pode ser estimada pela ocorrência da palavra "vídeo" em frases de *spam* e em frases normais. Sem auxílio de recursos estatísticos para suavização ou de uma taxa de aprendizado, o resultado seria que, se houver uma palavra vídeo no texto, ele será com absoluta certeza taxado como *spam*. Claramente um resultado exagerado, ocasionado por um desequilíbrio no conjunto de treinamento. Para os objetivos do trabalho, é necessário evitar situações semelhantes, selecionando de forma proporcional representantes de diferentes famílias de ncRNAs.

Poucas arquiteturas de redes neurais são capazes de tratar diretamente dados reais de um determinado problema (por exemplo, linguagem natural, sinal sonoro, pixels de uma imagem, ou sequências de RNA) (Kohonen et al., 1996a), utilizando, ao invés disso, representações matemáticas da informação que se deseja aprender, em forma de atributos numéricos, normalmente. O cuidado de análise e normalização de tais atributos é mister para identificar problemas no conjunto, ou diminuição de ruídos indesejáveis na rede, sem a necessidade de ajustes finos nos parâmetros da rede.

Uma preocupação constante na criação do conjunto de treinamento para dados biológicos é com a baixa qualidade dos dados utilizados, ou com a contaminação do conjunto codificante com excertos não codificantes, ou vice versa. Tal contaminação, experimentalmente comprovada (Eddy, 2001; Frith et al., 2006; Szymanski et al., 2007; Dinger et al., 2008), é devida às dificuldades explanadas no Capítulo 2 com relação à identificação de ncRNAs. Particularmente, estudos provaram que alguns transcritos não codificantes que atuam no processo celular em algum grau podem também conter trechos codificantes, que

são expressos como peptídeos no organismo, e, de forma simétrica, mRNA pode exercer também atividades funcionais dentro do organismo (Ulveling et al., 2011). O desafio de identificar tais grupos de transcritos de dupla função é enorme.

3.1.4 Avaliação do treinamento

O objetivo ao construir um bom conjunto de treinamento, como visto anteriormente, é o de escolher bons exemplos para cada classe e subclasse do problema, conforme se julgar necessário, que se quer tratar. De forma complementar ao treinamento, a avaliação, ou validação, do treinamento é constituída para ajuste de parâmetros estruturais do treinamento da rede, de forma que a rede obtenha a melhor performance possível nesse conjunto de validação. Normalmente, as medidas de performance mais comumente utilizadas para validar o treinamento de um algoritmo de aprendizado de máquina são acurácia e especificidade. Acurácia representa a precisão de um algoritmo, sua capacidade de identificar corretamente as classes interessantes ao problema. Já especificidade representa a capacidade inversa de identificar corretamente a classe, ou classes, que não são o alvo primário do problema. A representação do conhecimento obtida no treinamento da rede não é alvo de alteração dessa etapa, mas sim, normalmente, uma série de modificações para maximizar a acurácia do classificador. Os exemplares para validação não podem ser os mesmos do treinamento, por motivos óbvios.

É importante frisar a diferença entre conjunto utilizado para validação e o conjunto utilizado para teste. O conjunto de validação age de forma complementar ao treinamento, o que normalmente induz a rede a uma especialização com relação a esse conjunto, uma situação semelhante a super ajustamento (Bishop, 1995). O conjunto de teste é constituído por exemplares não pertencentes a nenhum dos conjuntos de treinamento ou validação. Normalmente, ele representa uma situação real de funcionamento do algoritmo, principalmente no que diz respeito à introdução de erros e ruídos inerentes à coleta de informações do ambiente por meio de sensores (Thrun and Norvig, 2011). Nesse sentido, o conjunto de teste verifica o grau de generalização da rede com relação ao conhecimento apreendido por meio da etapa de treinamento.

Múltiplos métodos existem para validar e auxiliar um treinamento. Nesse trabalho, empreende-se esforços utilizando os métodos por matrizes de confusão e avaliação de acurácia, especificidade, medida F, coeficiente de correlação de Matthews (Matthews, 1975). Além disso, um estudo detalhado dos atributos numéricos utilizados na rede é realizado utilizando análise de componentes principais. Para SOM, validação dos agrupamentos por meio da análise de u-matrizes (Hollmen, 2009) também é explicada. Outros métodos consagrados para validação de algoritmos de aprendizado de máquina também são brevemente explanados nesse Capítulo, porém não serão utilizados.

Matrizes de confusão

Matrizes de confusão têm por objetivo a representação da decisão de um determinado classificador com relação a um conjunto de validação. Tal decisão é avaliada, principalmente, em termos da acurácia do classificador (Stehman, 1997).

A Tabela 3.2 apresenta a organização clássica de uma matriz de confusão para um classificador binário qualquer. O classificador ideal deve ser capaz de diferenciar, dentro do conjunto de validação, os exemplares positivos dos exemplares negativos, de forma

Tabela 3.2: Exemplo de matriz de confusão para um classificador binário, com a definição de Verdadeiros Positivos (VP), Falsos Positivos (FP), Verdadeiros Negativos (VN) e Falsos Negativos (FN).

		Classificação	
		Positivo	Negativo
Real	Positivo	Verdadeiro Positivo	Falso Negativo
	Negativo	Falso Positivo	Verdadeiro Negativo

inequívoca. O conjunto de validação, para esse fim, é constituído por exemplares positivos e negativos.

Na matriz representada, os exemplares são distribuídos entre as 4 células disponíveis de acordo com o resultado da predição. Nas linhas nomeadas "Real", os dados reais de cada exemplar são levados em conta. Idealmente, o conjunto de validação deve conter exemplares de cada classe em quantidade igual ou pelo menos proporcional. Nas células "Verdadeiro Positivo" e "Verdadeiro Negativo" encontram-se os exemplares que foram corretamente identificados pelo algoritmo. Por meio desses dados experimentais, é possível extrair uma série de informações sobre a acurácia e o desempenho geral do classificador. Preferencialmente, medidas de acurácia que possibilitam a identificação de erros e acertos na classificação devem ser utilizadas em detrimento de medidas mais generalizadas, sua escolha baseada principalmente no objetivo que se deseja alcançar (Stehman, 1997).

As taxas de Verdadeiros Positivos (VP), Falsos Positivos (FP), Verdadeiros Negativos (VN) e Falsos Negativos (FN), extraídas de matrizes de confusão por meio da consulta do número de exemplares em determinada célula da matriz, possibilitam a extração de uma variedade de medidas relacionadas à precisão e especificidade do algoritmo. As Equações 3.1, 3.2 3.3 3.4, 3.5 e 3.6 definem, respectivamente, a taxa de **precisão**, a **acurácia**, a **especificidade**, a **medida F** harmônica e o **coeficiente de correlação de Matthews** (Matthews, 1975).

$$Prec = \frac{VP}{VP + FP} \tag{3.1}$$

$$Rec = \frac{VP}{VP + FN} \tag{3.2}$$

$$Acc = \frac{VP + VN}{VP + FP + VN + FN} \tag{3.3}$$

$$Spec = \frac{VN}{VN + FP} \tag{3.4}$$

$$FM = 2 \times \frac{Prec \times Rec}{Prec + Rec} \tag{3.5}$$

$$MCC = \frac{(VP \times VN) - (FP \times FN)}{\sqrt{(VP + FP) \times (VP + FN) \times (VN + FP) \times (VN + FN)}} \tag{3.6}$$

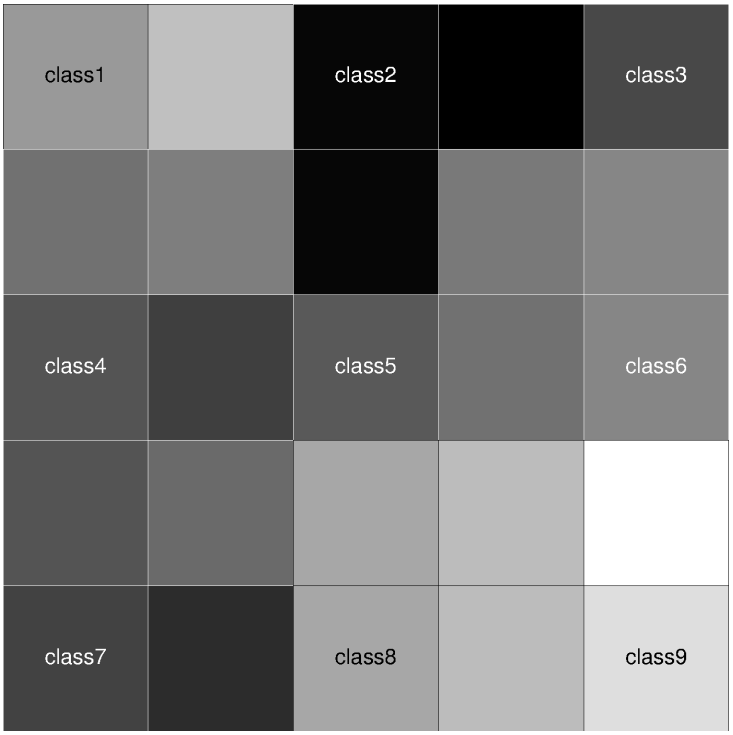


Figura 3.2: Exemplo de U-Matriz para uma rede SOM treinada com 9 neurônios dispostos em topologia retangular.

A grande maioria das medidas apresentadas relaciona-se com precisão, o que caracteriza um comportamento, ou procedimento, vinculante entre o desempenho de um algoritmo de aprendizado de máquina e a capacidade de identificar a classe principal de uma determinada tarefa T. Porém, em problemas de classificação, classes heterogêneas acontecem em grande frequência, normalmente com algumas classes principais sobressaindo, em quantidade de exemplares, a pequenas classes que aprenderam com outros estímulos menos frequentes, porém não menos importantes. Para tentar minimizar esse efeito, utiliza-se a medida de coeficiente de correlação de Matthews, capaz de minimizar o efeito da precisão em benefício de outras classes secundárias utilizadas no classificador (Matthews, 1975). Mais detalhadamente, o coeficiente orbita dentro dos valores $-1 \leq MCC \leq 1$, em que 1 representa um classificador perfeitamente sensível aos dados positivos do conjunto, -1 um classificador perfeitamente específico aos dados negativos do conjunto, e 0 um classificador aleatório.

Uma *U-matriz* (Hollmen, 2009) é uma representação gráfica do espaço de decisão gerado pelo treinamento da rede SOM. Ela representa a matriz correspondente à camada de saída $\bar{y}(n)$, de forma semelhante à ilustrada na Figura 3.2, gerada pela aplicação do procedimento *umatrix* da biblioteca SOM_PAK a uma rede SOM treinada com dados reais de ncRNAs e mRNAs.

Na imagem, o código de cores representa a distância entre os neurônios, ou, de forma equivalente, a distância entre os centróides dos *clusters*. A distância normalmente é euclideana (Kohonen et al., 1996b). Regiões mais escuras representam distâncias maiores entre os nós, e regiões mais claras indicam agrupamentos vicinais de nós. Para a Figura 3.2, existe um agrupamento visível entre as classes 5, 6, 8 e 9, enquanto que a classe 1 está

separada desse agrupamento por uma fronteira formada pelas classes 2, 3, 4, 5 e 7.

Outra medida de validação importante é o erro de quantização médio (*quantization error* ou *qerror*), calculado por meio da Equação 3.7 (Kohonen et al., 1996b).

$$Qerror = \sqrt{\frac{\sum \parallel \bar{x} - \bar{y} \parallel^2}{N}} \tag{3.7}$$

Em que N refere-se ao número de exemplares do conjunto de validação utilizado. Altos valores de $Qerror$ indicam um desvio considerável do neurônio escolhido y_i. No cálculo de *qerror* podem ser introduzidos pesos e critérios para escolha dos neurônios da camada de saída (Kohonen et al., 1996b).

Análise de Componentes Principais

Análise de Componentes Principais (PCA) é uma técnica estatística proposta primeiramente por Pearson, 1901 com propósito geral, e sucessivamente refinada para o problema de redução de dimensionalidade do vetor de atributos em IA e AM (Haykin, 1999). A técnica baseia-se na redução de dimensionalidade de um determinado vetor de atributos x_i por meio da minimização do erro quadrático médio causado por esse procedimento de excisão. Ou seja, supondo um vetor x_i de i dimensões que se deseja reduzir para $l < i$ dimensões, a técnica de PCA minimiza a perda de informação representada pela soma:

$$\sum_{j=i-l+1}^{i} \sigma_j^2$$

Onde σ_j^2 é a variância associada ao j^o atributo retirado de x_i. Por meio da adoção de uma transformação linear T, a retirada dos menores valores de variância do vetor transformado Tx_i alcança tal objetivo (Haykin, 1999).

PCA assume a média $\bar{x_i}$ de x_i como nula, subtraindo sua média caso não seja. A transformação T aplicada é feita projetando-se, então, o vetor x_i num vetor unitário q_i também de dimensão i. Tal projeção tem as seguintes propriedades:

$$T = x_i^T q_i = q_i^T x_i \tag{3.8}$$

$$E[x_i] = 0 \tag{3.9}$$

$$E[T] = q_i^T E[x_i] = 0 \tag{3.10}$$

Onde E é o valor esperado da variável. De posse das Propriedades 3.8 e 3.10, a variância σ_i^2 é calculada da seguinte forma:

$$\begin{aligned} \sigma_i^2 &= E[T^2] && (3.11)\\ &= E[(q_i^T x_i)(x_i^T q_i)] && (3.12)\\ &= q_i^T E[x_i x_i^T] q_i && (3.13)\\ &= q_i^T R q_i && (3.14) \end{aligned}$$

Tabela 3.3: Tabela de exemplo dos resultados da PCA de um conjunto de teste.

Dimensão	Autovalor	**Variância**	**Variância Cumulativa**
Dim1 (A)	1.8	44.96%	44.96%
Dim2 (C)	1.43	35.69%	80.65%
Dim3 (G)	0.58	14.42%	95.06%
Dim4 (T)	0.2	4.93%	100%

Onde R é a matriz $i \times i$ de **correlação** do vetor x_i. Pode-se formalizar a variância σ_i^2 como uma função em termos do vetor q_i, redefinindo a Equação 3.14 para a Equação 3.15.

$$\psi(q_i) = q_i^T R q \tag{3.15}$$

A partir da Equação 3.15, o problema resolve-se encontrando os valores de q_i em que $\psi(q_i)$ tem valores extremos. A Equação 3.16 resume o sistema de resolução desse problema, incluindo o conceito de *eigenvalor*, ou autovalor, de um vetor.

$$Rq_i = \lambda_i q_i \tag{3.16}$$

O autovalor λ_i que resolve a Equação 3.16 para o autovetor q_i reflete o valor extremo da variância de $\psi(q_i)$, enquanto que q_i representa a direção em que ocorre esse valor extremo. O processo de solução da Equação 3.16 pode envolver aprendizado de máquina, entre outros métodos (Haykin, 1999). Tais métodos utilizam cada exemplar j do conjunto de avaliação para extrair os valores de autovetores e autovalores. Posteriormente, ordena-se, para cada atributo i, os respectivos λ_i. Pode-se assim extrair os de menor valor para reduzir a dimensão de x_i. Por meio de uma análise gráfica de q_i, é possível também avaliar o grau de correlação entre atributos (Lê et al., 2008). É importante frisar que, para a correta construção da matriz de correlação R, os valores de x_{ij} dos atributos de cada exemplar j do conjunto de avaliação devem ser normalizados.

Representações gráficas de PCA são importantes para decisão do grau de correlação entre dois atributos. Tal correlação pode indicar atributos redundantes ou atributos complementares, informação crucial para uma redução de dimensionalidade bem sucedida. Entre várias soluções possíveis, a avaliação de um mapa bidimensional de autovetores associados a seus autovalores normalizados como exibido na Figura 3.3 é contribuição essencial para tal procedimento.

A biblioteca FactoMineR (Lê et al., 2008) foi utilizada para gerar o gráfico de análise de PCA para 4 atributos numéricos extraídos de um conjunto de 10 sequências codificantes e 10 sequências não codificantes, criado para fins de exemplificação.

Cada atributo tem seu autovetor e autovalor calculados, arranjados em ordem decrescente e exibidos num gráfico em relação às duas primeiras dimensões. As porcentagens referem-se à relevância em termos de variabilidade de σ^2 dos dois primeiros atributos. Na análise da Figura 3.3, os dois atributos $Dim1$ e $Dim2$, combinados, representam 80.65% do total de variância cumulativa nos dados de teste. O resumo dos autovalores, variância e variância cumulativa é dado na Tabela 3.3. Observa-se que a variância é dada em forma percentual, normalizada para o conjunto de atributos.

Uma medida muito utilizada para selecionar bons atributos é escolher dimensões k que tenham autovalores $\lambda_k > 1$ (Lê et al., 2008). Na Figura 3.3, cada atributo numérico

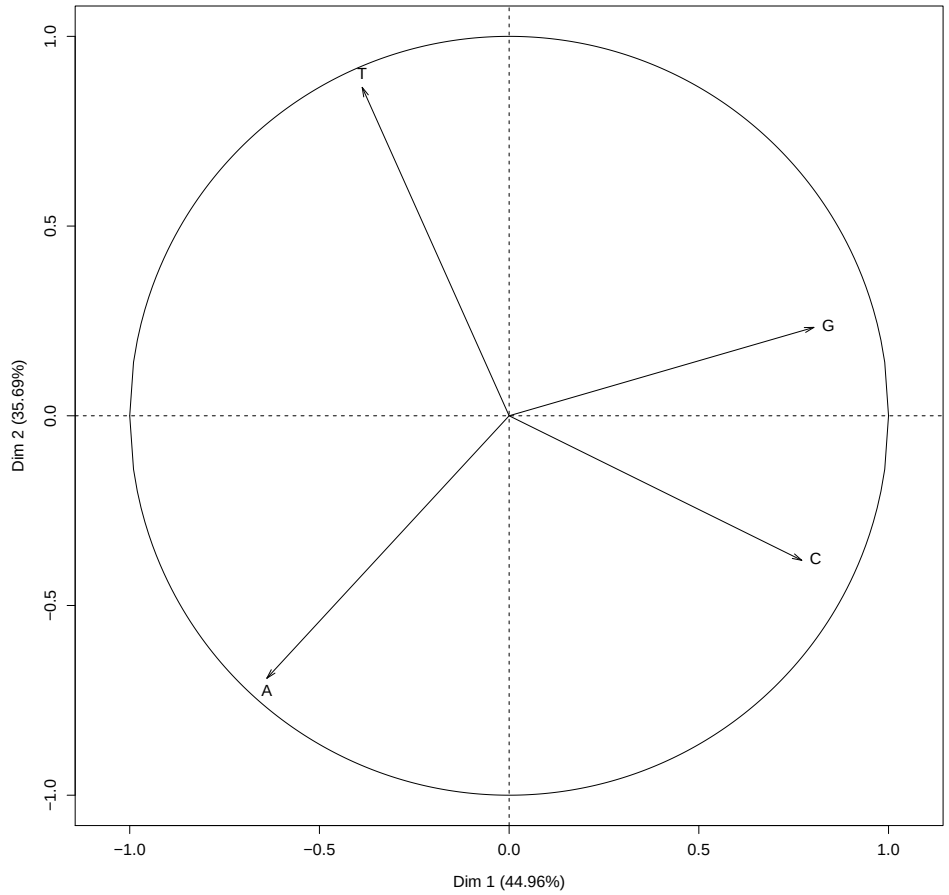


Figura 3.3: Gráfico com resultados da PCA para um conjunto de teste com 20 sequências aleatórias e 4 atributos numéricos, ilustrando o mapa de autovetores normalizados por seus respectivos autovalores.

i é representado por seu autovetor bidimensional com direção dada por $\psi(q_i)$ e módulo proporcional ao autovetor λ_i. Dados dois atributos a e b quaisquer, a projeção $\pi(q_a, q_b)$ representa a correlação entre os dois atributos, com $\pi(q_a, q_b) \in [0, 1]$. Quanto maior $\pi(q_a, q_b)$, maior é a correlação entre a e b. Em contrapartida, valores próximos de zero ou nulos representam dois valores com baixo grau de correlação entre si. No exemplo dado, o gráfico pode ser claramente dividido em dois hemisférios, leste e oeste. No hemisfério oriental, os atributos C e G, respectivamente $Dim2$ e $Dim3$ trabalham de forma correlata. No hemisfério ocidental, A e T trabalham de forma bastante correlata. Isso reforça o senso comum explanado no Capítulo 2 sobre a complementaridade entre bases purinas e pirimidinas. A conclusão importante desse fato é que, mesmo com um autovalor baixo e pouca variabilidade nos dados de teste, a retirada do atributo $Dim4$ poderá causar um efeito negativo maior do que o esperado no classificador, por causa de sua correlação com o atributo $Dim1$.

Nesse trabalho, matrizes de confusão, análise de u-matrizes e erros de quantização são os métodos escolhidos para validação do treinamento da rede. Argumenta-se sobre essa escolha a facilidade de implementação dessas medidas e a cobertura de diversos compor-

tamentos desejáveis para a rede, a saber: alta precisão, boa separação dos agrupamentos existentes no conjunto de treinamento utilizado e boa confiabilidade da predição, por sua vez estimada pelo valor do erro de quantização associado à predição.

Outros métodos

Como dito anteriormente, a grande maioria das técnicas de validação têm por objetivo refinar o treinamento, otimizando parâmetros de ajuste da rede para melhorar a acurácia do classificador, com relação a um conjunto de treinamento e a um conjunto de validação. Nesse ínterim, obter exemplares suficientes para satisfazer essas duas etapas pode ser difícil e oneroso. Técnicas de validação cruzada utilizam um procedimento simples em que o conjunto de treinamento X, com j exemplares, é dividido em m partes iguais, numa divisão estratificada, ou seja, com exemplares tomados randomicamente, observando-se a distribuição igualitária de classes, quando presentes (Hsu et al., 2010). Treina-se o algoritmo com $m - 1$ partes do conjunto de treinamento, e valida-se com a parte restante. Itera-se até que todas as partes tenham sido usadas na validação, e escolhe-se o melhor resultado obtido. O procedimento, por ser automatizado, permite iterações mais aceleradas do procedimento de treinamento e validação, permitindo um controle maior de toda a etapa. Entretanto, a escolha aleatória de exemplares pode ser suplantada pela seleção mais minuciosa dos dados de treinamento e validação, procedimento preferido nesse trabalho.

Métodos iterativos e híbridos para avaliação do comportamento de uma rede são utilizados, dependendo grandemente das características do problema abordado, das limitações e particularidades do algoritmo de aprendizado utilizado, da quantidade e qualidade dos exemplares utilizados para treinamento, entre outros. Como exemplo, pode-se citar o recente trabalho realizado por Taira et al., 2011 para reduzir o número de atributos numéricos em um algoritmo de SVM com *kernel* radial. O procedimento é sintetizado pelo Algoritmo 3.4.

Nesse procedimento, a variação de acurácia δ calculada é ajustada para cada passo de extração de um atributo, ou cumulativamente, até um valor estabelecido. O Algoritmo 3.4, apesar de simples e pouco otimizado, resultou na melhoria expressiva do método em questão, com a redução de $\approx 60\%$ do número de atributos incorrendo em uma redução de $\approx 4\%$ na acurácia.

3.2 Mapas Auto Organizáveis

Em uma rede neural baseada em Mapas Auto Organizáveis (*SOM* ou *Self Organizing Maps*), as unidades neuronais são dispostas como nós de um mapa de coordenadas euclideanas. A rede neural recebe por entrada estímulos sob forma de vetores n-dimensionais e os posiciona em um mapa discreto de dimensões reduzidas, geralmente mono, bi ou tridimensionais (Haykin, 1999; Kohonen, 2001). O modelo de funcionamento de mapas auto-organizáveis proposto por Kohonen é o mais difundido e utilizado, por permitir uma abordagem mais generalizada do problema de criação de mapas computacionais ordenados topologicamente utilizando redução dimensional nos dados de entrada (Haykin, 1999). A Figura 3.5 representa um mapa auto organizado bidimensional do modelo de Kohonen.

Input : Conjunto de treinamento X
Input : Número de divisões k
Output: Algoritmo treinado
Data: Conjunto de atributos relevantes M
Data: Variação da acurácia δ

```
1  Inicializar algoritmo de treinamento;
2  M ← ∅;
3  for cada atributo i do
4  |  Treinar algoritmo com i;
5  |  Treinar algoritmo sem i;
6  |  ACC ← Validar algoritmo com i;
7  |  ACC' ← Validar algoritmo sem i;
8  |  if ‖ACC' − ACC‖ ≤ δ then
9  |  |  M ← M ⋃ i;
10 |  end
11 end
12 Treinar algoritmo com conjunto M;
```

Figura 3.4: Algoritmo simples para remoção de atributos com baixo impacto na acurácia utilizando procedimento de validação cruzada em algoritmo de regressão baseado em Máquinas de Vetor de Suporte com kernel radial (Taira et al., 2011).

A camada de entrada da SOM, composta por unidades sem capacidade computacional, admite um vetor de entrada $\bar{x}(n) = [x_1(n)x_2(n)\dots x_i(n)]$, que geralmente é uma série de entradas numéricas contínuas ou discretas (Kasabov, 1998). n refere-se ao número de épocas do algoritmo de aprendizado da rede.

A camada de saída é formada pelas saídas $y_1(n), y_2(n), \dots, y_j(n)$ das unidades neuronais (nós) que compõem o mapa auto organizado. Os vetores de peso, ou *vetores de referência* $\bar{w}_j(n) = [w_{1,1}(n)w_{1,2}(n)\dots w_{i,j}(n)]$, conectam cada vetor de entrada $x_i(n)$ com todos os nós $y_j(n)$ do mapa. Na figura 3.5, a topologia que ordena os nós sobre o mapa bidimensional é a *retangular*, evidenciada pelas ligações dos quatro neurônios superiores. Existem várias outras topologias utilizadas em mapas auto organizáveis, com destaque para a topologia hexagonal, muito empregada em máquinas de reconhecimento de padrões visuais (Haykin, 1999). O Algoritmo 3.6 mostra o processo de aprendizado de um mapa auto organizável de Kohonen.

Em suma, o processo de aprendizado de uma rede neural baseada em SOM segue três etapas importantes: a etapa de *Competição*, a etapa de *Cooperação* e a etapa de *Adaptação dos Pesos Sinápticos*. As linhas 6 e 8 do Algoritmo 3.6 correspondem à etapa competitiva de treinamento. Já cálculo de V_m no final da etapa 8 corresponde à etapa cooperativa entre os neurônios. No passo de adaptação dos vetores de referência (linha 10 do algoritmo), a função de vizinhança $hj(m)$ escolhida pode ser de diversos tipos, sendo mais comumente empregado a função *gaussiana* (Haykin, 1999).

A etapa adaptativa do algoritmo pode ser ainda subdividida em uma etapa de ordenação do mapa e uma posterior etapa de convergência. Na etapa de ordenação, os pesos

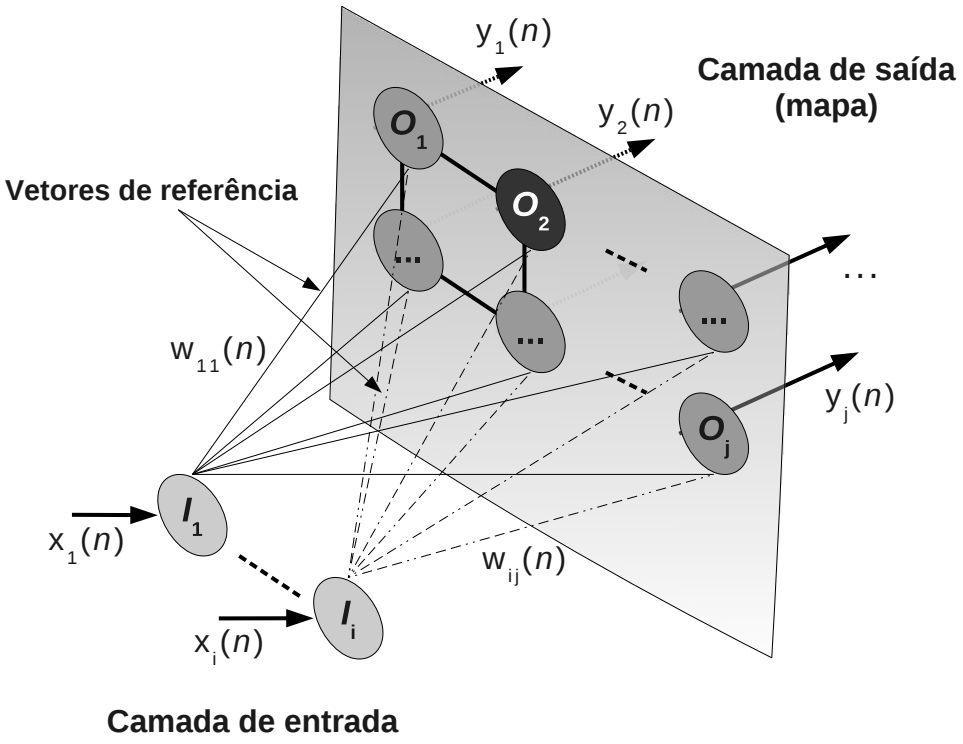


Figura 3.5: Arquitetura simplificada de uma rede neural baseada em mapas auto organizáveis de Kohonen e seus principais componentes.

$\bar{w}_j(m)$ são organizados e ordenados no mapa, o que requer em torno de 1.000 iterações do algoritmo de treinamento (Haykin, 1999). Já na etapa de convergência, ocorre uma verificação detalhada da configuração da rede, utilizando uma quantidade bastante superior de iterações e uma taxa de aprendizado $\alpha(m)$ comparativamente menor do que a utilizada na etapa de ordenação (Kohonen et al., 1996b).

3.3 Learning Vector Quantization

Learning Vector Quantization (LVQ) é um método de aprendizado de máquina supervisionado utilizado para melhorar as regiões de decisão de um classificador (Haykin, 1999). Regiões de decisão, por sua vez, são as fronteiras existentes entre as classes do classificador. Um exemplo de tal região é o hiperplano criado por um algoritmo de regressão baseado em SVM, ou as regiões criadas por um algoritmo de k-médias. A Figura 3.7 representa um espaço de decisão arbitrário de um classificador.

No algoritmo LVQ, o espaço de decisão, representado na Figura 3.7 por uma região bidimensional é dividido em regiões, identificadas pelas linhas tracejadas. Na Figura, os círculos representam os centros das regiões. O algoritmo age no espaço definindo um vetor para representar a região inteira, chamado *vetor de Voronoi*, num procedimento de compressão dos dados de treinamento (Haykin, 1999) ou quantização dos estímulos

Input : vetor de entrada $\bar{x}(n)$
Output: camada de saída organizada $\bar{y}(n)$
Data: vetor de referência $\bar{w}_j(n)$

1 Inicializar o vetor de entrada, $\bar{w}_j(0)$, com pequenos valores randômicos;
2 $m \leftarrow 0$;
3 **while** *época* $m < n$ **do**
4 Forneça um vetor de entrada $\bar{x}(m)$;
5 **for** *cada neurônio* j **do**
6 Calcule a distância Δ_j (num espaço i-dimensional) entre $\bar{x}(m)$ e $\bar{w}_j(m)$. Num espaço Euclideano podemos definir: $\Delta_j = \sum(\sqrt{(\bar{x}(m) - \bar{w}_j(m))^2})$;
7 **end**
8 O neurônio com função de ativação $y_k(m)$ mais próximo de $\bar{x}(m)$ é declarado o neurônio vencedor. Torna-se o centro da vizinhança V_m;
9 **for** *cada neurônio* j **do**
10 **if** $j \in V_m$ **then**
11 $\bar{w}_j(m+1) = \bar{w}_j(m) + hj(m)\alpha(m)(\bar{x}(m) - \bar{w}_j(m))$;
12 **end**
13 **else**
14 $\bar{w}_j(m+1) = \bar{w}_j(m)$;
15 **end**
16 onde $hj(m)$ é a função de vizinhança; ela depende da distância entre o neurônio vencedor e o j^o neurônio. Ela diminui com o tempo (épocas), e normalmente uma função gaussiana é utilizada;
17 **end**
18 $hj(m)$, taxa de aprendizado $\alpha(m)$ e V_m diminuem com as épocas;
19 $m \leftarrow m + 1$;
20 **end**

Figura 3.6: Passos do algoritmo de aprendizado de um mapa auto organizável de Kohonen (Kasabov, 1998).

de entrada de acordo com sua região no espaço de decisão. O Algoritmo 3.8 representa os principais passos para codificação de um espaço de decisão qualquer em vetores de Voronoi.

Assumindo um espaço de decisão de um algoritmo composto por vetores de referência w_j corretamente ajustados, o algoritmo LVQ aproxima iteradamente um estímulo de entrada $\bar{x}$ do seu mais próximo valor quantizado. Para todos os outros vetores quantizados não estimulados por $\bar{x}$, nada é feito. Tal aproximação é feita escolhendo-se, para cada estímulo $\bar{x}$, o vetor de Voronoi de índice c mais próximo.

$$c = \underset{j}{\operatorname{argmin}}\{\| \bar{x} - w_j \|\} \tag{3.17}$$

Existem várias formas para encontrar o vetor de Voronoi que satisfaz a Equação 3.17. A biblioteca LVQ_PAK (Kohonen et al., 1996a), utilizada nesse trabalho, o encontra por meio da aplicação do algoritmo k-vizinhos mais próximos (*k-nearest neighbours* ou KNN).

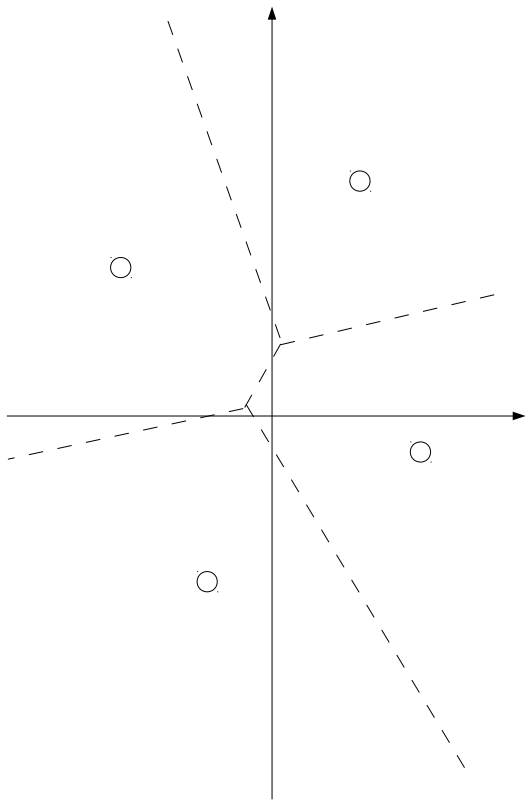


Figura 3.7: Espaço de decisão de um classificador em 4 classes para um problema qualquer (Haykin, 1999).

Como entrada do algoritmo LVQ, pode-se utilizar o mapa auto organizável treinado pelo algoritmo SOM. Pode-se perceber, portanto, que o objetivo de tal algoritmo não é o de aproximar a função de densidade associada a um determinado conjunto X de treinamento num espaço de reduzida dimensionalidade, mas sim o de definir fronteiras no espaço treinado de acordo com os dados de treinamento fornecidos ao algoritmo e ao próprio espaço de decisão (Kohonen et al., 1996a).

A taxa de aprendizado converge rapidamente nessa implementação otimizada do algoritmo feita pela biblioteca LVQ_PAK (Kohonen et al., 1996a). Uma boa escolha para a taxa inicial de treinamento é $\alpha_c(0) < 0.1$. Já o número de épocas é estimado em torno de 30 a 50 vezes o número de vetores de referência no espaço de decisão.

Aplicado ao problema de classificação de ncRNAs, o algoritmo LVQ permite o ajuste fino dos espaços de decisão determinados pelo algoritmo SOM. Por sua vez, tal refino contribui para a especialização de determinados *clusters* para uma família ou grupo de ncRNAs de interesse.

3.4 Teoria da Ressonância Adaptativa

Teoria da Ressonância Adaptativa (**ART** ou *Adaptive Resonance Theory*) é uma família de redes neurais auto organizáveis, comumente de treinamento não supervisionado. Ela

Input : vetor de referência $\bar{w}_j(n)$ do espaço de decisão
Input : vetores de entrada $\bar{x}_i(n)$
Output: vetores quantizados organizados $\bar{w}_j(n)$

1 $m \leftarrow 0$;
2 **while** *época* $m < n$ **do**
3 | Forneça um vetor de entrada $\bar{x}(m)$;
4 | $c = \text{argmin}_j\{\| \bar{x}(m) - w_j(m) \|\}$;
5 | $\bar{w}_c(m+1) = [1 - s(m)\alpha_c(m)] \cdot w_c(m) + s(m)\alpha_c(m)\bar{x}(m)$;
6 | onde $s(m) = 1$ se a classificação é correta, e $s(m) = -1$ se é incorreta. A taxa de aprendizado $\alpha_c(m)$ diminui com as épocas;
7 | $m \leftarrow m + 1$;
8 **end**

Figura 3.8: Passos do algoritmo otimizado de Learning Vector Quantization (Kohonen et al., 1996a; Haykin, 1999).

resolve eficientemente o problema de plasticidade *versus* estabilidade de uma rede neural auto organizável (Carpenter and Grossberg, 1987). O problema em questão relaciona a capacidade de estabilizar a assimilação de informação de um conjunto de treinamento com a representação desses dados na camada k-dimensional de saída. Mais precisamente, as redes ART é capaz de criar *clusters* agrupando dados conforme um valor de vigilância ρ dado. O algoritmo é muito utilizado para conceitualização, agrupamento em *clusters* e descoberta de tipos e número de classes em um conjunto de dados (Kasabov, 1998). A Figura 3.9 representa a arquitetura simplificada de uma rede da família de redes ART.

A camada de entrada da ART é composta também por unidades sem capacidade computacional, admitindo um vetor de entrada $I = (i_1, i_2 \dots i_m)$, que geralmente é uma série de entradas numéricas contínuas ou discretas (Frank et al., 1998). Cada neurônio da camada de saída $O = (o_1, o_2 \dots o_n)$ recebe um vetor de referência $W_1 = (w_{11} \dots w_{1m}) \dots W_n = (w_{n1} \dots w_{nm})$, chamados protótipos. Essencialmente, alterações feitas em w_{nm} refletem-se em w_{mn} e vice-versa. O algoritmo detalhado de uma rede ART-2E é ilustrado na Figura 3.10.

A rede ART difere estruturalmente da rede SOM ao incluir uma etapa de busca por um protótipo w_{jp} que satisfaça um critério de semelhança com uma entrada i_p, o que normalmente é obtido pela distância euclideana entre os dois vetores. Inicialmente, somente um protótipo (uma classe) retrata o espaço de busca da rede. À medida que a condição na linha 24 se repete, mais classes são geradas, sempre respeitando o limite de vigilância $\rho \in [0, 1]$ fornecido à rede. O parâmetro ρ, por sua vez, atua da seguinte forma: aproximando-se de 1, o algoritmo ativará mais vezes o neurônio "Reset", causando mais frequentemente a ressonância em um novo protótipo. Em $\rho = 1$, o algoritmo criará tantos protótipos quantos forem os estímulos de entrada, independente de sua diferença. Contrariamente, valores muito próximos de zero tornarão a ressonância em protótipos já existentes mais frequente, até o valor $\rho = 0$ em que todos os estímulos de entrada serão acumulados num único neurônio.

O critério de parada de algoritmos ART podem diferir conforme diferentes implementações. Normalmente, dada a forma de aprendizado hebbiano em sistemas auto organizáveis

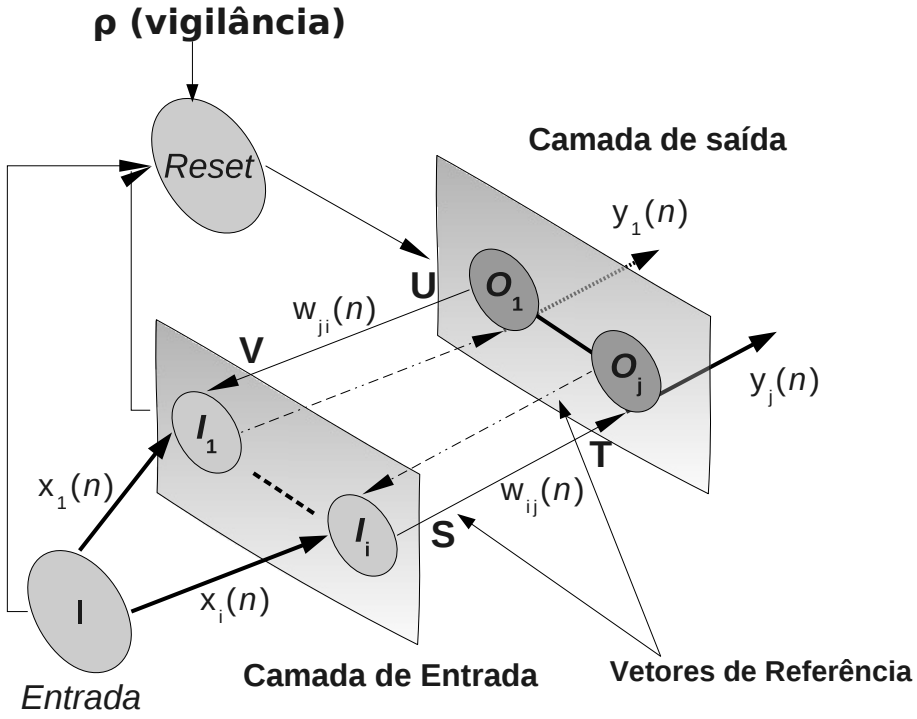


Figura 3.9: Arquitetura de uma rede da família de redes ART.

de tais redes, utiliza-se uma abordagem consoante com a abordagem SOM. Também é possível monitorar a quantidade de estímulos de entrada que tem seu protótipo de referência alterado, num procedimento de flutuação. A razão entre a quantidade de tais estímulos e a quantidade total de estímulos no conjunto de treinamento é utilizada, por exemplo, na biblioteca ART_distance (Hudik and Zizka, 2011).

A adoção de redes ART para o problema de classificação de ncRNA é especialmente importante por causa de sua extensão ao processo de aprendizado competitivo, evidente na competição entre os neurônios da camada de saída pelo estímulo de entrada no passo 5. Esse processo tende a gerar uma categorização instável, em que o treinamento não converge suficientemente, na medida em que os vetores de entrada divergem muito entre si. Os passos 13 a 21 evidenciam esse algoritmo diferenciado de busca por protótipos W já existentes que se assemelhem ao estímulo i_p recebido.

3.5 Auto Organização em Bioinformática

Métodos não supervisionados já têm aplicação prática no campo da expressão gênica (Eisen et al., 1998) principalmente para análise de agrupamentos (*cluster*) de dados. Tradicionalmente, métodos SOM e ART são aplicados para solução de problemas de reconhecimento de padrões e predição de padrões, como análise de imagens, reconhecimento visual e sonoro, entre outros (Haykin, 1999; Kasabov, 1998).

Input : vetor de entrada I
Output: camada de saída O
Data: protótipo W

1 O vetor de entrada I é normalizado para valores $\in [0, 1]$;
2 Inicializar o protótipo W com valores de entrada I quaisquer ou com valores randômicos ≥ 1;
3 O primeiro protótipo w_{11} é marcado como visitado;
4 **for** *cada entrada* $i_p \in I$ **do**
5 Calcule a atividade da rede $t_j = 1 - \sqrt{\frac{1}{m}\sum_{i=1}^{m}(i_i - w_{ji})^2}$;
6 Encontre o neurônio vencedor o_j:
7 **if** $t_j > max(t_k : k \neq j)$ **then**
8 $o_j = 1$;
9 **end**
10 **else**
11 $o_j = 0$;
12 **end**
13 **if** w_{jp} *já foi visitado* **then**
14 **if** $t_j > \rho$ **then**
15 O sistema entra em ressonância, ou seja, o protótipo é visitado e adaptado à entrada i_p:
16 $w_{pj}^{novo} = \eta \cdot i_p + (1 - \eta) \cdot w_{pj}^{velho}$, onde $\eta \in [0, 1]$ é a taxa de aprendizado da rede.;
17 **end**
18 **else**
19 $o_j = 0$;
20 Volta ao passo 6;
21 **end**
22 **end**
23 **else**
24 O sistema também entra em ressonância, o protótipo w_{jp} é visitado, e $\eta = 1$:
25 $w_{pj}^{novo} = i_p$;
26 **end**
27 **end**

Figura 3.10: Passos do algoritmo de aprendizado de uma rede ART 2_E (Kasabov, 1998; Frank et al., 1998).

SOM e ART são métodos tradicionalmente não supervisionados (Kasabov, 1998; Carpenter and Grossberg, 1987) que utilizam o conceito de Auto Organização. Auto Organização de uma rede neural fundamenta-se na modificação dos pesos das conexões entre os neurônios da rede, até que uma configuração global se estabeleça. Essa modificação, normalmente, é alcançada por meio do aprendizado não supervisionado de um conjunto de treinamento. Como descrito na Seção 3.1, redes neurais artificiais recebem um estímulo x_i de entrada. Em redes SOM e ART, esses estímulos são vetores finitos de valores

Tabela 3.4: Tabela de atributos comumente utilizados em identificadores e classificadores de ncRNAs.

Atributo	**Programa que o utiliza**
Composição de nucleotídeos	CONC, Portrait, SOM-Portrait
Tamanho de ORF	CONC, Portrait, CPC, SOM-Portrait
Tamanho da sequência	CONC, Portrait, CPC, SOM-Portrait
Composição de aminoácidos	Portrait, SOM-Portrait
Homologia com proteínas	CPC
Informação termodinâmica	RNAz (Markham and Zuker, 2008)
Informação estrutural	Infernal (Nawrocki et al., 2009), RNAz

reais, também chamados **vetores de atributos** ou **vetores de características**. Cada atributo, por sua vez, pode se constituir de uma ou mais variáveis numéricas coletadas da sequência de entrada, por meio de rotinas, ferramentas e métodos diversos. Nesse trabalho, todos os atributos coletados são dados *ab-initio*, isto é, dados estimados utilizando somente informação da sequência fornecida. Essa propriedade permite a descoberta de membros de uma determinada classe sem realizar comparações com tipos conhecimentos, possibilitando a descoberta de agrupamentos inéditos, um comportamento desejável para o estudo de ncRNAs e com bons resultados publicados (Arrial et al., 2009). Os atributos escolhidos para o presente trabalho foram definidos por meio de extensa leitura da literatura. A Tabela 3.4 apresenta os nomes e quantidade de variáveis de cada atributo.

A escolha de parâmetros como tamanho de sequência ou tamanho de ORF baseia-se em observações experimentais e, principalmente, no poder de filtragem de tipos de ncRNAs (Liu et al., 2006). O alvo de grande parte dos classificadores e identificadores de ncRNA são longos ncRNAs, que tem uma longa cadeia de nucleotídeos, diferentemente de pequenos ncRNAs como os miRNAs ou snoRNAs. A filtragem de longos ncRNAs faz-se necessária devido ao objetivo de tais métodos. Além disso, lncRNAs estão envolvidos em muitas funções desconhecidas no organismo, além de não terem uma forma clara de identificação.

Os classificadores atuais baseados em aprendizado de máquina utilizam SVM para a identificação de ncRNAs (p. ex. Liu et al., 2006; Kong et al., 2007; Arrial et al., 2009). Resumidamente, definem como mRNA ou ncRNA uma dada sequência de entrada. Já o método SOM-Portrait é baseado num método não supervisionado que permite a adição de mais classes além de mRNA e ncRNA, possibilitando um maior detalhamento biológico da sequência, e maior precisão para a classificação em múltiplas classes. A utilização de SOM para esta classificação em múltiplas classes é ágil e escalável: pode-se adicionar mais classes sem exigir grande esforço computacional ou de codificação. Entretanto, adiciona-se a dificuldade na escolha do número de classes disponíveis para decisão da rede. Muitas classes podem causar um agrupamento das sequências que não retrata as características biológicas da sequência. Poucas classes, por sua vez, podem exigir a fusão de classes muito dissonantes, o que, no ponto de vista biológico, também não é desejável. Outro problema aparente da escolha empírica do número de classes de uma SOM é o momento de identificação das classes na rede recém treinada. Uma etapa supervisionada posterior de calibragem da rede é necessária para que o algoritmo SOM reconheça as classes a que cada sequência se referencia. Porém, esse conjunto de calibragem é extremamente difícil

de se construir, dado que alguns tipos de ncRNAs possuem poucas sequências catalogadas que poderiam ser utilizadas para representar essa classe (Backofen et al., 2007).

Capítulo 4

Metodologia

Nessa Seção, descreve-se a metodologia de trabalho e ferramentas auxiliares utilizadas durante o trabalho para realizar os objetivos propostos no Capítulo 1. Para efeito de organização do capítulo, cada método ou abordagem desenvolvida durante o projeto de mestrado será introduzida por sua motivação, sua elaboração e a descrição detalhada de seu funcionamento.

4.1 O método SOM-Portrait

O método SOM-Portrait foi inicialmente proposto em 2009, como projeto de conclusão de graduação (Silva, 2009). Sua motivação principal foi a exploração de padrões reconhecíveis, ou identificáveis, em agrupamentos (*clusters*) de RNA, com o objetivo de classificá-los em conjuntos não sobrepostos com propriedades composicionais e estruturais semelhantes.

O método utiliza procedimentos *ab initio* de avaliação de atributos para identificação de ncRNAs transcriptômicos, ou seja, identifica ncRNA dentro de um conjunto de RNAs que passaram pelo processo de transcrição descrito na Seção 2. O método baseia-se na coleta de atributos e características do RNA e submissão deles a uma rede SOM previamente treinada para classificação em múltiplas classes.

Apesar de alcançar resultados satisfatórios para uma rede não supervisionada, a ferramenta tem a escolha empírica do número de classes determinada pelo desempenho da rede utilizando um conjunto de testes com sequências codificantes e não codificantes de fungos. Tal abordagem, apesar de trazer um caso real de uso para o algoritmo, contribui pouco para a avaliação da generalização do treinamento da rede, objetivo principal de tal forma de teste (Bishop, 1995). A justificativa para essa afirmação reside na incerteza da anotação sobre a natureza dos transcritos dos organismos estudados. Vários deles, por exemplo, possuem pouca anotação relativa a ncRNAs, quando presentes.

Outros questionamentos foram levantados com relação aos atributos utilizados, principalmente com relação à grande quantidade de atributos numéricos. Finalmente, a implementação do método necessitava de revisão e reformulação para melhoria do desempenho do algoritmo, principalmente na demorada etapa de extração de atributos. Portanto, motiva-se a remanufatura do método SOM-Portrait com base nas seguintes adições e procedimentos:

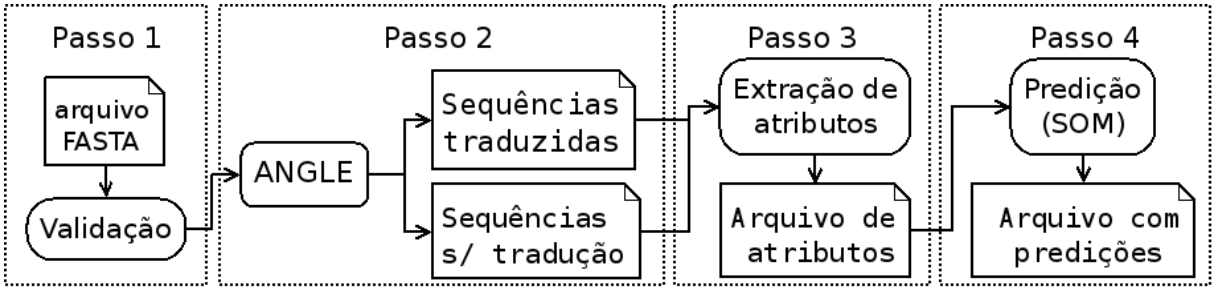


Figura 4.1: Fluxo de execução de tarefas do método SOM-Portrait.

- Encontro do melhor número de *clusters* baseado no desempenho do classificador, medido conforme descrito no Capítulo 3, com auxílio dos resultados obtidos com a classificação utilizando o algoritmo ART;
- Refinamento do espaço de decisão do classificador através de etapa supervisionada posterior, utilizando LVQ;
- Crítica ao número de atributos utilizados, sem inclusão de novos atributos, utilizando PCA;
- Recodificação do método para melhoria de desempenho.

4.1.1 Fluxo do programa

O programa foi inteiramente recodificado utilizando a linguagem Perl e a biblioteca BioPerl versão 1.6.901 (Stajich et al., 2002) seguindo o mesmo fluxo de tarefas da primeira versão proposta por Silva et al., 2009. A Figura 4.1 mostra o fluxo principal de tarefas do método SOM-Portrait.

É importante salientar que a divisão em etapas, como ilustrado na Figura 4.1, é simplesmente um recurso de organização do presente trabalho, e não a divisão modular ou descrição como diagrama de classes, que será descrito na Subseção 4.1.2. O treinamento da rede SOM é realizado por meio da execução dos Passos 1 a 3 do método sobre o conjunto de treinamento.

O algoritmo foi desenvolvido com foco em sequências de RNA transcriptômico, com tamanho maior do que $30nt$ e menor do que $65.535nt$, devido a restrições nos programas auxiliares ANGLE (Shimizu et al., 2006) e CAST (Promponas et al., 2000), respectivamente.

Pré-processamento

O Passo 1 refere-se à validação do arquivo de entrada fornecido pelo usuário, bem como dos parâmetros de configuração do método. O arquivo de entrada deve seguir o formato válido multiFASTA (for Biotechnology Information, 2011). A verificação do formato é feita pela biblioteca BioPerl. Até o momento, a etapa de pré-processamento e validação é bastante restrita. O método não verifica uso de nucleotídeos diferentes de A, C, T, G e U, somente excluindo sequências de RNA agindo de acordo com os seguintes critérios:

- Sequências com número de caracteres N acima de 20% do total de caracteres da sequência;
- Sequências que contêm número de caracteres menor do que $30nt$ ou maior do que $65.535nt$;
- Sequências com formato do cabeçalho FASTA incorreto;

A fase de pré-processamento também se encarrega de padronizar o arquivo de entrada, removendo caracteres especiais *CR*, *LF* e espaço.

Atualmente, o método SOM-Portrait suporta os seguintes parâmetros de entrada para o programa:

-i Arquivo de entrada (somente formato FASTA aceito);

-o Arquivo de saída (predições).

E os seguintes parâmetros de configuração do algoritmo:

- Parâmetros de configuração da rede SOM:
  - Diretório de execução da biblioteca SOM_PAK (Kohonen et al., 1996b);
  - Caminho para a rede SOM treinado;
  - Topologia (geometria) da rede SOM;
  - Número de classes no eixo "x";
  - Número de classes no eixo "y";
  - Tipo de função de vizinhança $hj(m)$;

- Parâmetros de configuração de outras bibliotecas e do ambiente de trabalho:
  - Diretório de execução do programa ANGLE (Shimizu et al., 2006);
  - Versão do programa ANGLE utilizada (32 bits ou 64 bits);
  - Diretório de execução do programa CAST (Promponas et al., 2000);
  - Versão do programa CAST utilizada (32 bits ou 64 bits);
  - Diretório temporário de trabalho

- Parâmetros de configuração do vetor de atributos:
  - Número de variáveis numéricas no vetor de atributos;
  - Formato do vetor de atributos (Formato aceito pela biblioteca SOM_PAK ou pela biblioteca LIBSVM (Chang and Lin, 2001));

- Miscelânea:
  - Quantidade de sequências por *thread* (somente para Passo 3 de extração de atributos).

Predição de ORFs

O Passo 2 refere-se à tradução das sequências de RNA utilizando o algoritmo ANGLE (Shimizu et al., 2006). O algoritmo combina informação composicional da sequência de RNA ou cDNA de entrada e informação estrutural da proteína utilizando métodos por AM aliados a HMM para calcular o potencial codificante dessa sequência. O método é proposto especialmente para pequenos trechos codificantes em sequências de cDNA de baixa qualidade, obtendo ótimos desempenhos para esse caso. Dada a dualidade mRNA-ncRNA explicada no Capítulo 2, associada à presença de pequenos trechos codificantes em sequências não codificantes, principalmente em lncRNAs (Dinger et al., 2008), a adoção dessa ferramenta torna-se muito interessante. Em correspondência pessoal com a autora do método, uma versão adaptada do algoritmo foi obtida, retornando também o escore associado à melhor ORF predita pelo método.

O algoritmo procura por ORFs nas seis possíveis fases de leitura, retornando as sequências de mRNA encontradas que codificam o peptídeo putativo, junto com a pontuação dada pelo programa. O algoritmo SOM-Portrait, de posse dessa informação, escolhe o mRNA putativo de maior pontuação e traduz o respectivo mRNA em sequência de peptídeos. Caso o ANGLE encontre uma possível ORF, uma correspondente estrutura de dados é criada com a informação do cabeçalho da sequência e sua melhor ORF predita, baseado no escore dado pelo algoritmo ANGLE. Caso ANGLE não encontre ORFs, a ausência de ORF predita é registrada na estrutura de dados com a marcação "none".

Extração de Atributos

O Passo 3 realiza a extração dos atributos numéricos das sequências de RNA e de suas respectivas ORFs traduzidas, quando presentes. Todos os valores de atributos numéricos são normalizados no intervalo $[0, 1]$. Tal procedimento evita distorções no espaço de decisão criado pelos algoritmos de aprendizado de máquina (Kohonen et al., 1996a). A Tabela 4.1 resume os parâmetros numéricos extraídos, seu significado biológico e a quantidade de variáveis numéricas que o compõem.

O método extrai os atributos numéricos das sequências e ORFs, gerando o vetor de características correspondente à sequência. Quando um atributo não pode ser retirado de uma sequência, uma marcação especial é feita de forma que o algoritmo de treinamento possa reconhecer essa ausência.

A frequência de nucleotídeos e aminoácidos, representada pelos atributos de Id 1 a 4, é extraída através da análise quantitativa da sequência de nucleotídeos e aminoácidos usando a Equação 4.1.

$$Freq(n, S) = \frac{ocorrencias(n, S)}{tamanho(S)} \tag{4.1}$$

Onde S é a sequência de RNA ou ORF predita e n é um nucleotídeo, dinucleotídeo ou trinucleotídeo válido. O cálculo de nucleotídeos ou aminoácidos desconhecidos ou faltantes (representados pelo caracter N e X) não é realizado, e sua contagem é subtraída do tamanho total de S. A função $ocorrencias(n, S)$ retorna o número de ocorrências da chave n em S utilizando uma janela de leitura de tamanho 1. O cálculo da frequência de aminoácidos é feito para os 20 tipos mais comuns de aminoácidos, conforme descritos no Capítulo 2, mais dois aminoácidos incorporados a partir da reescrita de códons de

Tabela 4.1: Atributos numéricos extraídos de cada sequência. A marcação ‡ denota atributos extraídos somente de ORFs preditas.

Id	Atributo	Detalhes	Números de variáveis
1	Frequência de nucleotídeos	4 nucleotídeos	4
2	Frequência dinucleotídeos	16 dinucleotídeos	16
3	Frequência trinucleotídeos	64 trinucleotídeos	64
4	Frequência aminoácidos ‡	22 aminoácidos	22
5	Tamanho da sequência (S)	$S \leq 100bp$; $100bp < S \leq 400bp$; $400bp < S \leq 900bp$; $S > 900bp$	4
6	Tamanho da ORF (L) ‡	$L \leq 20aa$; $20aa < L \leq 60aa$; $60aa < L \leq 100aa$; $L > 100aa$	4
7	Hidrofobicidade média ‡	Implementação do método Kyte-Doolittle (Kyte and Doolittle, 1982)	1
8	Ponto isoelétrico da proteína ‡	Ferramenta EMBOSS IEP (Rice et al., 2000)	1
9	Complexidade composicional da proteína ‡	Ferramenta CAST (Promponas et al., 2000)	1

parada, representados pelos caracteres U (Selenocisteína) e O (Pirolisina). Os caracteres B e Z e J, indicadores de tradução ambígua dos aminoácidos Asparagina ou Ácido Aspártico, Glutamina ou Ácido Glutâmico e Leucina ou Isoleucina, respectivamente, não são incluídos.

Os intervalos para o atributo 6 foi escolhido com base no intervalo ótimo evidenciado empiricamente em trabalhos prévios (Liu et al., 2006; Dinger et al., 2008) de tamanho de ORF $L = 100aa$. Os outros intervalos são criados como tentativas de pormenorizar ainda mais a participação de pequenos peptídeos em ncRNAs. O atributo retorna 4 variáveis, que assumem valor 1 ou 0, de acordo com a Equação 4.3.

$$var_k = 1 \text{ caso } tamanho(L) \in I \tag{4.2}$$

$$var_{i \neq k} = 0 \text{ caso contrário} \tag{4.3}$$

Onde i, k são índices $\in [1, 4]$ de atributos, $tamanho(L)$ retorna o tamanho da ORF L e I é o conjunto de intervalos para avaliação conforme descrito na Tabela 4.1. Para uma dada ORF L, o algoritmo SOM-Portrait ativa somente um dos intervalos, ajustando-o para 1. O atributo 5, tamanho S da sequência, é calculado de forma equivalente.

O cálculo da hidrofobicidade média da proteína (atributo 7) é feito por rotina interna à biblioteca BioPerl, utilizando um método para o cálculo da hidropatia de resíduos (Kyte

Tabela 4.2: Valores de hidrofobicidade para cada resíduo (Kyte and Doolittle, 1982).

Resíduo	Índice de hidropatia
Isoleucina	4,5
Valina	4,2
Leucina	3,8
Fenilalanina	2,8
Cisteína	2,5
Metionina	1,9
Alanina	1,8
Glicina	-0,4
Treonina	-0,7
Triptofano	-0,9
Serina	-0,8
Tirosina	-1,3
Prolina	-1,6
Histidina	-3,2
Ácido Glutâmico	-3,5
Glutamina	-3,5
Ácido Aspártico	-3,5
Asparagina	-3,5
Lisina	-3,9
Arginina	-4,5

and Doolittle, 1982). A rotina utiliza uma janela deslizante de leitura de tamanho três que se movimenta um caractere por iteração. Para cada trinca, consulta-se a Tabela 4.2 de hidropatia para resíduo.

Esse valor H, por sua vez, é somado para cada janela e normalizado através da Equação 4.4

$$HidroNorm = \frac{4.5 \times tamanho(L) + H}{9 \times tamanho(L)} \tag{4.4}$$

Onde $HidroNorm$ é o valor de H normalizado, hidrofobicidade média da proteína putativa L. É fácil observar pela Tabela 4.2 que $HidroNorm \in [0, 1]$.

O atributo 8, de predição do ponto isoelétrico da proteína, é extraído de uma ORF predita por uma rotina baseada no funcionamento do programa IEP da suíte EMBOSS (Rice et al., 2000). O programa calcula, para cada resíduo a partir do valor acumulado de seu pH em solução neutra, o valor de pK, conforme a Tabela 4.3.

O programa analisa o valor da constante de dissociação iônica pK para os resíduos com carga listados na Tabela 4.3 e retorna o valor final de pH no ponto isoelétrico do peptídeo. Finalmente, para efeitos de normalização do valor do atributo na faixa padrão $[0, 1]$, o pH é divido pelo máximo valor $pH = 14$. O programa descarta, para sua análise, os caracteres X de resíduos não identificados.

O cálculo do atributo 9, entropia composicional da proteína, é feito através do programa CAST (Promponas et al., 2000). O programa recebe a sequência de aminoácidos por entrada, encontrando e marcando regiões de baixa complexidade na estrutura da pro-

Tabela 4.3: Valores padrão de pK utilizados pelo programa IEP (Bleasby, 1999).

Resíduo	Valor de pK	Nome
N_term	8,6	N terminal
K	10,8	Lisina
R	12,5	Arginina
H	6,5	Histidina
D	3,9	Ácido Aspártico
E	4,1	Ácido Glutâmico
C	8,5	Cisteína
Y	10,1	Tirosina
C_term	3,6	C terminal

teína, por exemplo regiões com longas repetições de um certo tipo de aminoácido, de que decorrem estruturas pouco complexas mas que guardam algum grau de homologia com regiões funcionais de outras proteínas. A rotina é particularmente interessante para o problema de identificação de ncRNAs como forma de controle dos mRNA encontrados pela tradução do ANGLE. O algoritmo utiliza um procedimento de Smith-Waterman (Smith and Waterman, 1981) adaptado para encontrar as regiões de baixa complexidade composicional. O algoritmo marca tais as regiões de baixa complexidade com o caracter X. Por causa da ambiguidade entre esse caracter e a marcação de aminoácidos desconhecidos, também denotada pelo caracter X, o método SOM-PORTRAIT recupera a quantidade de X na sequência marcada pelo CAST e calcula sua frequência relativa na sequência original de aminoácidos, descontando os caracteres X relativos a aminoácidos desconhecidos já presentes na sequência.

Predição de ncRNAs

Ao final do Passo 3, o algoritmo produz uma estrutura de dados com os atributos numéricos de cada sequência fornecida pelo usuário. Como dito anteriormente, para realizar a PCA e o treinamento das redes SOM e ART e do método LVQ, o algoritmo é executado até o Passo 3 sobre os conjuntos de treinamento e validação. No caso do método SOM-Portrait, o Passo 4 consiste na execução de rotina simples para encontrar o neurônio mais próximo de um estímulo x_i retirado do conjunto de atributos numéricos criado no Passo 3. O procedimento utilizado pode ser sumarizado pela mesma Equação 3.17.

O resultado dado pelo algoritmo SOM-Portrait é ilustrado pela Tabela 4.4. O cabeçalho da biblioteca SOM_PAK contém informações necessárias para a execução de suas rotinas. Já o cabeçalho da predição é extraído da sequência de entrada. As coordenadas da classe definem o neurônio estimulado $y_c(m)$ na camada de saída, onde c é o índice do neurônio vencedor na camada de saída conforme formalismo proposto no Capítulo 3. O erro de quantização é calculado de forma semelhante ao procedimento *qerror*, explicado no Capítulo 3. Quanto maior esse valor, maior a probabilidade de erro de classificação. A predição p é dada de acordo com o nome da classe l associado a cada neurônio. Esse nome, para o presente trabalho, pode assumir valores genéricos, como "Classe1", "Classe2", etc, ou pode assumir valores "Coding" e "Noncoding", de acordo com o experimento realizado, a ser explanado na Seção 4.4.

Tabela 4.4: Formato de saída do arquivo de predição de ncRNAs do método SOM-Portrait.

Cabeçalho SOM_PAK *>cabeçalho: [coordenadas do neurônio] [erro de quantização] [predição p]*

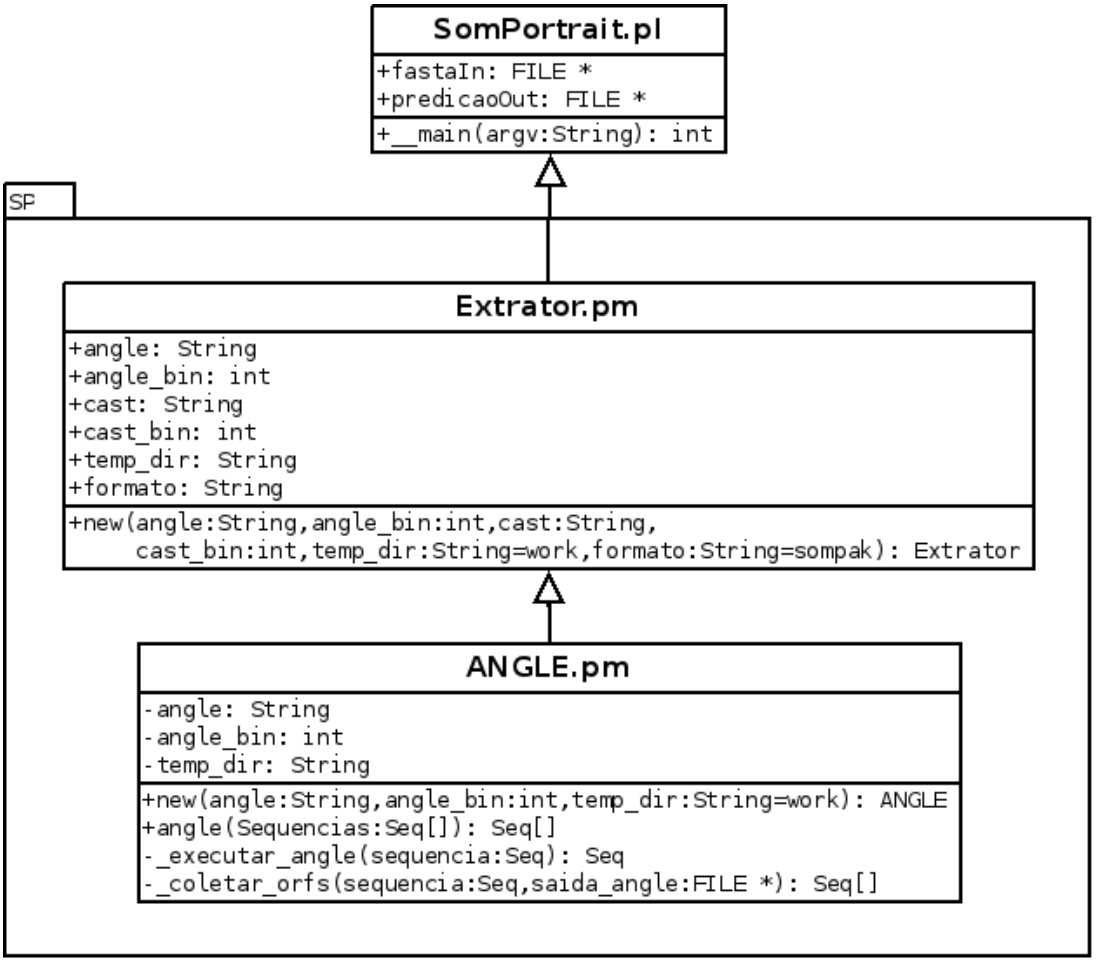


Figura 4.2: Diagrama com classes envolvidas no Passo 2.

4.1.2 Detalhamento do código

O método SOM-Portrait foi inteiramente desenvolvido utilizando a linguagem Perl, versão 5.14.2 e a biblioteca BioPerl versão 1.6.901. O paradigma de orientação a objetos foi preferido, por facilidade de leitura e manutenção de código. O programa foi modularizado em arquivos *.pm* (*Perl Module*), em formato de biblioteca própria. Os diagramas de classes 4.2 e 4.3 representam alguns dos relacionamento entre as diferentes classes do algoritmo SOM-Portrait.

O laço principal do algoritmo é detalhado no Algoritmo 4.4, junto com as principais estruturas de dados utilizadas pelo programa. Tais estruturas são:

- Conjunto de Sequências de Entrada: Vetor do tipo Seq (bio, 2012);
- Conjunto de ORFs traduzidas: Vetor do tipo Seq;
- Vetor de Atributos: Mapa *hash* de vetores de Números Reais;

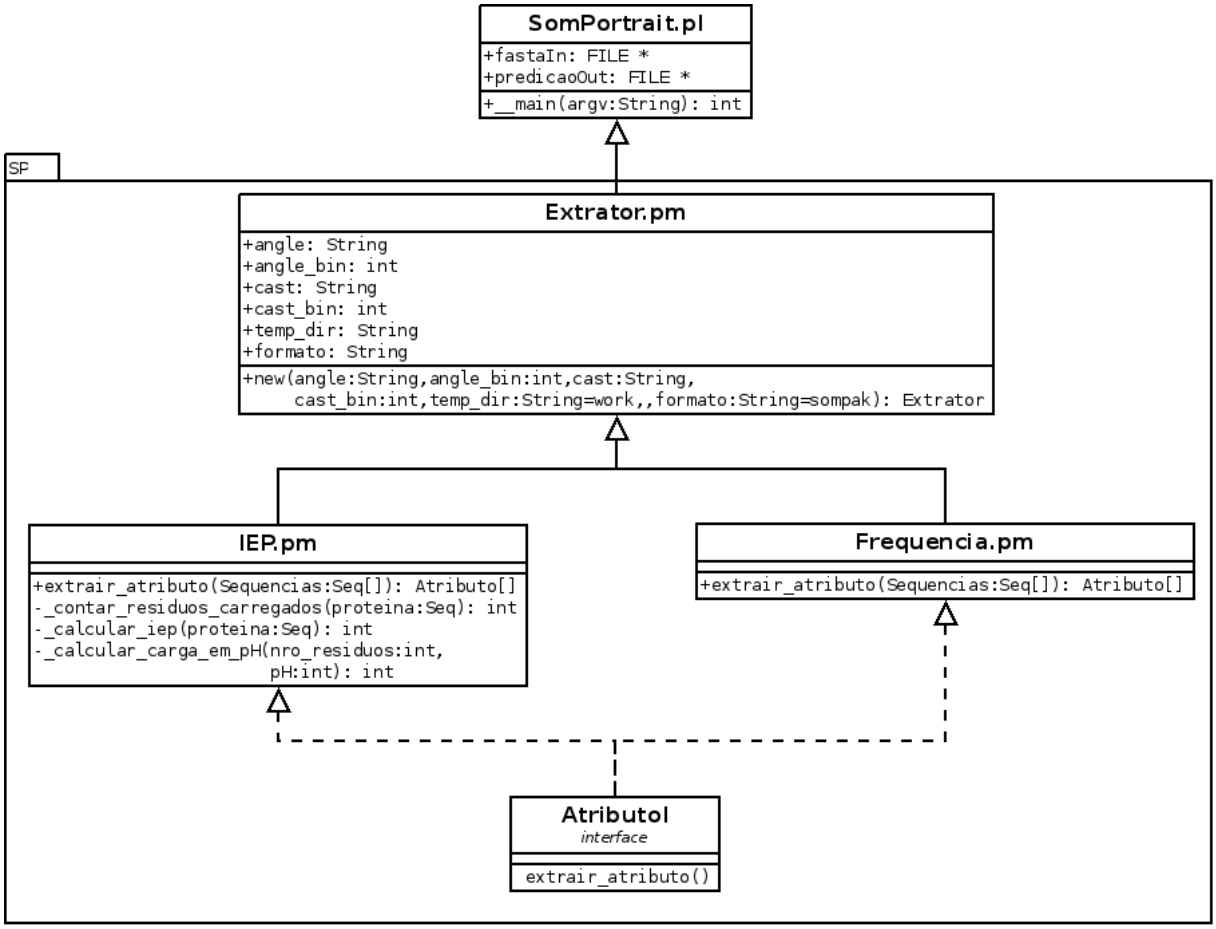


Figura 4.3: Diagrama com algumas das classes envolvidas no Passo 3.

Entrada: conjunto de entrada multiFASTA F
Saída : conjunto de predições P
Data: rede SOM treinada S

1 $P \leftarrow \emptyset$;
2 Inicialização do método, carregando bibliotecas externas e programas utilizados;
3 Validar(F);
4 **for** *cada sequência RNA* $f \in F$ **do**
5 | $ORF \leftarrow$ ANGLE(f);
6 | Vetor de atributos $\bar{v} \leftarrow$ ExtrairAtributos(f, ORF);
7 | Predição $p \leftarrow$ Predizer($\bar{v}$);
8 | $P \leftarrow P \cup \{p\}$;
9 **end**
10 **return** P;

Figura 4.4: Laço principal do programa SOM-PORTRAIT, com as principais estruturas de dados e programas auxiliares utilizados.

Tabela 4.5: Exemplificação do formato SOM_PAK para o vetor de atributos (Kohonen et al., 1996b).

[número de variáveis] [topologia] [dimensão x] [dimensão y] [vizinhança]			
0.1	...	0.3	classe 1
1	...	0	classe 2

A estrutura de dados correspondente ao vetor de atributos é implementada por meio de uma coleção de vetores de números reais, ordenados por *id* única definida na interface **AtributoI**. Cada classe de extração, assim como exemplificado na Figura 4.3, retorna um vetor de números reais relativo à função de extração utilizada. A convenção definida também na interface **AtributoI** determina que, caso o valor de uma das variáveis de um determinado atributo seja desconhecido, o caracter *UNDEF_ATR* deve ser utilizado. No caso do método SOM-Portrait, os primeiros experimentos foram realizados com *UNDEF_ATR* = "X", enquanto que, para os outros treinamentos da SOM, ART, LVQ e para o uso da PCA, *UNDEF_ATR* recebeu o valor 0 (zero). Experimentalmente, observou-se um desempenho melhor, em termos de acurácia, do classificador treinado com *UNDEF_ATR* = 0. O método também pode imprimir o vetor de atributos no formato especificado (SOM_PAK, LIBSVM ou CSV, atualmente). O formato SOM_PAK, utilizado para o treinamento dos algoritmos SOM e LVQ, é ilustrado na Tabela 4.5.

4.2 Dados de treinamento e validação

Para explanar as diversas atividades realizadas nessa Seção, faz-se necessário a divisão em dois momentos distintos. No primeiro momento, denotado de agora em diante "Primeiro Experimento", utilizou-se o mesmo conjunto de treinamento da ferramenta CONC (Liu et al., 2006), com a principal motivação de realizar o treinamento de um método não supervisionado com o mesmo conjunto de treinamento utilizado para um método supervisionado e avaliar seu desempenho por meio de um comparativo com outros métodos supervisionados. Os conjuntos perfazem 8.203 sequências de RNA, sendo 2.650 ncRNAs e 5.553 mRNAs. Como o conjunto inicial está desbalanceado, optou-se por selecionar aleatoriamente 2.000 sequências codificantes e 2.000 sequências não codificantes para construir um conjunto de treinamento balanceado. Algumas sequências codificantes foram descartadas no processo. As sequências de mRNA são oriundas do banco de dados GenBank (Benson et al., 2005), extraídas, por sua vez, através de seu identificador no banco de proteínas Swiss-Prot (Boeckmann et al., 2002). Já as sequências de ncRNA são oriundas dos bancos NONCODE (Liu et al., 2005) e RNAdb (Pang et al., 2005). No Primeiro Experimento, tal conjunto foi subdividido de acordo com a Tabela 4.6.

Foi dada preferência para conjuntos balanceados, com mesmo número de sequências codificantes e não codificantes. O conjunto *dbTr.dat* é utilizado unicamente para treinar o modelo SOM. O conjunto *dbCal.dat* é utilizado pela biblioteca SOM_PAK para nomear as classes, de acordo com a Equação 4.5.

$$label(y_c) = k \text{ se } \frac{\sum_{label(i)=k} \bar{x}_c(i)}{N} > 0,5 \tag{4.5}$$

Tabela 4.6: Nomes dos arquivos de treinamento utilizados no Primeiro Experimento, seus propósitos e a quantidade de sequências que os compõem.

Nome	**Função**	**Sequências**
dbTr.dat	sequências de treinamento	4.000
dbCal.dat	sequências para calibragem dos nós da rede treinada	800
dbVal.dat	sequências de validação do treinamento	500

Onde c é o índice do neurônio mais próximo do estímulo $\bar{x}$ recebido, calculado de acordo com a Equação 3.17, i é o índice de cada vetor $\bar{x}_c$ de entrada cujo estímulo é mais próximo de y_c e N o número total de vetores de atributo. A função $label(v)$ retorna a classe atribuída, no conjunto de calibragem, a um vetor de atributos v, de acordo com o formato descrito pela Tabela 4.5.

No segundo momento desse trabalho, de agora em diante chamado "Segundo Experimento", utilizou-se outro conjunto de dados de treinamento, motivado pelo tamanho limitado dos conjuntos do Primeiro Experimento. Seguindo a metodologia proposta por Arrial, 2008, 110.744 sequências codificantes (**Conjunto Negativo** ou **CN**) foram recuperadas do banco de dados EMBL (Cochrane et al., 2008), utilizando identificadores de proteínas do banco Swiss-Prot, e 360.864 sequências não codificantes (**Conjunto Positivo** ou **CP**) dos bancos NONCODE, RNAdb e Rfam (Gardner et al., 2009). A Figura 4.5 sumariza as atividades realizadas para criação do conjunto de sequências de treinamento e de validação.

Para a criação do CN, a versão 50.8 do banco de dados Swiss-Prot foi descarregada da página do projeto (swi, 2012). Cada sequência contém, em sua descrição, a referência ao identificador de sequências de cDNA no banco de dados EMBL de onde se originam. Eliminou-se redundâncias por sequências muito semelhantes utilizando o algoritmo CD-HIT (Li and Godzik, 2006). O algoritmo agrupa em *clusters* sequências com semelhança acima de um certo valor c fornecido, utilizando algoritmos de buscas textuais otimizados para padrões curtos. Nesse experimento, o valor $c = 0.9$ foi utilizado, o que representa a eliminação de sequências com mais de 90% de sua composição semelhante a alguma outra sequência do conjunto.

De posse das sequências não redundantes do banco Swiss-Prot, por meio de *script* fornecido pela página do projeto EMBL, as sequências de cDNA foram recuperadas. As sequências contêm dados genômicos, inclusive com genomas inteiros de procariotos. Tais dados foram removidos, por razão de sua extensão e complexidade desnecessárias para o treinamento do algoritmo. Um filtro para eliminar sequências com tamanho maior do que $65.535nt$ e menor do que $30nt$ também foi aplicado ao conjunto. Finalmente, para eliminar redundâncias no banco EMBL, o algoritmo CD-HIT foi novamente executado, com valor de corte $c = 0.9$.

Para a construção do CP, descarregou-se sequências dos bancos NONCODE (versão 3.0), RNAdb (versão 2.0) e Rfam (versão 10.0), utilizando o seguinte critério:

- **RNAdb:** *download* somente de sequências curadas pela literatura e preditas pelo software RNAz (Markham and Zuker, 2008);
- **NONCODE:** *download* de todas as sequências;

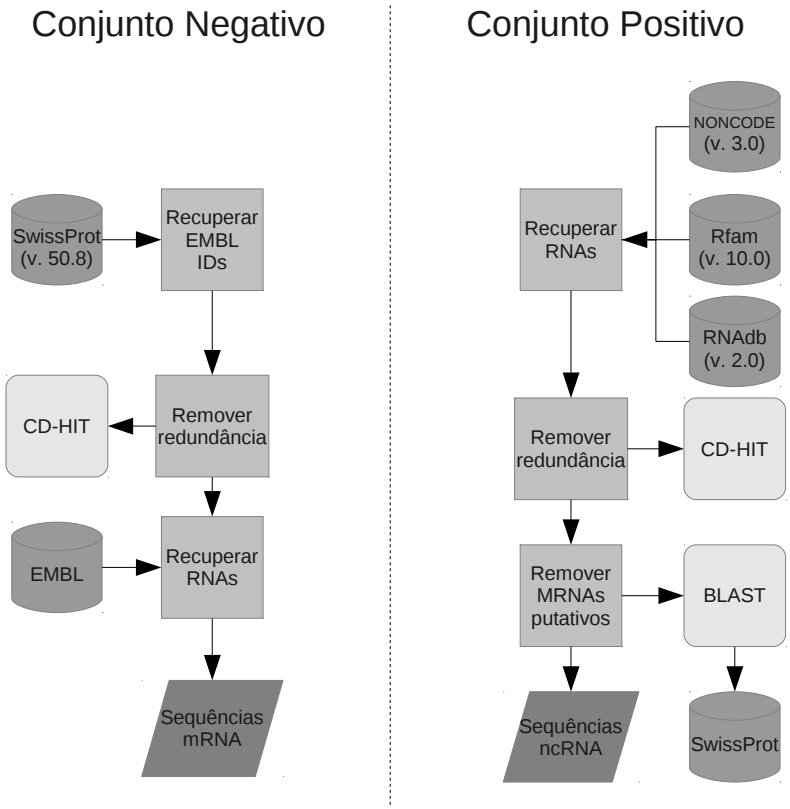


Figura 4.5: Diagrama exibindo a metodologia para criação dos conjuntos negativo e positivo de treinamento utilizados no Segundo Experimento, envolvendo os algoritmos de aprendizado de máquina.

- **Rfam:** *download* de sequências do conjunto de famílias-modelo.

O conjunto de famílias-modelo (*seed*) escolhido no banco Rfam é constituído somente por dados biológicos reais de sequências de ncRNAs e estruturas fundamentais, por exemplo, a estrutura da ferradura ou *hairpin* de miRNAs, assim como exemplificado na Figura 2.5. Dessa forma, assegura-se a qualidade das sequências de ncRNAs utilizadas, minimizando os erros de classificação por outros algoritmos. Também é importante frisar a preferência por algoritmos baseados em análise termodinâmica e estrutural, como Infernal (Nawrocki et al., 2009), donde o banco Rfam se baseia, e RNAz, que compõe parte do conjunto extraído do banco RNAdb. Já o banco NONCODE, por apresentar sequências de diferentes fontes, foi descarregado complemetamente, como forma de inserção de ncRNAs com outras características formadoras.

Um programa em C++ foi construído para recuperar famílias do Rfam com mais do que 20 membros por família. Esse passo foi incluído para eliminar a presença de várias famílias de pequenos ncRNAs muito específicos, fator de desequilíbrio no conjunto de treinamento final que poderia causar a especialização indesejada do algoritmo para pequenos ncRNAs dessas famílias. Muitas famílias de miRNAs, snRNAs e estruturas raras de RNA foram removidas dessa forma. Para os bancos NONCODE e RNAdb o programa CD-HIT com valor $c = 0.9$ foi usado para remover redundâncias. Para retirar sequências de ncRNAs muito similares a proteínas, o programa blastx da suíte de programas

Tabela 4.7: Número de sequências descarregadas dos bancos de dados Rfam (Gardner et al., 2009), NONCODE (Liu et al., 2005) e RNAdb (Pang et al., 2005) após filtragem utilizando os vários algoritmos e procedimentos descritos.

Bancos	Inicial	CD-HIT	Famílias	BLAST	Final
Rfam	715.846	–	28.800	–	28.800
NONCODE e RNAdb	955.694	357.942	–	332.064	332.064

Tabela 4.8: Nomes dos arquivos de treinamento utilizados no Segundo Experimento, seus propósitos e a quantidade de sequências que os compõem.

Nome	**Função**	**Sequências**
dbTr2.dat	sequências de treinamento	60.000
dbVal2.dat	sequências de validação do treinamento	50.000

BLAST (Altschul et al., 1997) foi utilizado. O valor de *e-value* utilizado para determinar um alinhamento satisfatório foi ajustado para 10^{-5}. Tal valor permite a presença de trechos codificantes esparsos em ncRNAs, evento mais provável do que longos trechos de material codificante em ncRNAs. A Tabela 4.7 lista a quantidade de sequências de cada banco de dados após a execução dos procedimentos descritos.

Os procedimentos do Segundo Experimento não utilizaram todas as 360.864 sequências obtidas para o CP. Deu-se preferência à utilização das sequências do banco Rfam, pelo rigor da anotação de seus dados, e pela presença de informação estrutural. A Tabela 4.8 sumariza a composição dos conjuntos de treinamento e validação utilizados no Segundo Experimento, juntando-se sequências escolhidas aleatoriamente de CN, as sequências do Rfam e outras sequências escolhidas também aleatoriamente de CP.

Utilizando o algoritmo SOM-Portrait até o Passo 3 de extração de atributos, recuperou-se os vetores de atributo do conjunto *dbTr.dat*. Uma análise estatística indicou que duas das variáveis do atributo 6 (tamanho L de ORF), $L \leq 20aa$ e $20aa < L \leq 60aa$, não foram utilizadas. Sua inclusão no conjunto de treinamento, entretanto, é irrelevante, pois todas as bibliotecas utilizadas ignoram variáveis nulas. Nenhuma outra anormalidade foi constatada. O resumo da análise estatística é dado no arquivo *estatisticaTr.ods*, em Materiais Complementares.

4.3 Dados de teste

O objetivo dos conjuntos de teste, nesse experimento, refletem o conceito explanado no Capítulo 3, com a motivação de avaliar o grau de generalização do treinamento realizado. No caso do **Primeiro Experimento**, dados reais de três diferentes fungos foram submetidos ao método SOM-Portrait e a outros três diferentes métodos, e seus resultados comparados. Já o **Segundo Experimento** utilizou dados de ncRNAs de 4 organismos bastante diferentes entre si, extraídos por outros métodos e ferramentas, em repositório bem curado, para avaliar diretamente a capacidade de generalização dos vários métodos treinados.

4.3.1 Organismos utilizadas no Primeiro Experimento

Os três fungos alvos do Primeiro Experimento desse trabalho foram escolhidos de acordo com critério de proximidade filogenética e de patogenicidade, ou seja, são fungos com genoma parecido, e principalmente, cujo funcionamento patogênico é parecido, com exceção do fungo *Aspergillus oryzae*, que não é patogênico.

O fungo *Aspergillus oryzae* (*Ao*) é um fungo filamentoso (Machida et al., 2005) com grande aplicação em indústrias alimentícias. O fungo é utilizado na fermentação de cereais para produção de álcool, açúcares e alimentos. É utilizado desde a Antiguidade com essa finalidade no leste asiático, a exemplo da levedura da cerveja (*Saccharomyces cerevisiae*). O isolado utilizado, $RIB40$, teve seus 9.051 *contigs* descarregados do projeto DOGAN, vinculado ao NITE (*National Institute of Technology and Evaluation*), no Japão (DOGAN, 2012).

O *Coccidioides immitis* (*Ci*) é um fungo dimórfico causador da Coccidioidomicose, comumente conhecida como a Febre do Vale (*Fever Valley*) ou também Reumatismo do Deserto (Valley-Fever.org, 2006). Sua distribuição geográfica abrange regiões áridas do oeste dos Estados Unidos, do México e do semiárido brasileiro, entre outras. Em seu hábitat natural é um fungo saprófita, em forma de hifas, mas quando inalado, muda para forma de levedura, causando desde reações alérgicas e pneumonia a dores e inflamações na pele e articulações. Estima-se que o número de infectados aumente a uma taxa elevada a cada ano. Os 9.757 *contigs* do isolado utilizado, *RS*, foram descarregados do Instituto Broad (of Harvard and MIT, 2012a).

O *Paracoccidioides brasiliensis* (*Pb*) é um parente próximo do *C. immitis*. Também é um fungo dimórfico, encontrado na forma de micélio ou esporos à temperatura de 24^oC a 26^oC e na forma de levedura à temperatura de 37^oC. O homem é um dos hospedeiros naturais do fungo, cuja distribuição geográfica abrange principalmente países da América Latina e Caribe, em regiões de florestas tropicais úmidas (Andrade, 2006). O *P. brasiliensis* é o agente patogênico da *Paracoccidiodomicose* (*PCM*), também chamada de *Blastomicose sul americana* ou *Blastomicose brasileira*. A PCM é uma micose sistêmica com expressivo número de infectados (10 milhões em toda a América Latina), sendo endêmica em regiões não contínuas da América Latina (México e América do Sul, exceto Guianas e Chile) (Andrade, 2006). Nesse trabalho, o isolado $Pb01$, de característica virulência, será utilizado. Os dados de 6.022 *contigs* sobre o *P. brasiliensis* foram obtidos por intermédio da equipe do Projeto do Genoma Funcional do *P. brasiliensis* (Felipe and Brígido, 2009), também vinculado ao Instituto Broad (of Harvard and MIT, 2012b).

O Projeto do Genoma Funcional do *P. brasiliensis* envolve diversos laboratórios da região central do Brasil com o objetivo de coletar informações sobre o transcriptoma do fungo referente à sua forma miceliana como em sua forma de levedura.

Outra contribuição do trabalho refere-se à análise de interseções entre as predições para os três fungos. Os procedimentos e resultados obtidos podem auxiliar na compreensão do funcionamento, a nível transcriptômico, dos mecanismos patogênicos e dimórficos do *P. brasiliensis*, fungo com comparativamente menos anotações, à luz de informações existentes no fungo *C. immitis*. Munindo-se de tais informações, novas drogas poderão ser desenvolvidas tendo por alvo os elementos descobertos por esse tipo de análise.

4.3.2 Organismos utilizados no Segundo Experimento

Para o segundo experimento, os seguintes ncRNAs de organismos foram escolhidos:

- 133 ncRNAs H/ACA e C/D Box e scaRNAs de corpos de Cajal de *Homo sapiens*;
- todos os 154 ncRNAs disponíveis de *Escherichia coli*;
- 413 ncRNAs não intrônicos de *Saccharomyces cerevisiae*;
- 421 snoRNAs de *Arabdopsis thaliana*.

As sequências foram descarregadas, respectivamente, do banco de dados snoRNA-Base (Lestrade and Weber, 2006), dos materiais complementares disponibilizados pelos autores de artigo sobre um novo método baseado em algoritmos genéticos para identificação de ncRNA (Sætrom et al., 2005), do banco de dados *Saccharomyces Genome Database* (SGD, 2011), e, finalmente, do banco de dados *Arabidopsis Small RNA Project* Gustafson et al. (2005) e de materiais complementares disponibilizados pelos autores de artigo sobre um método para predição de ncRNAs em *A. thaliana* (Song et al., 2009).

A escolha desses 4 organismos baseou-se na grande diferença de sua assinatura transcriptômica, em termos de evolução e complexidade. Foi dada preferência a famílias de snoRNAs, por causa de sua estrutura secundária bastante estudada e característica, conforme exemplo na Figura 4.6.

A bactéria *E. coli*, particularmente, foi alvo de extensas análises por outros trabalhos e pesquisa (p. ex. Eddy, 2001; Sætrom et al., 2005), resultando em interessantes observações, como a correlação entre pequenas ORFs putativas e ncRNAs que compõem a informação genética do organismo. Também é interessante notar que dados de ncRNAs em bactérias são comparativamente escassos; sua correta identificação, portanto, constitui um desafio de generalização do conhecimento aprendido pela rede.

4.4 Treinamento da rede SOM

Para o **Primeiro Experimento**, o ambiente de trabalho utilizado para treinamento foi formado por uma máquina com processador de dois núcleos Intel®Core™2 Duo (2,0Ghz), 2.024 Mb de RAM e Sistema Operacional Linux Ubuntu 8.04 (kernel 2.6.24-28-generic). A rede SOM utilizada pelo método SOM-Portrait foi treinada utilizando a biblioteca SOM_PAK (Kohonen et al., 1996b) versão 3.1.

Inicialmente, a rede SOM foi construída com os pesos de seus neurônios da camada de entrada atribuídos aleatoriamente, através do programa *randinit* e utilizando o conjunto *dbTr.dat*. O mapa foi configurado para uma rede de 2×2 nós em topologia hexagonal, o que possibilita ligações entre todos os nós, maximizando o potencial de classificação da rede. A escolha de 4 classes na camada de saída baseia-se nos resultados obtidos por Silva et al., 2009 ao treinar redes com 2 e 3 nós. A quarta classe permite treinar uma rede com dimensões mais favoráveis para o mapeamento das funções de densidade de probabilidade do conjunto de estímulos x_i de entrada na camada de saída (Kohonen et al., 1996b), possibilitando um estudo mais preciso da aplicação desse tipo de algoritmo ao problema de identificação e classificação de ncRNAs.

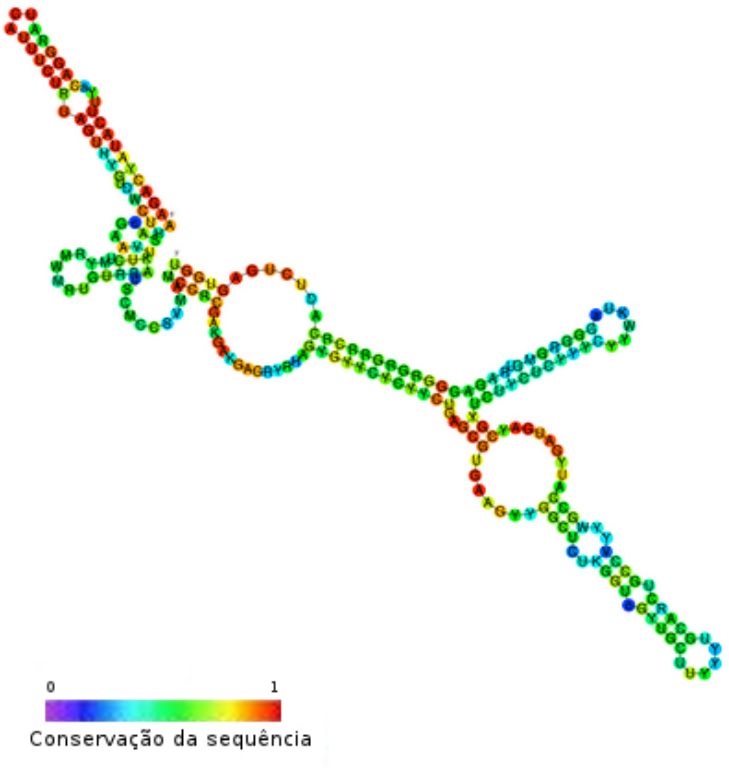


Figura 4.6: Exemplo de estrutura secundária de snoRNA U3 de fungo, mostrando também os índices de conservação dos pares de base da estrutura secundária, do mais conservado (cores frias) aos mais sujeitos a mutação (cores quentes) (Gruber et al., 2007; Gardner et al., 2009).

A função de vizinhança foi definida como gaussiana, e seu raio inicial $V_c(0) = 2$ na primeira etapa (ordenação) e $V_c(0) = 1$ na segunda etapa (convergência). Esses valores permitem, respectivamente, o ordenamento global dos estímulos de entrada e seu sucessivo refino utilizando inibições laterais regionais, potencializando o sinal de *clusters* próximos. A Figura 4.7 ilustra a representação gráfica da adoção desses valores de raio para a rede com topologia hexagonal 2×2 nós.

A notação para os neurônios, dada por coordenadas no plano xy, segue a convenção comumente adotada (Kohonen et al., 1996b; Sinha et al., 2010).

A adoção de função gaussiana, por sua vez, permite transições suaves entre diferentes graus de ativação dos neurônios da vizinhança V_c, com melhores resultados na etapa de convergência da rede. As taxas de aprendizadonas duas etapas foram ajustadas, respectivamente, para $\alpha(0) = 0,1$ e $\alpha(0) = 0,01$. A adoção de taxas de aprendizado menores permite maior exposição aos estímulos de entrada sem aumento do risco de super ajustamento por sua repetição. Assim, pode-se usar valores altos para as épocas de treinamento, reduzindo o impacto de cada época no aprendizado da rede. O mapa é então calibrado, ou seja, seus nós recebem a classes apropriada, utilizando o conjunto $dbCal.dat$, e finalmente o mapa é validado utilizando o conjunto $dbVal.dat$.

A validação do método foi feita utilizando-se um algoritmo simples para relacionar acurácia e número de épocas, com o objetivo de identificar regiões de estabilidade em que a rede treinada convergiu, de forma análoga a trabalho semelhante envolvendo treina-

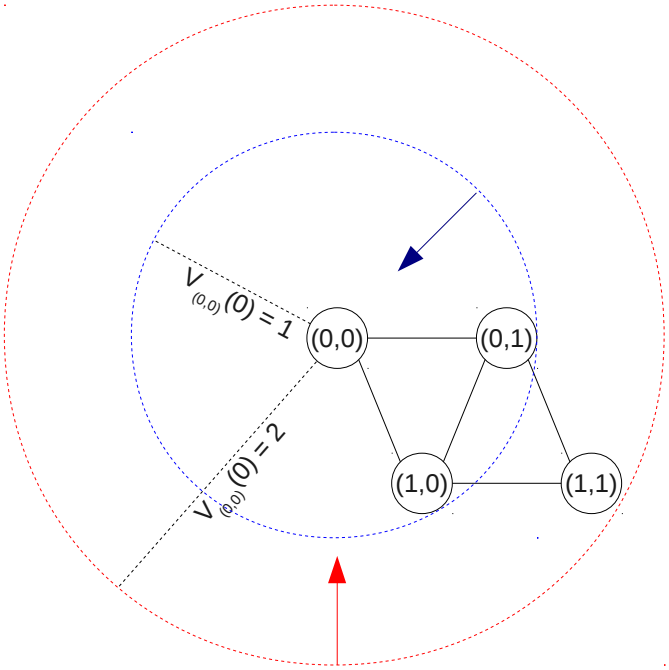


Figura 4.7: Raios de vizinhança V_c adotados. O raio inicial para a etapa de ordenação é indicado pela cor vermelha. O raio inicial para a etapa de convergência, menor, é indicado pela cor azul. As seta indicam o decréscimo do valor do raio em função do tempo.

Entrada: conjunto de treinamento T
Entrada: conjunto de calibragem C
Entrada: conjunto de validação V
Entrada: épocas M
Saída : conjunto de acurácias A
Data: rede SOM treinada S

Inicialização da rede SOM utilizando o procedimento *randinit*;
$A \leftarrow \emptyset$;
for *cada época* $m \in M$ **do**
    $mapa \leftarrow Treinar(T, m)$;
    $mapa \leftarrow Calibrar(C, mapa)$;
    $acc \leftarrow Validar(V, mapa)$;
    $A \leftarrow A \bigcup acc$;
end
return A;

Figura 4.8: Algoritmo de treinamento da rede SOM.

mento de redes não supervisionadas (Sinha et al., 2010). O funcionamento do algoritmo é detalhado na Figura 4.8.

Com a informação de épocas e do conjunto A de acurácias coletadas, o gráfico resultante é analisado e as regiões com melhor desempenho do classificador são escolhidas. Finalmente, o mapa SOM com a época e parâmetros de treinamento especificados através da rotina 4.8 é treinado.

A implementação das rotinas $Treinar$, $Calibrar$ e $Validar$ é detalhada pela chamada das respectivas rotinas da biblioteca SOM_PAK, a saber: *randinint* e *vsom*, *vcal* e *visual*. Tais rotinas são detalhadas por meio da descrição de seus parâmetros no Anexo I, Seção I.1.

De posse do modelo SOM treinado, é possível extrair o valor de *qerror* e também a u-matriz, ambos com relação ao conjunto de validação, para avaliar o grau de generalização da rede. Os resultados desse treinamento são exibidos e discutidos na Seção 5.1.

A análise da matriz de confusão foi refinada utilizando os seguintes procedimentos. As sequências correspondentes ao conjunto putativos mRNA fornecidos pela rede e de putativos ncRNA dos foram submetidas a uma busca por homologia no banco de dados de proteínas Swiss-Prot. O número de sequências de putativos mRNA restantes, depois dessa filtragem, representam mRNAs putativos que se parecem com mRNAs reais, presentes no banco de dados de proteínas. A contrapartida positiva, por sua vez, representa um provável falso positivo na identificação (PFP). Uma análise pormenorizada da anotação das sequências, presente no cabeçalho FASTA, retira do conjunto PFP os mRNAs hipotéticos ou putativos assim denominados segundo anotação de biólogos. O conjunto restante é o de falsos positivos obtidos para a rede. Finalmente, os conjuntos de putativos ncRNAs dos quatro métodos são comparados, mostrando a porcentagem de similaridade entre os métodos. Uma concordância relevante entre os métodos pode indicar sequências com sinais fortes nos organismos, construindo assim sequências-alvo de investigações mais detalhadas em busca de outros sinais característicos de ncRNA. É importante notar a metodologia utilizada nesse experimento, facilmente implementável como *pipeline* para anotação inicial automatizada de bons candidatos a ncRNAs.

Para o **Segundo Experimento**, o ambiente de trabalho utilizado foi formado por uma máquina com processador de quatro núcleos físicos Intel®Core$^{\text{TM}}i5-2410M$ $(2.30GHz)$, $4GB$ de RAM e Sistema Operacional Linux Fedora 16 (kernel $3.2.2-1.fc16.x86\_64$). As bibliotecas foram compiladas para arquitetura $64bits$, exceto o *software* CAST, que demandou suporte a bibliotecas de $32bits$. Mais detalhes sobre a construção do ambiente de trabalho estão no Anexo II. A rede SOM utilizada pelo método SOM-Portrait foi treinada utilizando a biblioteca SOM_PAK versão 3.1.

A rede SOM foi treinada de várias formas diferentes, cada uma com objetivo distinto, conforme enumeração descrita na Tabela 4.9. A enumeração respeita a ordem cronólogica de criação de cada rede, essencial para o entendimento da evolução dos experimentos desenvolvidos nesse trabalho.

O treinamento da rede 2×1 de ***Id*** 1 segue a metodologia de treinamento realizada no Primeiro Experimento. Inicialmente dá-se a chamada à rotina *randinit* utilizando o conjunto *dbTr2.dat*. A topologia usada, entretanto, é a retangular, adotada para maior controle da vizinhança adotada na etapa de convergência do algoritmo, principalmente na adoção de maior número de classes na camada de saída. A representação feita na Figura 4.9 demonstra a diferença na escolha da vizinhança feita pela adoção dessa topologia numa rede 2×2. Como primeira rede SOM treinada com o novo conjunto de treinamento, a escolha de 2 classes na camada de saída facilita a nomeação das classes encontradas e posterior avaliação de resultados.

Tabela 4.9: Diferentes redes SOM treinadas, de acordo com o número e disposição de nós na camada de saída, e objetivos a que se propõem.

Id	Topologia da Rede	Objetivo
1	Rede 2×1	Aplicada com conjunto $dbTr2.dat$ e 117 variáveis;
2	Rede 3×1	Aplicada com conjunto $dbTr2.dat$ e 117 variáveis;
3	Rede 3×2	Aplicada após análise do número de classes ideal utilizando rede ART
4	Rede 2×1	Aplicada com conjunto de atributos reduzido através da metodologia PCA;
5	Rede 2×2	Aplicada com conjunto de atributos reduzido através da metodologia PCA

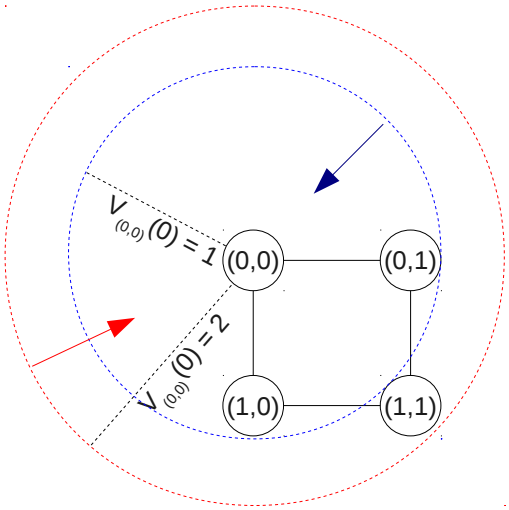


Figura 4.9: Topologia retangular e raios de vizinhança V_c adotados para a rede 2×2. O raio inicial para a etapa de ordenação é indicado pela cor vermelha. O raio inicial para a etapa de convergência, é indicado pela cor azul. As seta indicam o decréscimo do valor do raio em função do tempo.

Novamente, a função de vizinhança foi definida como gaussiana, e seu raio inicial $V_c(0) = 2$ na etapa de ordenação e $V_c(0) = 1$ na etapa de convergência.

As taxas de aprendizado nas duas etapas foram ajustadas, respectivamente, para $\alpha(0) = 0,5$ e $\alpha(0) = 0,05$. O aumento da taxa de aprendizado é justificado pelo aumento de estímulos diferentes utilizados. Proporcionalmente, aumentou-se o número de épocas fornecido para o programa de avaliação de acurácia utilizado no Primeiro Experimento. Esse programa foi também utilizado para treinamento do mapa SOM no Segundo Experimento. Para tal, 1.000 sequências do arquivo $dbVal2.dat$ foram utilizadas para

construir um arquivo de calibragem $dbCal2.dat$. Esse arquivo é utilizado somente para essa avaliação de acurácia do treinamento.

Para a rede treinada após o procedimento acima, a matriz de confusão utilizando o conjunto $dbVal2.dat$ foi construída e analisada, com as taxas e medidas de desempenho da rede descritas no Capítulo 3 calculadas, bem como os valores de $qerror$ e a gráfico representando a u-matriz do espaço de decisão treinado. Os resultados dos diversos treinamentos de redes SOM utilizando somente o novo conjunto de treinamento são expostos e discutidos na Seção 5.2.

A rede de ***Id*** 2 usa topologia retangular, função de vizinhança gaussiana, valores de $\alpha(0)$ idênticos a ***Id*** 1, e raio inicial de vizinhança nas etapas de ordenação e convergência de, respectivamente, $V_c(0) = 3$ e $V_c(0) = 1$. Em especial, essa rede representa o valor máximo de classes que os procedimentos empíricos conseguiram alcançar. Redes com mais classes apresentaram desempenho inferior a essa rede (dados não exibidos). Após a adoção do procedimento ART, porém, outras classes puderam ser adicionadas de forma eficaz. Especialmente para essa rede, a etapa de avaliação inclui a avaliação do desempenho da ferramenta BLAST (Altschul et al., 1997) aplicada às sequências do conjunto $dbVal2.dat$ agrupadas conforme os resultados da matriz de confusão. Tal análise objetiva analisar a capacidade do classificador em comparação ao método BLAST, principalmente como forma de encontrar bons candidatos a mRNA no conjunto de falsos positivos, comportamento indesejado, que demonstra baixa especificidade da rede.

A rede de ***Id*** 3 foi treinada de forma idêntica à rede ***Id*** 2, e a rede de ***Id*** 4 foi treinada de forma idêntica à rede ***Id*** 1.

A rede de ***Id*** 5 foi criada usando topologia retangular e função de vizinhança gaussiana. Na etapa de ordenação foi atribuído $\alpha(0) = 0,5$ e $V_c(0) = 2$. A etapa de convergência utilizou $\alpha(0) = 0,05$ e $V_c(0) = 1$.

4.5 O método ART-Portrait

O método ART-Portrait é proposto como adaptação dos Passos 1, 3 e 4 do algoritmo descrito na Seção 4.1. Especificamente, o vetor de características criado na etapa de extração de atributos segue o formato de valores separados por vírgula (*comma separated values* ou CSV), e as rotinas de treinamento e execução utilizam a biblioteca ART-distance (Hudik and Zizka, 2011). No Passo 1, alterações foram feitas nos parâmetros de configuração do algoritmo:

- Parâmetros de configuração da rede ART:
  - Diretório de execução da biblioteca ART-distance;
  - Caminho para a rede ART treinado;
  - Topologia (geometria) da rede ART;
  - Número de classes no eixo “x”;
  - Número de classes no eixo “y”;

- Parâmetros de configuração de outras bibliotecas e do ambiente de trabalho:
  - Diretório de execução do programa ANGLE (Shimizu et al., 2006);

  - Versão do programa ANGLE utilizada (32 bits ou 64 bits);
  - Diretório de execução do programa CAST (Promponas et al., 2000);
  - Versão do programa CAST utilizada (32 bits ou 64 bits);
  - Diretório temporário de trabalho

- Parâmetros de configuração do vetor de atributos:
  - Número de variáveis numéricas no vetor de atributos;
  - Formato do vetor de atributos (Formato CSV);

- Miscelânea:
  - Quantidade de sequências por *thread* (somente para Passo 3 de extração de atributos).

A rede ART não utiliza conexões laterais entre os neurônios, que caracterizariam a necessidade de utilização de topologia e disposição de neurônios através dos parâmetros x e y (Frank et al., 1998; Kasabov, 1998; Hudik and Zizka, 2011). Entretanto, a rotina *visual* da biblioteca SOM_PAK é utilizada para escolher o neurônio mais próximo y_c do estímulo x_i de entrada. Tal procedimento pode ser utilizado sem impedimento, assumindo raio de vizinhança $V_c = 1$.

A biblioteca ART-distance é uma implementação do algoritmo explanado na Seção 3.4. Os parâmetros utilizados para treinamento são descritos no Anexo I, Seção I.2. É importante frisar o funcionamento do parâmetro α fornecido à rede para treinamento. Sua implementação é descrita por Frank et al., 1998 como uma etapa de escolha do protótipo mais similar ao estímulo x_i. O valor $\alpha \in [0, \frac{1}{\sqrt{m}}]$, onde m é o número de atributos numéricos do vetor de características, determina a quantidade de protótipos que serão avaliados, de acordo com a Equação 4.7.

$$t_j = i_p \cdot w_{pj} \text{ , se } w_{jp} \text{ tiver sido visitado} \tag{4.6}$$

$$= \alpha \cdot \sum_{i=1}^{m} i_i \text{ , caso contrário} \tag{4.7}$$

O limite superior é decorrência do cálculo da atividade t_j da rede, conforme algoritmo exibido na Figura 3.10. Em suma, o valor de α determina a profundidade da busca por protótipos existentes. O valor $\alpha = 0$ determina que o algoritmo buscará o protótipo de estímulo mais próximo a x_i em todos os protótipos visitados antes de utilizar um protótipo não visitado.

A validação do treinamento da rede ART é feita avaliando o critério de parada implementado pela biblioteca ART-distance, valor de flutuação e dos estímulos de entrada na rede. O critério baseia-se no percentual de estímulos que trocam de *cluster* entre duas épocas consecutivas de treinamento. Um valor alto de e representa sinais muito confusos ou pouca convergência dos valores dos protótipos. Um valor máximo k de flutuação é fornecido à rede, e caso $e \leq k$, o algoritmo considera a rede consolidada e termina o

treinamento. Uma época máxima n também é fornecida ao algoritmo como critério de parada. Ao alcançar esse critério, o algoritmo retorna a melhor configuração dos protótipos encontrada em relação a e. Fixando todos os outros parâmetros, redes ART foram treinadas iterando o valor de vigilância ρ, dentro de uma faixa de valores estimada como ótima. Para cada rede treinada, o gráfico relacionando ρ adotado e e encontrado é apresentado como resultado na Subseção 5.2.4.

A rede ART foi treinada fixando os valores de treinamento $\eta = 0,25$, $\alpha = 0,01$, número máximo de épocas $n = 40$ e $e = 0,05$. O valor de número máximo de épocas é reduzido para o algoritmo porque, diferentemente da biblioteca SOM_PAK, cada época do algoritmo ART-distance equivale à apresentação de todos os estímulos à rede. Portanto, a quantidade máxima de épocas reflete a quantidade máxima de vezes que o estímulo é apresentado. O valor de α é menor do que o valor padrão dado pela biblioteca. A adoção desse valor permite uma pesquisa mais ampla por protótipos existentes compatíveis com o estímulo de entrada. Como efeito observável, a criação de classes com poucos estímulos associados é menos provável, dependendo quase exclusivamente do valor de ρ adotado. Isso permite realizar o procedimento de validação descrito anteriormente. O valor da taxa de aprendizado η foi estimado acima do valor correspondente utilizado na rede SOM dado o baixo número de épocas necessário, em alguns casos de treinamento, para o algoritmo convergir.

A análise da matriz de confusão permite avaliar a distribuição de sequências codificantes e não codificantes por classe e nomear, de acordo com a predominância de uma ou outra, a classe como "Codificante" ou "Não codificante". A partir dessa nomeação, as medidas de desempenho descritas no Capítulo 3 são calculadas e discutidas na Seção 5.2.6. A medida de *qerror* também é reportada para cada rede ART treinada.

4.6 Análise de Componentes Principais

A motivação principal para a análise de componentes principais é a de reduzir o grande número de variáveis numéricas extraídas no Passo 3 do algoritmo SOM-Portrait. Essa redução implica, além da redução do tempo necessário para executar as operações de treinamento no algoritmo, a redução da complexidade dimensional do conjunto de treinamento. Bem realizada, essa redução pode suprimir sinais muito fracos, muito difusos ou pouco relacionados aos sinais mais intensos. A escolha dos atributos, explanada na Subseção 4.1.1 claramente contém redundâncias desnecessárias á rede. Como exemplo, a ocorrência do trinucleotídeo AAA incorre na excitação proporcional dos parâmetros de frequência do dinucleotídeo AA e do nucleotídeo A. Essa forte correlação, nesse caso, não proporciona contribuição relevante ao problema de classificação de ncRNAs.

Avaliar o grau de correlação entre duas variáveis não é tarefa trivial. Para alcançar esse objetivo, a biblioteca FactoMineR (Lê et al., 2008) foi utilizada. A biblioteca permite uma vasta gama de análises dimensionais e de agrupamento de dados numéricos em *clusters*. Análise de componentes principais, análises de agrupamentos hierárquicos e análises estatísticas diversas são algumas das funções disponíveis. Nesse trabalho, somente a funcionalidade de execução de PCA foi utilizada.

O conjunto de dados de treinamento $dbTr2.dat$ foi utilizado para a PCA. O *script* R com todas as operações realizadas é disponibilizado como Material Complementar. Os comandos utilizados são exibidos no Anexo I, Seção I.4.

Para selecionar os melhores atributos, utilizam-se duas abordagens: a primeira, baseada somente na informação de autovalores relativos às 6 variáveis com maior contribuição de variância no conjunto, e a segunda, utilizando também a matriz de correlação R entre as 117 variáveis. Foram escolhidas 6 variáveis baseado na contribuição cumulativa da variância dessas variáveis, totalizando $\approx 49\%$, um valor considerado satisfatório para o experimento. Também analisa-se a representação gráfica para as 6 melhores variáveis, construída nos moldes do exemplo dado na Figura 3.3. Para a construção desses gráficos, utilizou-se um critério de seleção de variáveis baseado na contribuição da projeção dessas variáveis no plano, dada pelo valor $\cos^2$ da projeção $\pi(q_a, q_b)$ explicada na Subseção 3.1.4.

Outro gráfico apresentado como resultado da PCA relaciona os exemplares do conjunto *dbTr*2.*dat* em relação às duas variáveis de maior contribuição de variância utilizadas como eixos x e y da análise. O resultado exibe o grau de separação dos exemplares codificantes e não codificantes, o que auxilia a observação do espaço de decisão do conjunto antes da aplicação de algoritmos de agrupamento.

Os resultados da PCA são aplicados às redes ART e SOM, treinando os mapas já mencionados nas Seções 4.4 e 4.5, e discutidos na Seção 5.2.7.

4.7 Etapa supervisionada utilizando LVQ

O propósito da implantação da rotina supervisionada baseada no algoritmo de *Learning Vector Quantization* tem raiz na necessidade de refino do espaço de decisão construído pelas redes ART e SOM, propósito ideal do algoritmo (Kohonen et al., 1996a; Haykin, 1999), conforme descrito na Seção 3.3.

A aplicação do algoritmo baseia-se na adaptação dos vetores de peso $\bar{w}_j$ de uma rede ART ou SOM previamente treinadas e nomeadas. Essa adaptação é feita com conjunto de treinamento construído de forma a reforçar os pesos com os estímulos de entrada cujo nome de classe é equivalente ao nome de classe do neurônio vencedor y_c. Sendo assim, faz-se necessária a adaptação de um conjunto de treinamento a partir dos dados de treinamento descritos na Seção 4.2. Obviamente, a aplicação do próprio conjunto *dbTr*2.*dat* novamente à rede contribuirá pouco. Portanto, optou-se por adaptar as 50.000 sequências do conjunto *dbVal*2.*dat* para esse fim. Os exemplares oriundos de **CP** foram nomeados "Noncoding", e os vindos de **CN**, "Coding", de acordo com o formato SOM_PAK exemplificado pela Tabela 4.5. Esse conjunto foi nomeado *dbLvqOpt*1.*dat*. Outro conjunto de treinamento foi proposto, esse mais direcionado, contendo somente sequências da família Rfam não utilizadas no treinamento, e, em número igual, sequências codificantes para o controle negativo. Esse conjunto totaliza 7.200 sequências. O conjunto foi nomeado *dbLvqOpt*2.*dat*.

A biblioteca LVQ_PAK versão 3.1 (Kohonen et al., 1996a) foi utilizada para os procedimentos de treinamento. Para o treinamento, o algoritmo OLVQ1 foi escolhido, por sua rápida convergência, simplicidade de código e suporte aos dados de treinamento das redes ART e SOM utilizadas. O método otimizado utiliza procedimento semelhante ao descrito na Seção 3.3, com parâmetros e configuração descritos no Anexo I, Seção I.3. O método recebe como principais entradas o conjunto de treinamento e a rede treinada a ser otimizada no formato SOM_PAK.

O treinamento do algoritmo com o conjunto *dbLvqOpt*1.*dat* foi realizado com taxa de aprendizado $\alpha_c(0) = 0,1$ com decréscimo linear em relação á época, que representa

um bom compromisso entre número de épocas e quantidade de informação no conjunto *dbLvqOpt*1.*dat*. Um valor de época fixa 2.500.000 foi utilizado, baseado num valor empírico do número de iterações desejadas para o algoritmo. Para o conjunto *dbLvqOpt*2.*dat*, o valor de épocas foi reduzido para 360.000, refletindo o tamanho menor do conjunto. O valor de $\alpha_c(0)$, porém, foi mantido em $0,1$.

O conjunto de validação $dbVal.dat$ não pode ser utilizado, nesse caso, para avaliação do treinamento LVQ. Para esse fim, o trabalho baseia-se somente na interpretação dos dados de teste descritos na Seção 4.3.2. Para os redes SOM refinadas com essa etapa de treinamento supervisionado, calcula-se também o valor de $qerror$, utilizando o próprio conjunto $dbVal.dat$, como forma de eliminar erros grosseiros no treinamento, e também apresenta-se a representação por u-matriz da rede refinada. A etapa supervisionada utilizando os dois conjuntos diferentes de treinamento foi aplicada às redes antes e depois da redução dimensional proporcionada pelos resultados da PCA, todos os resultados descritos em respectivas subseções, para cada experimento não supervisionado discutido no Capítulo 5.

Capítulo 5

Resultados

Nesse capítulo, exibe-se e discute-se os resultados de todos os treinamentos e procedimentos explanados no Capítulo 4. Para efeitos de organização, divide-se, como no Capítulo 4, os resultados em dois blocos, chamados doravante "Primeiro Experimento", explanado na Seção 5.1 e relativo ao treinamento SOM-Portrait com o conjunto descrito na Seção 4.2, e "Segundo Experimento", detalhado na Seção 5.2 e relativo à Análise de Componente Principal e treinamento dos algoritmos SOM, ART e LVQ.

5.1 Primeiro Experimento

Apresenta-se os resultados do treinamento para o Primeiro Experimento, a validação da rede e estudo de caso, ou etapa de teste da rede, com objetivo de avaliar o grau de generalização do treinamento utilizando dados biológicos reais. O Primeiro Experimento refere-se à etapa inicial de estudo da aplicação de redes não supervisionadas a um conjunto de treinamento utilizado pelo identificador CONC (Liu et al., 2006), usando, como medida de desempenho da rede, a comparação com outras ferramentas de propósito semelhante.

5.1.1 Treinamento da rede SOM

Conforme descrito na Seção 4.2, um programa foi desenvolvido para relacionar os valores de acurácia do método aplicado ao conjunto *dbVal.dat* de validação com o número de épocas m do algoritmo, com o objetivo de encontrar um ponto satisfatório de convergência do algoritmo. Para o ambiente de execução do Primeiro Experimento descrito no Capítulo 4, cada etapa de treinamento e validação do modelo consumiu de aproximadamente $\approx 10s$, iteradas 500 vezes, totalizando $\approx 1,2h$ de execução. O gráfico na Figura 5.1 exibe os resultados do treinamento da rede SOM 2×2 utilizando os dados de treinamento e validação do Primeiro Experimento.

A linha azul no gráfico significa o valor da acurácia. Os saltos no gráfico evidenciam regiões em que a rede não consegue convergir eficientemente. Uma boa região nesse gráfico, portanto, é uma região de alta acurácia que esteja distante de saltos. É importante notar que, apesar do algoritmo apresentar uma aparente convergência já nas primeiras épocas de treinamento, ainda há relevante componente aleatório, resultante da adoção de pesos aleatórios pela rotina *randinit*. O contrário também é verdade: à medida em que acumulam-se épocas na etapa de convergência da rede, os sinais relevantes para a

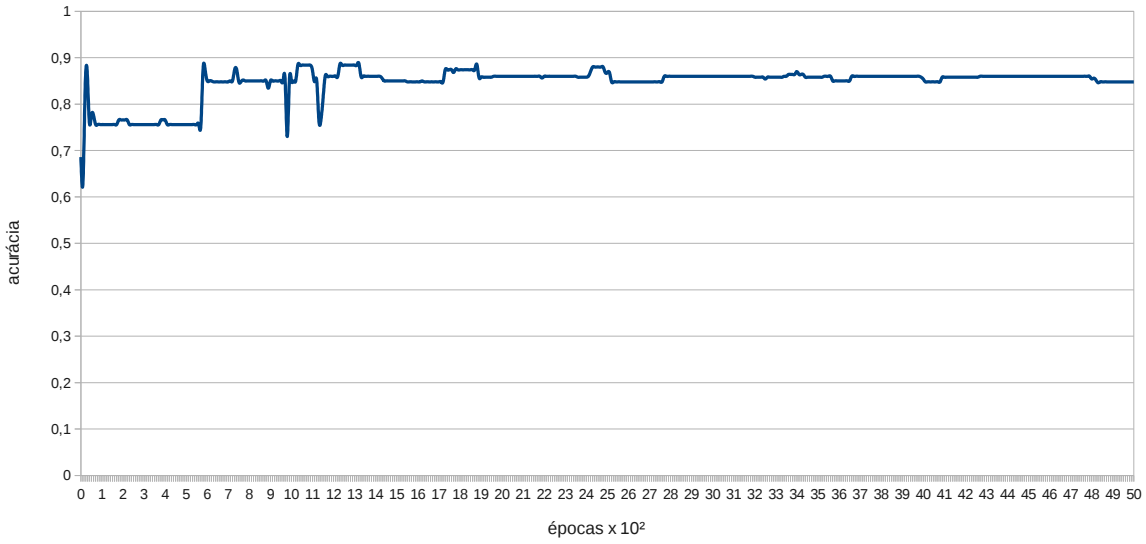


Figura 5.1: Gráfico de treinamento da rede SOM 2×2 treinada com o conjunto *dbTr.dat* no Primeiro Experimento.

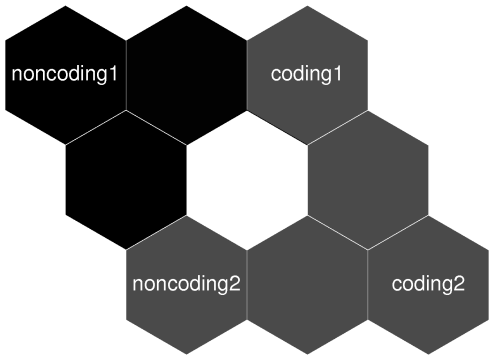


Figura 5.2: U-matriz para a rede SOM treinada no método SOM-Portrait.

classificação são continuamente estimulados. Esse estímulo contínuo pode cessar, situação em que a rede converge e o treinamento estabiliza os pesos nos vetores de referência. Entretanto, também é possível que um estímulo contínuo leve a um super ajustamento da rede para o conjunto de treinamento *dbTr.dat*. O valor ótimo de épocas encontrado para o treinamento da rede é de 2.340 para a etapa de ordenação e 23.400 para a etapa de convergência, o que resultou num mapa com acurácia estimada em 88%.

A partir da calibragem do nós do mapa SOM, feita através do procedimento *vcal* da biblioteca SOM_PAK (Kohonen et al., 1996b), foi possível determinar o espaço de decisão criado pelo algoritmo após o treinamento. A sua divisão das classes e a topologia do mapa podem ser observados pelo diagrama da Figura 5.2, gerado pela rotina *umatrix* da biblioteca.

A Figura 5.2 revela que uma das classes de ncRNAs está mais próxima das outras duas classes de mRNAs do que da outra classe de ncRNAs. Essa classe afastada provavelmente

Tabela 5.1: Matriz de confusão para a rede SOM 2×2 treinada no Primeiro Experimento.

		Predito			
		Classe 1	Classe 2	Classe 3	Classe 4
Real	Codificante	35	63	2	150
	Não Codificante	149	19	67	15

Tabela 5.2: Medidas de performance P para a rede SOM 2×2 do Primeiro Experimento.

Precisão	*Recall*	Especificidade
$0,864$	$0,854$	$0,862$
Acurácia	**Medida F**	**MCC**
$0,858$	$0,859$	$0,620$

é gerada por um sinal bem caracterizado, enquanto que as classes mais miscigenadas podem indicar uma interação entre classes codantes e classes não codantes, talvez exercida por conjuntos de ncRNAs com trechos codantes. Mais provavelmente, porém, tal propriedade é causada por uma insuficiência de dados de treinamento para uma caracterização fidedigna da variedade de agrupamentos possíveis para o problema, dado o tamanho reduzido de $dbTr.dat$.

A matriz de confusão para a melhor rede treinada foi construída para avaliar precisamente o treinamento, conforme as medidas de performance da rede apresentadas anteriormente. A Tabela 5.1 sumariza os resultados obtidos.

Os resultados utilizam as 500 sequências do conjunto de dados de validação $dbVal.dat$. As informações foram utilizadas para nomear as Classes 1, 2, 3 e 4, respectivamente, como "Noncoding1", "Coding1", "Noncoding2" e "Coding2". A utilização reflete somente a disposição topológica dos neurônios, conforme descrição na Figura 4.9. A caracterização de classe codificante ou não codificante é conceituada pela quantidade de exemplares que a perfaz. A partir dos dados coletados, constrói-se a Tabela 5.2 com os valores de performance obtidos pela rede, conforme medidas descritas no Capítulo 3.

Os resultados demonstram acurácia próxima da estimada pela etapa de treinamento. A especificidade, relacionada com a capacidade inerente à rede de reconhecer o conjunto negativo de exemplares, orbitou em torno de um número similar à precisão. Isso indica um equilíbrio no sinais codantes e não codantes, refletido na medida F como um valor bem próximo à acurácia. Em comparação a outros métodos identificadores, porém, o método é inferior em termos de acurácia. Comparando-se diretamente com o método CONC, de onde as sequências de treinamento se originam, a sensibilidade e especificidade obtidas através de método de validação cruzada com 10 partições foram de $0,98$ e $0,97$, respectivamente.

O valor da medida F obtido reforça o entendimento dado pelos valores de precisão e *recall*. O valor de *recall* menor, entretanto, é sinal de alerta, podendo significar uma taxa elevada de falsos positivos. Os experimentos utilizando dados de fungos deverão solucionar esse questionamento com um cálculo detalhado da taxa de falsos positivos. Já o valor do coeficiente de correlação de Matthews retorna valores na faixa $[-1, 1]$, como descrito no Capítulo 3. O valor obtido demonstra, portanto, que o classificador atua de

forma razoável, com evidente vantagem na identificação do conjunto positivo dos dados de validação.

É importante frisar também que, para os objetivos desse Primeiro Experimento, expostos no Capítulo 4, os valores encontrados demonstraram a capacidade da rede não supervisionada identificar corretamente os sinais codantes e não codantes dentro de um conjunto de treinamento não especificamente criado tendo em mente a utilização desse tipo de algoritmo de AM.

Outro ponto importante a salientar diz respeito à capacidade de classificação dos dados em múltiplas classes. Tendo por referência algoritmos supervisionados baseados em SVM, o problema de divisão em múltiplas classes é normalmente implementado por uma abordagem *tudo-ou-nada*, em que vários classificadores binários são treinados para reconhecer ou não uma das k classes de dados do problema (Crammer et al., 2001; Manning et al., 2008). Tal abordagem pode utilizar o resultado discreto da classificação do algoritmo, ou o valor contínuo da função f de decisão utilizada. De qualquer forma, a exigência do treinamento de várias redes impõe uma série de cuidados e restrições aos conjuntos de treinamento. Aplicado ao problema de identificação e classificação de ncRNAs, as medidas restritivas podem tornar inviável a seleção de exemplares para treinamento.

Para os resultados de análise do erro de quantização das redes, deve-se considerar o intervalo $qerror \in [0; 10, 82]$, baseado num valor mínimo e máximo atribuído a todas as 117 variáveis numéricas normalizadas extraídas para os diversos conjuntos de dados. O valor de $qerror$ extraído para a rede 2×2 treinada foi de $\approx 0, 765$. Nesse trabalho, redes com valores de $qerror > 1, 1$ de um conjunto qualquer de dados foram consideradas redes com considerável dispersão dos agrupamentos criados, o que corresponde a $\approx 10\%$ do valor máximo de $qerror$. O valor obtido para essa rede é satisfatório, porém, é um valor que, novamente, pode não refletir fidedignamente o grau de convergência da rede, dado o tamanho reduzido do conjunto de treinamento e do conjunto de validação $dbVal.dat$ utilizado. Todavia, o resultado dá um bom indício da eficácia do treinamento da rede não supervisionada utilizando um conjunto criado inicialmente para uma rede supervisionada baseada em SVM (Liu et al., 2006).

5.1.2 Estudo de Caso

Os métodos Infernal (Nawrocki et al., 2009) e CPC (Kong et al., 2007) foram executados em ambientes de trabalho diferentes do detalhado no Capítulo 4, inviabilizando a obtenção comparativa dos tempos de execução. Empiricamente, o tempo de execução desses dois métodos tende a ser elevado, por suas características respectivas de análise estrutural por meio de complexos modelos matemáticos (Nawrocki et al., 2009) e busca de homologia em grandes bancos de dados de proteínas (Kong et al., 2007).

Resultados com o *P. brasiliensis*

A verificação de formato válido de sequências, para o método Portrait, descartou 11 das 6.022 sequências de RNA do *P. brasiliensis*. A Tabela 5.3 mostra os resultados para a obtenção dos falsos positivos nos 4 métodos. Na tabela, as linhas referem-se às predições obtidas pelos diversos algoritmos, e as colunas referem-se, respectivamente, ao número de sequências preditas, a quantidade de sequências com semelhança elevada ao banco de

Tabela 5.3: Resultados para obtenção da taxa de falsos positivos dos 4 métodos avaliados para o organismo *P. brasiliensis*.

	Sequências	Sequências semelhantes	Falsos Positivos (%)
SOM-Portrait (ncRNA)	1.127	61	5,28%
Portrait (ncRNA)	959	41	4,28%
CPC (ncRNA)	1.985	30	1,51%
Infernal (ncRNA)	420	25	5,95%

proteínas Swiss-Prot (Boeckmann et al., 2002) utilizando BLAST (Altschul et al., 1997) com $e = 1 \cdot 10^{-5}$, conforme descrito na Seção 4.4 do Capítulo 4.

A porcentagem de falsos positivos mantém-se equivalente entre 3 dos 4 métodos. Os resultados do método Infernal, proporcionalmente, são equivalentes, porém, o método conseguiu identificar menos ncRNAs nas sequências do *P. brasiliensis*. O Infernal utiliza um valor equivalente ao escore de similaridade e do BLAST. Para o experimento, utilizou-se $e = 0,01$, uma medida convencional de busca utilizando toda a extensão de famílias do Rfam (Gardner et al., 2009). Apesar disso, os resultados sugerem que o valor de e foi muito restritivo, por causa do baixo número de similaridades obtidas. Mesmo com tal restrição, a taxa de falsos positivos seguiu a proporção encontrada nos outros métodos, exceto para o método CPC, com apenas $1,51\%$ de sequências semelhantes ao banco Swiss-Prot. O bom desempenho do CPC, por sua vez, é resultado direto do uso de informação de homologia com o banco de proteínas UniRef90 (Kong et al., 2007; swi, 2012).

Observação pertinente é dirigida ao valor de falsos positivos obtido pelo método SOM-Portrait. Apesar da verificação, na fase de validação do treinamento, de um desequilíbrio entre precisão e *recall*, que poderia incorrer em aumento de predições incorretas de ncRNAs, constatou-se uma proporção similar aos outros métodos. Particularmente, Portrait baseia-se num conjunto de treinamento diferente do conjunto utilizado pelo SOM-Portrait nesse experimento. Isso consolida o entendimento de que o método não sofre com o problema de baixa sensibilidade.

Outra ponderação importante é a interpretação da primeira linha, referente a mRNAs putativos, incluídos para controle da especificidade da contrapartida negativa da classificação do método. O método BLAST, nesse caso, atua para verificar a taxa de verdadeiros negativos com forte sinal codificante. Como indicado na tabela, mais da metade das sequências apresenta tal sinal. Comparado às informações disponíveis sobre o organismo, $\approx 38\%$ dos transcritos correspondem a sinais codantes confiáveis, conforme anotação do projeto (Arrial et al., 2009), proporção respeitada pela classificação do método SOM-Portrait. Essas informações indicam boa especificidade do método para o conjunto de teste desse organismo.

A Figura 5.3 retrata o diagrama de *Venn* entre os métodos CPC, Portrait (SVM) e SOM-Portrait (SOM). Os resultados mostram, em média, uma concordância de aproximadamente metade das sequências putativas ncRNAs dos métodos. Apesar de divergentes, as predições podem ser refinadas por uma medida de qualidade para obter as sequências de consenso unânime entres os métodos. Tal medida é dada por um valor de corte fornecido ao método Infernal. Essa interseção entre os 4 métodos (incluindo o método Infernal) foi de aproximadamente 140 sequências, reduzida pelo número reduzido de sequências pu-

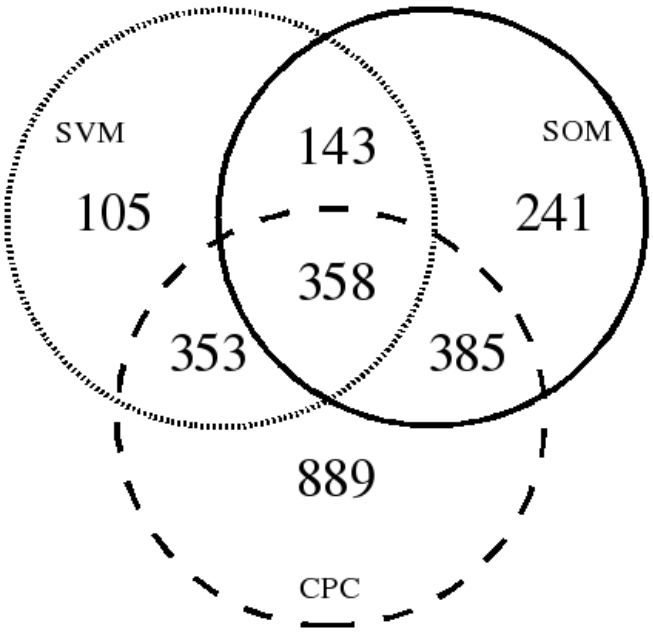


Figura 5.3: Comparações entre os conjuntos de putativos ncRNAs dos métodos CPC, Portrait (SVM) e SOM-Portrait (SOM) para o organismo *P. brasiliensis*.

tativas ncRNAs encontradas pelo Infernal. O Infernal utiliza exclusivamente informação estrutural sobre ncRNAs. Alguns ncRNAs, porém, não contém informação estrutural, ou contém informação pouco descrita nos bancos de referência utilizados pelo programa. Isso reduz o poder classificatório da ferramenta, principalmente por não conter informação explícita de exemplares codantes.

A proximidade entre SOM-Portrait e CPC é mais acentuada, apesar do SOM-Portrait não utilizar informação de homologia que o CPC utiliza, mostrando que a rede é capaz de assimilar informações complexas sobre as sequências sem precisar de métodos que exigem recursos computacionais vultosos. Outra observação é a de quantidade de sequências. O método CPC, com quase 2.000 sequências putativas ncRNAs, tem as maiores interseções com outros métodos, sem implicar, necessariamente, na qualidade das predições. Os autores utilizam muitas informações relacionadas a proteínas para construir o vetor de características (Kong et al., 2007), o que pode causar falta de bons parâmetros para a identificação do conjunto negativo, e reduzir a precisão do algoritmo. No método SOM-Portrait, o cuidado com a seleção de informações relevantes para a identificação de ncRNAs, como concentração de dinucleotídeos GC, presença de pequenas ORFs, baixa complexidade do peptídeo formado, entre outras, é um diferencial que tenta minimizar esse risco.

Finalmente, considerando-se a proposta de *pipeline* idealizada na Seção 4.4 do Capítulo 4 para anotação automática de ncRNAs, essas 140 sequências teriam anotação equivalente ao trabalho de 4 identificadores diferentes, utilizando sinais estruturais, composicionais, termodinâmicos e por homologia para sua decisão final.

Tabela 5.4: Resultados para obtenção da taxa de falsos positivos dos 4 métodos avaliados para o organismo *C. immitis*.

	Sequências	Sequências semelhantes	Falsos Positivos (%)
SOM-Portrait (ncRNA)	1.072	48	4,48%
Portrait (ncRNA)	850	18	2,12%
CPC (ncRNA)	851	10	1,18%
Infernal (ncRNA)	474	28	5,91%

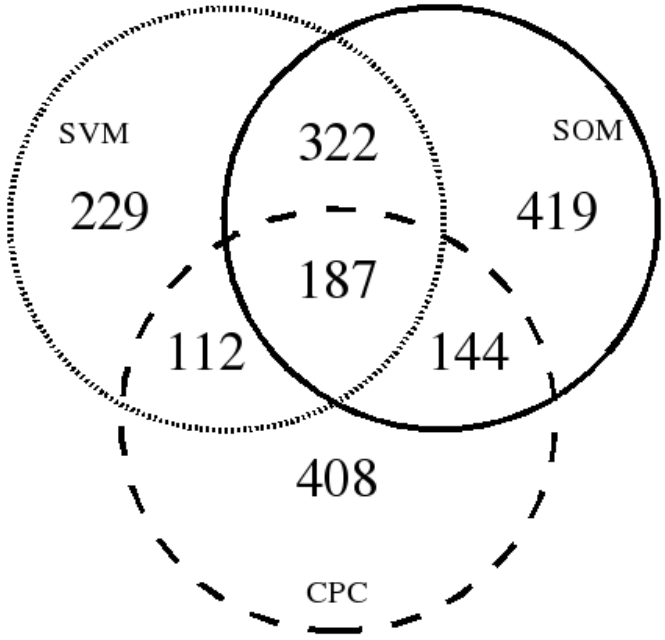


Figura 5.4: Comparações entre os conjuntos de putativos ncRNAs dos métodos CPC, Portrait (SVM) e SOM-Portrait (SOM) para o organismo *C. immitis*.

Resultados com o *C. immitis*

Nenhuma sequência foi descartada na filtragem das 9.757 sequências de RNA do fungo *C. immitis*. A Tabela 5.4 mostra os resultados para a obtenção dos falsos positivos nos 4 métodos.

Comparados aos resultados obtidos para o organismos *P. brasiliensis*, a Tabela 5.4 é muito semelhante. Dada a proximidade filogenética entre os dois organismos dos métodos, a proximidade de resultados é realmente esperada. O número reduzido de Falsos Negativos obtidos em ambos os experimentos é decorrente da especificidade obtida de $0,862\%$. De uma forma geral, a rede SOM identificou melhor a contrapartida negativa do conjunto de validação e também, de forma presumida, nos testes. Análises biológicas mais detalhadas devem ser ainda realizadas para confirmar a performance da rede com esses dados de fungos. A Tabela 5.4 mostra as comparações dos conjuntos de putativos ncRNAs entre os métodos CPC, Portrait e SOM-Portrait.

Seguindo o comportamento evidenciado no experimento com o fungo *P. brasiliensis*, a Figura 5.4 demonstra, mais uma vez, que o método SOM-Portrait, mesmo ao identificar

Tabela 5.5: Resultados para obtenção da taxa de falsos positivos dos 4 métodos avaliados para o organismo *A. oryzae*.

	Sequências	Sequências semelhantes	Falsos Positivos (%)
SOM-Portrait (ncRNA)	1.127	61	26,13%
Portrait (ncRNA)	1.445	264	18,27%
CPC (ncRNA)	1.868	223	11,94%
Infernal (ncRNA)	376	110	29,26%

quantitativamente mais ncRNAs do que os outros métodos, não se distancia das predições desses métodos. Pelo contrário, infere-se que a rede treinada conseguiu assimilar as principais características exploradas pelas redes neurais de CPC e Portrait. Essa assimilação, em redes de alta plasticidade como SOM e ART, pode ser explicada pela alta complexidade do espaço multidimensional dos vetores de entrada que a função de *kernel* da rede SVM precisa tratar.

Além disso, é interessante notar que, apesar dos métodos Portrait e CPC terem encontrado número de sequências próximos, 850 e 851, respectivamente, a Figura 5.4 mostra que os dois métodos tiveram as menores interseções. Em contrapartida, as interseções entre SOM-Portrait e os dois métodos se mantêm proporcionalmente iguais aos outros experimentos, sendo que o número de putativos ncRNAs é um pouco maior. Esses resultados reforçam a idéia de um *pipeline* envolvendo os vários métodos utilizados para uma anotação automatizada de ncRNAs mais segura, acelerando a análise e estudo de ncRNAs.

Resultados com o *A. oryzae*

Nenhuma sequência foi descartada na filtragem das 9.051 sequências de RNA do fungo *A. oryzae*. A Tabela 5.5 mostra os resultados para a obtenção dos falsos negativos nos 4 métodos.

É importante notar que a análise textual das anotações do organismo não conseguiu filtrar possíveis proteínas putativas e hipotéticas, portanto, o número de Falsos Positivos pode ainda ser bastante reduzido. De uma maneira geral, assim como nos organismos *P. brasiliensis* e *C. immitis*, o método SOM-Portrait parece se aproximar de forma equivalente dos métodos Portrait e CPC, evidenciando talvez uma assimilação das características fundamentais de classificação utilizadas pelos dois métodos. A porcentagem de Falsos Positivos, próxima à encontrada para os dois métodos, indica que essa assimilação não significa que os critérios de classificação foram relaxados, pelo contrário, pode-se inferir que o método SOM-Portrait conseguiu incorporar informações bastante diferentes entre os dois métodos. A Figura 5.5 torna evidente essa análise.

Como dito anteriormente, o método SOM-Portrait pode representar um consenso entre os métodos CPC e Portrait, incorporando sequências putativas ncRNAs de ambos os métodos, sem perder com isso rigor na classificação. Adicionando o método Infernal, obteve-se uma interseção final entre os 4 métodos de 120 sequências. Os resultados equivalentes entre os três organismos, filogeneticamente próximos, reforça a correição do método SOM-Portrait, que evidenciou comportamentos parecidos para esses organismos.

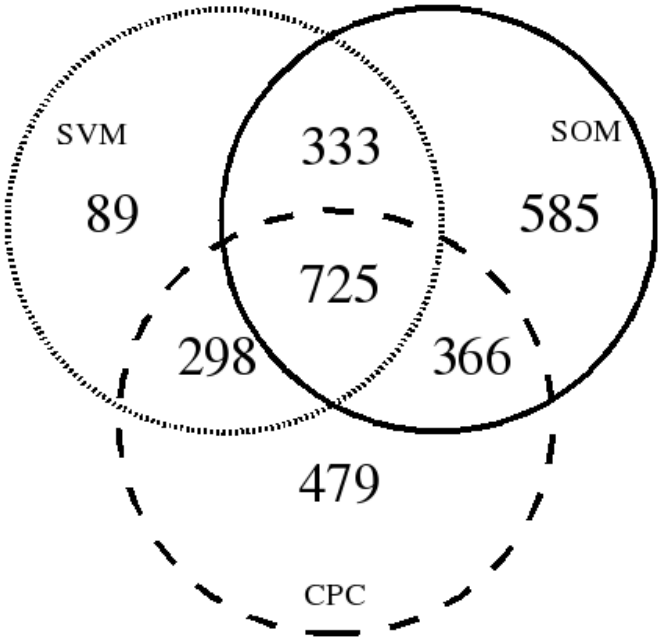


Figura 5.5: Comparações entre os conjuntos de putativos ncRNAs dos métodos CPC, Portrait (SVM) e SOM-Portrait (SOM) para o organismo *A. oryzae*.

Os resultados discutidos no Primeiro Experimento conduziram as atividades iniciais do trabalho. Revelou-se boa proposta a utilização de classificadores de ncRNAs baseados em métodos por AM não supervisionados ao problema. Apesar dos problemas de pequenos conjuntos de treinamento e validação, a rede foi capaz de generalizar o conhecimento adquirido de forma satisfatória. As próximas análises referem-se ao Segundo Experimento, realizados numa etapa posterior desse trabalho, que tentam resolver os vários problemas e desafios encontrados nessa primeira abordagem, conforme explanado na Seção 4.1.

5.2 Segundo Experimento

De forma semelhante, apresentam-se os resultados do Segundo Experimento. O Segundo Experimento versa sobre a utilização de um conjunto de treinamento especialmente fabricado para o problema de identificação e classificação de ncRNAs utilizando redes neurais de treinamento não supervisionado. Para medida de desempenho da rede, novo caso de estudo também foi designado, conforme detalhes informados na Seção 4.3.

Para os algoritmos SOM, ART e LVQ, os resultados do treinamento, validação da rede e estudo de caso são explicados, bem como os resultados da Análise de Componente Principal aplicada aos atributos numéricos utilizados pelos métodos.

5.2.1 Treinamento da rede SOM

Utilizando os novos conjuntos de treinamento, o procedimento de pesquisa da melhor acurácia relativa ao conjunto de validação através da variação do número máximo de

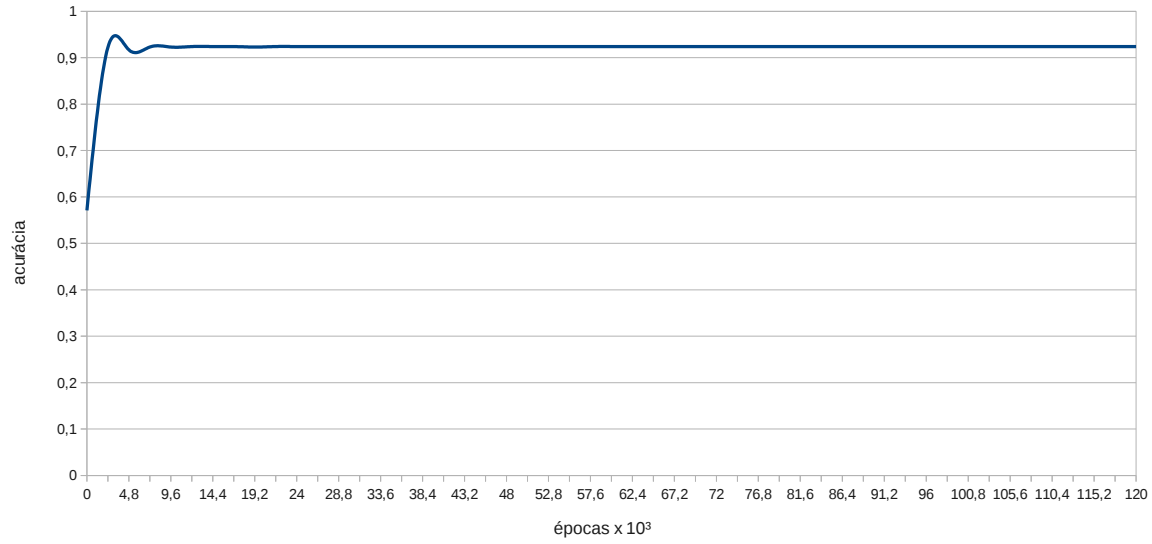


Figura 5.6: Resultados de acurácia do treinamento da rede SOM com topologia 2×1 retangular utilizando o conjunto de validação $dbVal2.dat$.

épocas foi conduzido para redes SOM de 2×1, 3×1 e 3×2 nós na camada de saída. Os resultados para a rede 2×1 são exibidos na Figura 5.6.

Os valores de épocas máximas E transitaram de 0 a 120.000. O valor $E = 120.000$ máximo foi escolhido como limite superior para que cada estímulo de entrada x_i tenha a probabilidade de aproximar-se dos 2 neurônios da camada de saída, tendo em vista os 60.000 exemplares do conjunto de treinamento. Aliado à pequena taxa de aprendizado, esse procedimento permite um aprendizado mais vagaroso, ideal para essa etapa de pesquisa. Todavia, o gráfico demonstra que a conversão ocorreu de forma rápida, possivelmente por causa da semelhança dos conjuntos $dbTr2.dat$ de treinamento e $dbVal2.dat$ de validação.

Partindo de uma acurácia próxima de uma situação aleatória, a rede rapidamente converge para uma acurácia média de $92,4\%$. A época final escolhida para treinamento da rede 2×1 foi de 120.000 para a etapa de ordenação e 1.200.000 para a etapa de convergência.

A Figura 5.7 exibe o gráfico de treinamentos para a rede 3×1. Os valores de épocas máximos escolhidos foram $0 \leq E \leq 360.000$. Novamente, uma situação de estabilidade da acurácia foi atingida rapidamente. O valor médio de acurácia obtido na situação nessa situação foi de $91,1\%$. A redução sofrida não era esperada, pelos resultados obtidos no Primeiro Experimento, em que redes de até 2×2 neurônios foram treinadas com valores de acurácia próximos de 88%. É importante lembrar, como justificativa desse comportamento, que o aumento de estímulos de entrada altera significativamente todo o processo de treinamento.

Na Figura 5.8, os resultados para os diversos treinamentos da rede 3×2 são exibidos. A acurácia ótima calculada para o experimento foi de $91,8\%$. As flutuações observáveis entre 100.000 e 200.000 épocas de treinamento reforçam o entendimento dos valores de época máxima utilizados, proporcionais ao número de neurônios a serem estimulados na camada

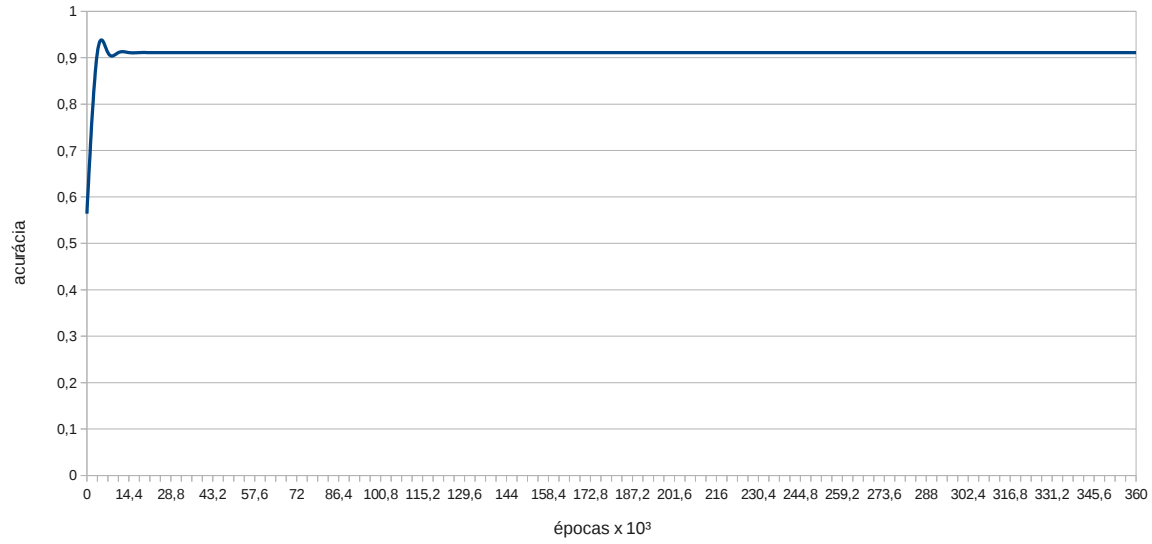


Figura 5.7: Resultados de acurácia do treinamento da rede SOM com topologia 3×1 retangular utilizando o conjunto de validação $dbVal2.dat$.

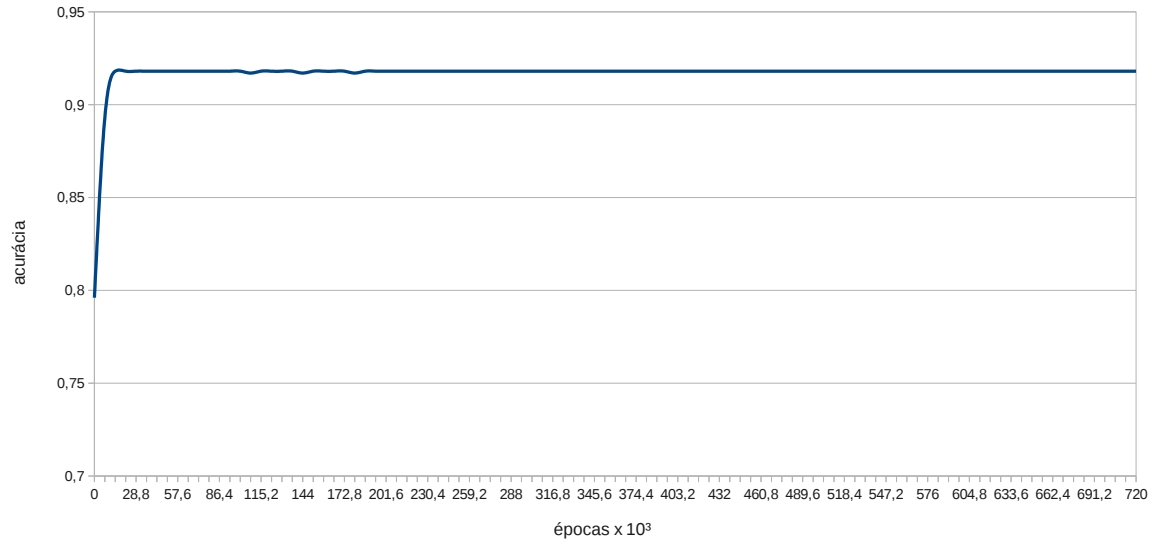


Figura 5.8: Resultados de acurácia do treinamento da rede SOM com topologia 3×2 retangular utilizando o conjunto de validação $dbVal2.dat$.

de saída. Uma observação experimental recorrente foi a de mudança de nomeações dadas às classes durante o treinamento, comportamento que influencia diretamente a acurácia medida pelo procedimento.

Os valores de época máxima selecionados para as etapas de ordenação e convergência do algoritmo SOM aplicado à rede 3×2 foram de, respectivamente, 720.000 e 7.200.000.

Tabela 5.6: Matriz de confusão com resultados da validação da rede SOM 2×1 utilizando o conjunto $dbVal2.dat$.

		Predito	
		Classe 1	Classe 2
Real	Codificante	1249	23751
	Não Codificante	21266	3734

Tabela 5.7: Medidas de performance P para a rede SOM 2×1.

Precisão	*Recall*	Especificidade
$0,851$	$0,945$	$0,864$
Acurácia	**Medida F**	**MCC**
$0,900$	$0,895$	$0,670$

5.2.2 Validação da rede SOM

As três redes descritas anteriormente foram treinadas utilizando o conjunto de validação $dbVal2.dat$. Para a rede 2×1 treinada, a Tabela 5.6 sumariza as predições obtidas nas colunas, uma para cada neurônio da camada de saída, confrontadas com o resultado esperado (linhas "Real").

Dos resultados exibidos, depreende-se que a Classe 1 representa o grupamento positivo, com sinal não codificante mais intenso do que na Classe 2, que representa o grupamento negativo. É importante notar que, durante os experimentos, nem todas as redes obtiveram uma separação entre conjuntos codantes e não codantes; os resultados ruins oriundos de más escolhas para os valores de épocas máximas. Como o algoritmo não possui outra forma de critério de parada, vários experimentos tiveram que ser realizados até encontrar uma conformação estável para o conjunto de validação $dbVal2.dat$. Aqui, é mister reiterar a necessidade de um conjunto de validação com variabilidade e número de exemplares próximo do conjunto de treinamento, propriedade fundamental para uma boa validação do método.

A partir dos dados exibidos na matriz da Tabela 5.6, os cálculos das medidas de performance foram realizados. Os resultados aparecem na Tabela 5.7. Nesse trabalho, os identificadores e classificadores de ncRNAs que não utilizam informações específicas de famílias de ncRNA obtiveram valores de especificidade maiores do que os de sensibilidade (Liu et al., 2006; Arrial et al., 2009). No caso do PORTRAIT, as observações são confirmadas com dados do autor. No caso do CPC, o resultado evidencia um comportamento esperado, devido à grande quantidade de informação referente a proteínas utilizado pelo programa. De forma geral, entretanto, classificadores de ncRNAs não específicos tendem a apresentar maior coesão do sinal codificante. Os sinais não codantes são mais variáveis, consequência óbvia da imensa variedade de famílias, classes e estruturas de ncRNAs existentes.

O alto valor de *recall* obtido pela rede, acompanhado por um valor inferior de precisão, indicam a tendência da rede SOM treinada de escolher menos candidatos a ncRNAs, porém com maior qualidade. O valor de *qerror* extraído para a rede 2×1, utilizando o conjunto de validação $dbVal2.dat$ foi de $0,934$, valor próximo ao obtido pela rede 2×2 no Primeiro

Figura 5.9: Representação por u-matriz da rede SOM 2×1 treinada com o conjunto $dbTr2.dat$.

Tabela 5.8: Matriz de confusão com resultados da validação da rede SOM 3×1 utilizando o conjunto $dbVal2.dat$.

		Predito		
		Classe 1	Classe 2	Classe 3
Real	Codificante	1241	864	22895
	Não Codificante	21262	619	3119

Experimento, de $\approx 0,765$. Levando-se em conta o número reduzido de classes, que pode ocasionar maior variância dos sinais dentro de um agrupamento, e, principalmente, do número muito reduzido de sequências de validação utilizadas para obter o valor de *qerror* no Primeiro Experimento, os valores encontrados para o erro de quantização podem ser considerados equivalentes. Esse entendimento demonstra que o conjunto de treinamento criado tem boa conformação de exemplares de ambos os controles. O valor aprimorado de MCC obtido também consolida os resultados favoráveis dessa nova rede.

Analisando o espaço de decisão descrito pela u-matriz da rede 2×1 treinada, representada pela Figura 5.9, a divisão entre os agrupamentos "Coding" e "Noncoding" é bem visível, representada pela separação dos vetores de peso por grandes distâncias (representadas pela cor escura). Já o comportamento dentro dos *clusters* é refletido na imagem pelos conjuntos codificante e não codantes tonalizados em branco, indicando conjuntos com dados bastante esparsos. Essa observação confirma o valor elevado de *qerror* encontrado.

A análise da u-matriz do mapa 2×1 evidencia o potencial de construção de outras classes coerentes no mapa SOM. Partindo desse pressuposto, a análise do mapa 3×1 é realizada em moldes semelhantes. A Tabela 5.8 os resultados da validação.

Frente aos desempenhos bem caracterizados das Classes 1, para sinais não codantes, e 3, para sinais codantes, os resultados da Classe 2, proporcionalmente, podem ser considerados inconclusivos. Para o cálculo das medidas de desempenho da Tabela 5.9, entretanto, considerou-se o sinal mais numeroso, portanto codificante. Análises mais detalhadas, no entanto, devem esclarecer a composição dessa classe, em termos de características dos exemplares do conjunto de validação que pertencem a essa classe.

Comparado aos resultados da rede 2×1, na Tabela 5.7, a rede 3×1 parece piorar o desempenho do classificador, principalmente no tocante ao controle positivo. A queda do coeficiente de correlação é resultado direto da perda de precisão. Dada a escolha arbitrária de classes realizada para a coleta desses valores, pode-se argumentar quanto ao erro dessa

Tabela 5.9: Medidas de performance P para a rede SOM 3×1.

Precisão	*Recall*	Especificidade
$0,850$	$0,945$	$0,864$
Acurácia	**Medida F**	**MCC**
$0,900$	$0,895$	$0,629$

Figura 5.10: Representação por u-matriz da rede SOM 3×1 treinada com o conjunto $dbTr2.dat$.

escolha utilizando a análise menos generalizada do treinamento dada pelo valor de *qerror* dos estímulos de entrada do conjunto de validação. O valor de *qerror* encontrado foi de $0,68$, o que indica uma queda considerável do erro associado às classificações dadas pelo método. Esse comportamento indica que a nomeação de classes arbitrária dada para a coleta dos valores de performance da rede realmente não conseguem refletir os agrupamentos criados pelo treinamento SOM.

Para tentar elucidar a composição dessa misteriosa Classe 2, a representação por u-matriz, dada pela Figura 5.10, foi construída. Os resultados mostram que a Classe 2 é bastante separada tanto de sinais codantes característicos tanto de sinais não codantes. Essa distinção bem caracterizada, indício de forte sinal, é um resultado interessante para o problema de classificação em múltiplas classes. Uma abordagem utilizando BLAST *versus* banco de dados de proteínas Swiss-Prot foi novamente utilizada para tentar compreender qual é a composição majoritária dessa classe.

Os parâmetros de execução dessa nova rodada da ferramenta BLAST foram restritivos, novamente com corte de qualidade do alinhamento ajustado para $e = 1 \cdot 10^{-5}$, retornando somente o melhor alinhamento encontrado para cada exemplar analisado. A análise é separada para os exemplares codantes e não codantes do conjunto de validação $dbVal2.dat$, conforme disposto na Tabela 5.10. O número de acertos obtidos para cada classe e sua porcentagem relativa ao número total de exemplares na classe, o valor médio, variância e valores máximo e mínimo de *e-value* obtido para cada *cluster* também é reportado.

Os resultados mostram um comportamento muito peculiar da Classe 2. Para as sequências codantes, observando o valor médio de *e-value* obtido por seus exemplares, bem como o percentual de acertos obtidos no Swiss-Prot, pode-se concluir que a Classe 2 aproxima-se dos sinais codantes. Contudo, ao analisar a contrapartida não codificante, a Classe 2 parece agir também como agrupamento de sequências não codantes com fortes características codantes, ou talvez identificando falsos positivos presentes no subconjunto não codificante do conjunto $dbVal2.dat$, hipótese nunca descartada no complexo mundo nos ncRNAs.

Finalmente, as análises de classificação do conjunto de validação pela rede 3×2 são resumidas na Tabela 5.11. A importância dos resultados desse experimento residem na concepção do mapa 3×2 utilizado. Sua motivação foi dada pelos bons resultados de convergência obtidos com a rede ART, a serem descritos e analisados na Subseção 5.2.4.

Tabela 5.10: Análises utilizando ferramenta BLAST para as sequências dos *clusters* construídos a partir da execução da rede SOM 3×1 usando o conjunto $dbVal2.dat$.

Sequências Codantes			
	Classe Não Codificante	Classe 2	Classe Codificante
Acertos	941(75.83%)	735(85.07%)	2830(90.73%)
Média	1.85×10^{-7}	8.28×10^{-8}	4.45×10^{-9}
Variância	9.79×10^{-13}	4.68×10^{-13}	4.33×10^{-15}
Máximo	9.00×10^{-6}	1.00×10^{-5}	2.00×10^{-6}
Mínimo	6.00×10^{-68}	1.00×10^{-121}	1.00×10^{-180}
Sequências Não Codantes			
	Classe Não Codificante	Classe 2	Classe Codificante
Acertos	62(0.29%)	28(4.52%)	7(0.03%)
Média	9.39×10^{-7}	1.89×10^{-7}	1.29×10^{-6}
Variância	6.30×10^{-12}	4.46×10^{-13}	1.16×10^{-11}
Máximo	1.00×10^{-5}	3.00×10^{-6}	9.00×10^{-6}
Mínimo	2.00×10^{-35}	2.00×10^{-44}	1.00×10^{-82}

Tabela 5.11: Matriz de confusão com resultados da validação da rede SOM 3×2 utilizando o conjunto $dbVal2.dat$.

		Predito					
		Classe 1	Classe 2	Classe 3	Classe 4	Classe 5	Classe 6
Real	Codificante	3.694	8	56	19.520	529	1.193
	Não Codificante	455	0	15.022	1.777	1.167	6.579

Tabela 5.12: Medidas de performance P para a rede SOM 3×2.

Precisão	*Recall*	Especificidade
$0,911$	$0,928$	$0,912$
Acurácia	Medida F	MCC
$0,920$	$0,919$	$0,763$

A divisão de sequências entre os agrupamentos evidencia uma distribuição principal de sequências não codantes nas Classes 3 e 6, enquanto que as Classes 1 e 4 agem como controle negativo da rede. A Classe 2 teve exercício mínimo, estimulada por 8 exemplares do subconjunto negativo de $dbVal2.dat$. Apesar do resultado inconclusivo, consideram-se codantes as Classes 1, 2 e 4, e não codantes as Classes 3, 5 e 6. De posse dessas denominações, a Tabela 5.12 apresenta os valores de performance do mapa.

O valor de acurácia obtido foi o mais próximo da acurácia estimada durante o treinamento, mostrando uma estabilidade sem precedentes da rede. Essa estabilidade nas classificações é verificada pelo valor de $qerror = 0,59$, o mais baixo encontrado até o momento. Os problemas de precisão e *recall* parecem diminuir também, exibindo o melhor desempenho de MCC confrontado às redes 2×1, 3×1 do Segundo Experimento, e 2×2, do Primeiro Experimento.

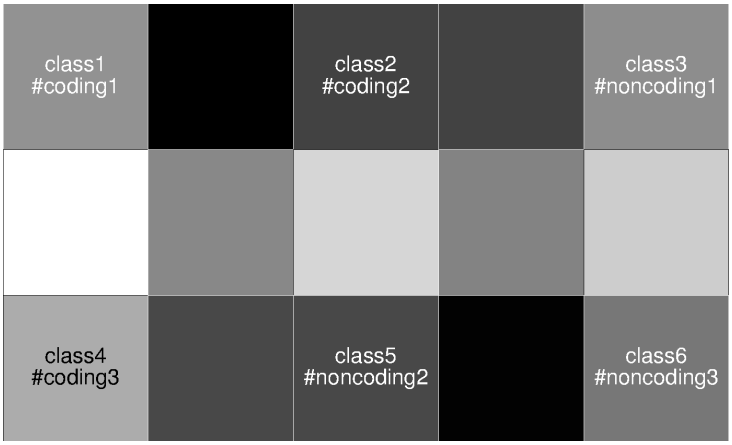


Figura 5.11: Representação por u-matriz da rede SOM 3×2 treinada com o conjunto $dbTr2.dat$.

A distribuição aparentemente não convencional das classes é explicada pela topologia da rede e consequente nomeação dos agrupamentos. A Figura 5.11 clarifica essas definições.

É notável a divisão precisa do espaço de decisão em dois grandes agrupamentos, com as classes mais representativas de cada conjunto de controle assumindo os extremos da configuração espacial. Pode-se afirmar que a redução dimensional de um complexo espaço de 117 variáveis numéricas obteve resultados coerentes com a expectativa de classificação, baseado nos dados da matriz de confusão e à configuração espacial visível na u-matriz. Dentre todas as classes exibidas, a Classe 2 parece se aproximar mais de sinais não codantes do que de sinais codantes, principalmente da Classe 5, formando um 3^o agrupamento destacado tanto dos sinais codificantes das Classes 1 e 4, quanto dos sinais não codificantes das Classes 3 e 6. A quinta classe, por sua vez, contém $\approx 33\%$ de sua composição formada por exemplares codantes. Essa interação comum de duas classes tão heterogêneas é marcada por sua posição no espaço de decisão, exatamente na fronteira entre os agrupamentos mais concisos. Análises de *qerror* específicos para cada classe podem clarificar esse entendimento. Espera-se um valor mais alto de *qerror* nessas fronteiras, o que confirmaria o caráter fronteiriço de sequências classificadas nessa região. Para fins de anotação biológica de ncRNAs, sequências dessa classe poderiam ser descartadas segundo a justificativa de baixa qualidade de classificação.

5.2.3 Estudo de caso da rede SOM

A execução da predição das 1.121 sequências de teste detalhadas na Subseção 4.3.2, para todas as diferentes redes treinadas, demandou $\approx 15min$ de tempo real de usuário, conforme o comando *time* do Unix. Os experimentos foram executados para todas as redes do Segundo Experimento, exceto a rede 2×1, descartada por causa de seu alto valor de *qerror*, em comparação às outras redes treinadas. A Tabela 5.13 sumariza os valores encontrados. As primeiras linhas sumarizam as estatísticas de execução por organismo, incluindo a informação de quantidade de sequências e de sequências descartadas pelo método SOM-Portrait na etapa de pré-processamento. Finalmente, as análises são divididas pela nomeação "Coding" ou "Noncoding" dada na etapa de validação da rede.

Tabela 5.13: Resultados do estudo de caso da rede 3×1 usando o conjunto de ncRNAs de 4 organismos filogeneticamente distantes.

	H. sapiens	*E. coli*	*S. cerevisiae*	*A. thaliana*
Sequências	133	154	413	421
Descartadas	0	0	0	6
Não Codantes	130	137	386	415
Codantes	2	15	15	0
Classe 2	1	2	12	0

Tabela 5.14: Resultados do estudo de caso da rede 3×2 usando o conjunto de ncRNAs de 4 organismos filogeneticamente distantes.

	H. sapiens	*E. coli*	*S. cerevisiae*	*A. thaliana*
Sequências não Codantes	130	139	388	415
Codantes	3	15	15	0

Especificamente para a rede 3×1, as predições da Classe 2 também são exibidas, devido à discussão sobre sua melhor interpretação feita na Subseção 5.2.2. As 6 sequências descartadas dos snoRNAs do organismo *A. thaliana* eram pequenas demais $S < 30nt$ para análise.

Nos resultados, dá-se dois valores de acurácia, dependentes da interpretação dada à Classe 2. Se interpretada como tendo somente sinais codantes, o valor de acurácia encontrado é de $0,958$. Caso a consideração de que a Classe 2 é constituída por sequências não codantes que contém sinais codantes, a acurácia medida atinge $0,971$. Ambos os resultados são bons, e equivalentes a métodos supervisionados baseados em SVM Liu et al. (2006); Wang et al. (2006); Kong et al. (2007); Arrial et al. (2009).

Para a rede 3×2, a matriz com os resultados do estudo de caso é exibida na Tabela 5.14. A acurácia obtida foi de $0,961$. Os valores encontram-se numa faixa intermediária entre a primeira e a segunda acurácia extraídas para a rede 3×1, mostrando como a inclusão de classes na rede possibilitou a determinação dos exemplares no espaço de decisão sem perder acurácia.

Etapa supervisionada usando LVQ

A aplicação da etapa supervisionada LVQ foi realizada nos mapas 3×1 e 3×2. É importante reiterar que a validação dos resultados foi feita somente através das análises de u-matriz e resultados do estudo de caso proposto. Os valores de *qerror* constam somente para identificar erros grosseiros no treinamento, não servindo para validar a melhoria do espaço de decisão, devido à sua constituição. A Tabela 5.15 inaugura os resultados do estudo de caso para a rede 3×1 melhorada com a etapa supervisionada LVQ. A apresentação é feita de forma similar às tabelas de resultados da Seção 5.2.3. Na primeira

Tabela 5.15: Resultados do estudo de caso sobre ncRNAs de 4 organismos filogeneticamente distantes da rede 3×1 treinada com etapa supervisionada LVQ.

	H. sapiens	*E. coli*	*S. cerevisiae*	*A. thaliana*
Sequências não Codantes	130	137	386	415
Codantes	3	17	27	0
Não Codantes	130	138	386	415
Codantes	3	16	27	0

parte da tabela, os resultados do treinamento LVQ com o conjunto *dbLvqOpt*1.*dat* são apresentados, e na segunda parte, os resultados utilizando o conjunto *dbLvqOpt*2.*dat*.

O valor de *qerror* encontrado para o algoritmo LVQ aplicado à rede 3×1 utilizando o conjunto *dbLvqOpt*1.*dat* foi de $0,48$. Já o resultado de *qerror* para LVQ utilizando o conjunto *dbLvqOpt*2.*dat* acumulou $qerror = 0,63$. Os dois valores apontam para uma convergência suficiente do algoritmo LVQ. Para a construção dos resultados da Tabela 5.15, o pior resultado de acurácia obtido foi utilizado, ou seja, a Classe 2 foi considerada codificante. Os resultados mostram uma acurácia de $0,957$ para o primeiro conjunto e $0,958$ para o segundo. Comparado ao valor anterior de acurácia obtido, $0,958$, a aproximação dos dois métodos é bastante evidente.

A análise da u-matriz das redes 3×1 treinadas com o conjunto *dbLvqOpt*1.*dat* 5.12(a) e com o conjunto *dbLvqOpt*2.*dat* 5.12(b), exibem um comportamento muito interessante. Comparando-se os resultados exibidos na Figura 5.12(a) e 5.12(b) com os resultados da u-matriz do modelo SOM 3×2 na Figura 5.11, pode-se observar uma aproximação das fronteiras entre os agrupamentos das Classes 2 e 5 e, respectivamente, os agrupamentos codificantes das Classes 1 e 4, e os agrupamentos não codificantes das Classes 3 e 6. Além disso, na primeira figura, os sinais das Classes 3 e 6 aproximaram-se ainda mais, formando um agrupamento mais homogêneo. De forma totalmente oposta, o algoritmo LVQ, na segunda figura, tornou a classe codificante mais coesa. Esse comportamento simétrico pode ser explicado pela composição dos conjuntos de treinamento e com o comportamento observado anteriormente na Subseção 5.2.3.

Os sinais codantes mais numerosos no conjunto *dbLvqOpt*1.*dat*, por terem coesão de seu sinal maior do que a de sinais não codantes, corrige o espaço de decisão atenuando os pesos não codantes em favor dos pesos codantes. Por sua vez, isso aproxima as Classes 2 e 5 do agrupamento codificante das Classes 1 e 4. Já o conjunto *dbLvqOpt*2.*dat* contém somente ncRNAs estruturais em sua composição. De forma semelhante, o algoritmo exercita os pesos não codantes das Classes 2 e 5, aproximando-as das classes não codificantes 3 e 6. Nesse caso, os exemplares utilizados no conjunto de treinamento influenciaram a rede, numa "condução" de suas classes intermediárias de fronteira de acordo com o sinal predominante. Sinais codificantes reforçaram o conhecimento das classes codificantes e aproximaram a fronteira da região codificante, mostrando que sequências de RNA dessa fronteira têm características semelhantes a mRNA. Já o segundo conjunto, com informação predominantemente estrutural, consolidou os agrupamentos não codificantes, reforçando o conhecido critério estrutural para identificação de ncRNA, e também atraí-

(a) LVQ utilizando o conjunto *dbLvqOpt*1.*dat*.

(b) LVQ utilizando o conjunto *dbLvqOpt*2.*dat*.

Figura 5.12: Representação por u-matriz da rede SOM 3×1 com treinamento supervisionado LVQ.

Tabela 5.16: Resultados do estudo de caso sobre ncRNAs de 4 organismos filogeneticamente distantes da rede 3×2 treinada com etapa supervisionada LVQ.

	H. sapiens	*E. coli*	*S. cerevisiae*	*A. thaliana*
Sequências não Codantes	130	137	386	415
Codantes	50	22	69	71
Classe 2	47	7	44	71
Não Codantes	130	139	388	415
Codantes	3	15	25	0

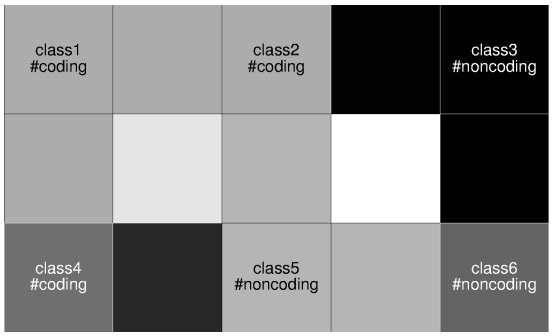
ram as classes de fronteira para sua posição. Isso demonstra que sequências nas classes de fronteira têm as duas propriedades, e que o método é capaz de discerni-las. Pode-se inferir que a composição dessa classe é de RNAs com características estruturais mais débeis e, também, características codantes pouco marcantes. Vários ncRNAs representantes da classe de longos ncRNAs (lncRNAs) tem, por exemplo, essa mesma característica (Mercer et al., 2009; Gibb et al., 2011). Além disso, mostra-se uma forma rápida e eficiente de se refinar a rede treinada para torná-la mais sensível a determinadas características próprias de ncRNAs ou de mRNAs. Estudos futuros poderão permitir uma especialização dessas redes, ou de determinadas classes de seus espaços de decisão, para identificar e classificar ncRNAs em famílias, ou grupos com propriedades em comum que se deseja estudar em maior profundidade.

A Tabela 5.16 resume os valores encontrados para a rede 3×2 com etapa supervisionada treinada com o conjunto *dbLvqOpt*1.*dat* e *dbLvqOpt*2.*dat*.

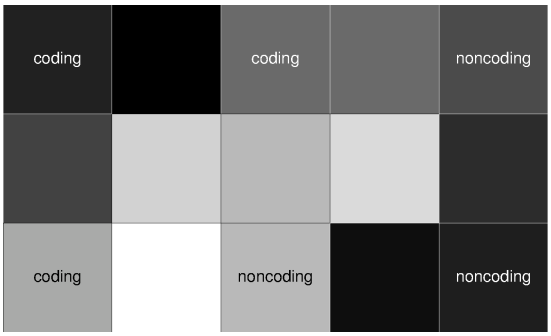
O valor de *qerror* encontrado para o algoritmo LVQ aplicado à rede 3×2 utilizando o conjunto *dbLvqOpt*1.*dat* foi de $0,28$. Já o resultado de *qerror* para LVQ utilizando o conjunto *dbLvqOpt*2.*dat* acumulou $qerror = 0,35$.

Para o treinamento LVQ com o primeiro conjunto, mantendo a nomenclatura anterior para o espaço de decisão, a acurácia despenca de $0,961$ para $0,809$. De maneira geral, o espaço de decisão para as classes secundárias (sem sinais dominantes) da rede 3×2 variou bastante, principalmente as nomeadas como codantes. De fato, ao alterar a nomeação da Classe 2 para "Noncoding", o valor de acurácia retorna para o valor encontrado anteriormente de $0,961$. De forma semelhante à análise da rede 3×1, a Classe 2 da rede 3×2 parece comportar-se novamente como fronteira entre ncRNAs sem grande definição estrutural e sequências de RNA com tendências codantes pouco definidas.

O treinamento LVQ utilizando o segundo conjunto acumulou acurácia de $0,961$, valor



(a) LVQ utilizando o conjunto *dbLvqOpt*1.*dat*.



(b) LVQ utilizando o conjunto *dbLvqOpt*2.*dat*.

Figura 5.13: Representação por u-matriz da rede SOM 3×2 com treinamento supervisionado LVQ.

idêntico ao obtido anteriormente. Esse resultado interessante conclui, de forma esperada, que, para essa etapa supervisionada, os exemplares tomados de acordo com funcionalidades bem delineadas, como RNAs com estrutura secundária bem definida, são mais relevantes do que a profusão de sinais diferentes presentes no primeiro conjunto de treinamento. Pode-se especializar as diferentes classes da rede de acordo com os conjuntos definidos, possibilitando a criação de classificadores capazes de reconhecer famílias ou conjuntos semelhantes de ncRNAs de forma eficiente.

A análise da u-matriz para o primeiro caso de treinamento LVQ 5.13(a) e para o segundo caso 5.13(b) revelam conclusão semelhante sobre a Classe 2. Entretanto, ponderação semelhante pode ser aplicada ao comportamento da Classe 5. Entretanto, a mudança da conformação dessa classe, para o estudo de caso realizado, não incorreu em mudança na acurácia, necessitando, portanto, de outras medidas para avaliação precisa.

5.2.4 Treinamento da rede ART

O treinamento da rede ART foi conduzido conforme explicado na Seção 4.5. O resultado do treinamento utilizando o conjunto *dbTr*2.*dat* é exibido na Figura 5.14. No Eixo y, ao lado esquerdo do gráfico, o número de *clusters* consolidados ao fim do treinamento é exibido pela linha vermelha, enquanto que, ao lado direito, os valores de flutuação associados são representados pelas barras azuis. O Eixo x representa os diversos valores de ρ iterados pelo procedimento de treinamento. A escolha de ρ foi conduzida de acordo com os resultados de ρ anteriores, e não de forma iterada, como no procedimento de treino de redes SOM. A duração de cada etapa de treinamento varia muito conforme a convergência da rede e os valores de ρ fornecidos. Em média, entretanto, cada etapa consumiu $15min$ na máquina de especificações definidas para o Segundo Experimento na Seção 4.4.

Partindo de uma situação muito restritiva do fator de vigilância ρ, em que todos os sinais foram agrupados num único protótipo, a partir do valor $\rho = 0,7$, a rede começa a diferenciar os estímulos em novos protótipos. O algoritmo alcança o critério de parada de flutuação $e < 0,05$ somente para $\rho = 0,74$, criando 6 diferentes *clusters*. Desse resultado decorre a construção da rede SOM 3×2 conforme explanado na Subseção 5.2.2.

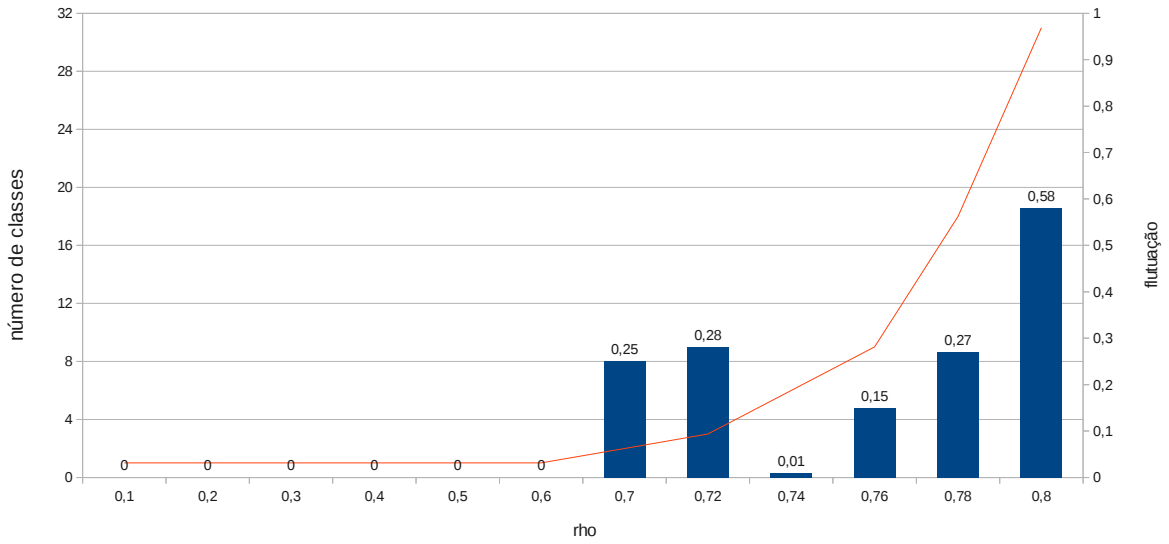


Figura 5.14: Gráfico de barras relacionando o valor de vigilância ρ adotado com o número de *clusters* consolidados ao final do treinamento da rede ART (linha vermelha) e sua respectiva flutuação atingida (barras azuis), indicada também pelo valor no topo de cada barra.

Tabela 5.17: Matriz de confusão com resultados da validação da rede ART de 6 classes utilizando o conjunto $dbVal2.dat$.

		Predito					
		Classe 1	Classe 2	Classe 3	Classe 4	Classe 5	Classe 6
Real	Codificante	918	275	3.702	19.520	56	529
	Não Codificante	5.635	607	455	1.777	16.247	279

Ressalta-se a comparativa facilidade, em relação ao método SOM, de escolha do número de classes ótimo para representar o conjunto de treinamento fornecido ao algoritmo ART, baseada no critério objetivo de estabilidade da rede.

5.2.5 Validação da rede ART

Para a validação da rede ART, a matriz de confusão dos resultados da predição do conjunto *dbval2.dat* foi construída de forma semelhante a Subseção 5.2.2. O resultado para a rede ART de 6 *clusters* é exibido na Tabela 5.17.

É notável a distribuição mais esparsa dos exemplares pelas diferentes classes. Pode-se alcançar, entretanto, a seguinte convenção para nomenclatura das classes: Classes 1, 2 e 5, predominantemente não codantes, e Classes 3, 4 e 6 predominantemente codantes. O valor de *qerror* encontrado para a rede ART foi de $1,11$, o que indica uma dispersão maior dos exemplares nas classes. Comparado ao valor obtido para a rede SOM 3×2, $0,59$, o aumento expressivo do erro de quantização pode ser justificado pela ausência de uma etapa de convergência dos estímulos no treinamento da rede ART, etapa essa com o propósito

Tabela 5.18: Medidas de performance P para a rede ART de 6 classes.

Precisão	*Recall*	Especificidade
$0,900$	$0,947$	$0,904$
Acurácia	Medida F	MCC
$0,925$	$0,923$	$0,760$

Tabela 5.19: Resultados do estudo de caso sobre ncRNAs de 4 organismos filogeneticamente distantes da rede ART com 6 classes.

	H. sapiens	*E. coli*	*S. cerevisiae*	*A. thaliana*
Sequências Não Codantes	130	137	386	415
Codantes	2	17	27	0

exclusivo de reduzir o erro de quantização da rede SOM. Não obstante essa observação, as medidas de performance da rede ART treinada, detalhadas na Tabela 5.18, demonstram o excelente treinamento realizado, com ótima acurácia e índice MCC bastante elevado. Novamente, os resultados do treinamento ART não superaram por pequena margem os resultados obtidos pela rede SOM.

5.2.6 Estudo de caso da rede ART

O estudo de caso para a rede ART treinada tem resultados exibidos de forma similar aos resultados da rede SOM na Subseção 5.2.3. A Tabela 5.19 mostra os resultados da predição do conjunto de ncRNAs pela rede ART.

O resultado mostra acurácia de $0,958$, comparável aos resultados obtidos pela rede SOM 3×2. De fato, a classificação obtida foi idêntica à rede SOM.

Etapa supervisionada usando LVQ

A rede ART treinada foi submetida ao treinamento não supervisionado LVQ utilizando os conjuntos *dbLvqOpt*1.*dat* e *dbLvqOpt*2.*dat*. Para o primeiro conjunto, o valor de *qerror* calculado foi de $0,28$. Para o segundo conjunto, esse valor subiu para $0,36$. Apesar da tendência de considerar esse um bom resultado, deve-se ter em mente que o valor é apenas referência para verificar um treinamento coerente, mas que, para efeitos de validação, não pode ser utilizado. Para validação, os valores do estudo de caso com ncRNAs dos 4 organismos são a única fonte de avaliação do grau de generalização da rede para o problema de classificação de ncRNAs. Tais valores são exibidos na Tabela 5.20.

Os resultados mostram acurácias de $0,847$ para o algoritmo treinado com o primeiro conjunto e $0,882$ para o segundo conjunto. A redução significativa da acurácia, configurando o pior resultado obtido para o presente estudo de caso, talvez seja decorrência da grande dispersão de sequências entre as classes criadas, comportamento observado na Subseção 5.2.6. Essa dispersão, por sua vez, pode ser consequência da adoção do valor de $\alpha = 0,01$, que não garante a busca por um protótipo existente coerente com cada sinal

Tabela 5.20: Resultados do estudo de caso sobre ncRNAs de 4 organismos filogeneticamente distantes da rede ART com 6 classes treinada com etapa supervisionada LVQ.

	H. sapiens	*E. coli*	*S. cerevisiae*	*A. thaliana*
Sequências não Codantes	100	133	348	363
Codantes	33	21	65	52
Não Codantes	116	135	356	376
Codantes	17	19	57	39

de entrada recebido. Isso pode criar agrupamentos artificiais dentro do espaço de decisão, que são muito penalizados pelo algoritmo LVQ.

Apesar dos baixos valores de acurácia obtidos, a aplicação do método ART para encontrar um número ótimo de classes representativas do espaço de decisão do conjunto de treinamento foi muito bem sucedida, culminando com a proposição da rede SOM 3×2, que obteve os melhores resultados para os experimentos propostos.

Já a etapa supervisionada LVQ aplicada ao final da validação dos métodos propostos mostrou-se peça fundamental para a crítica do espaço de decisão treinado pelas redes, e para nomeação mais criteriosa das classes, podendo inclusive ser aplicado para especialização de uma ou várias classes em torno de sinais específicos de alguns tipos ou famílias de ncRNAs.

5.2.7 Avaliação dos atributos usando PCA

As dimensões do vetor de características extraído das sequências no Passo 3 do método proposto nesse trabalho, descrito na Seção 4.1.1, foram analisadas à luz de seus componentes, utilizando dados conjuntos de autovalores e da matriz de correlação R. O mapa de indivíduos, ou exemplares, do conjunto de treinamento $dbTr2.dat$ também foi construído, como forma de ilustrar a variabilidade dos dados e identificar possíveis agrupamentos favoráveis à atuação das redes neurais de aprendizado não supervisionado utilizadas. O método PCA inicialmente resolveu a Equação 3.16, ordenando em seguida os autovalores de forma decrescente. Os dados detalhados dos resultados dessa análise são fornecidos como Material Complementar, nas tabelas $PCAx\_e\_y.csv$, x e y indicando as variáveis tomadas como referência para as análises. As variáveis 1 e 2 foram tomadas por referência, inicialmente. Nesse trabalho, exibem-se detalhadamente as análises realizadas para essas duas variáveis. As outras duas análises, para as variáveis 3 e 4 e para as variáveis 5 e 6, retornaram valores idênticos para a seleção das melhores variáveis.

De posse das variáveis de referência, o gráfico representado pela Figura 5.15 mostra a distribuição dos autovetores de todas as 117 variáveis i com contribuição de variância $\psi(q_i)$ relevante para a projeção $\pi(q_i, q_k)$, onde $k = \{1, 2\}$ as duas variáveis com maior contribuição de variância para o conjunto $dbTr2.dat$. A relevância da contribuição foi medida pelo valor $\cos^2 \pi > 0,1$. A nomenclatura utilizada no gráfico, em inglês, nomeia de forma equivalente nt (nucleotídeos) como bp (*base pair* ou pares de bases).

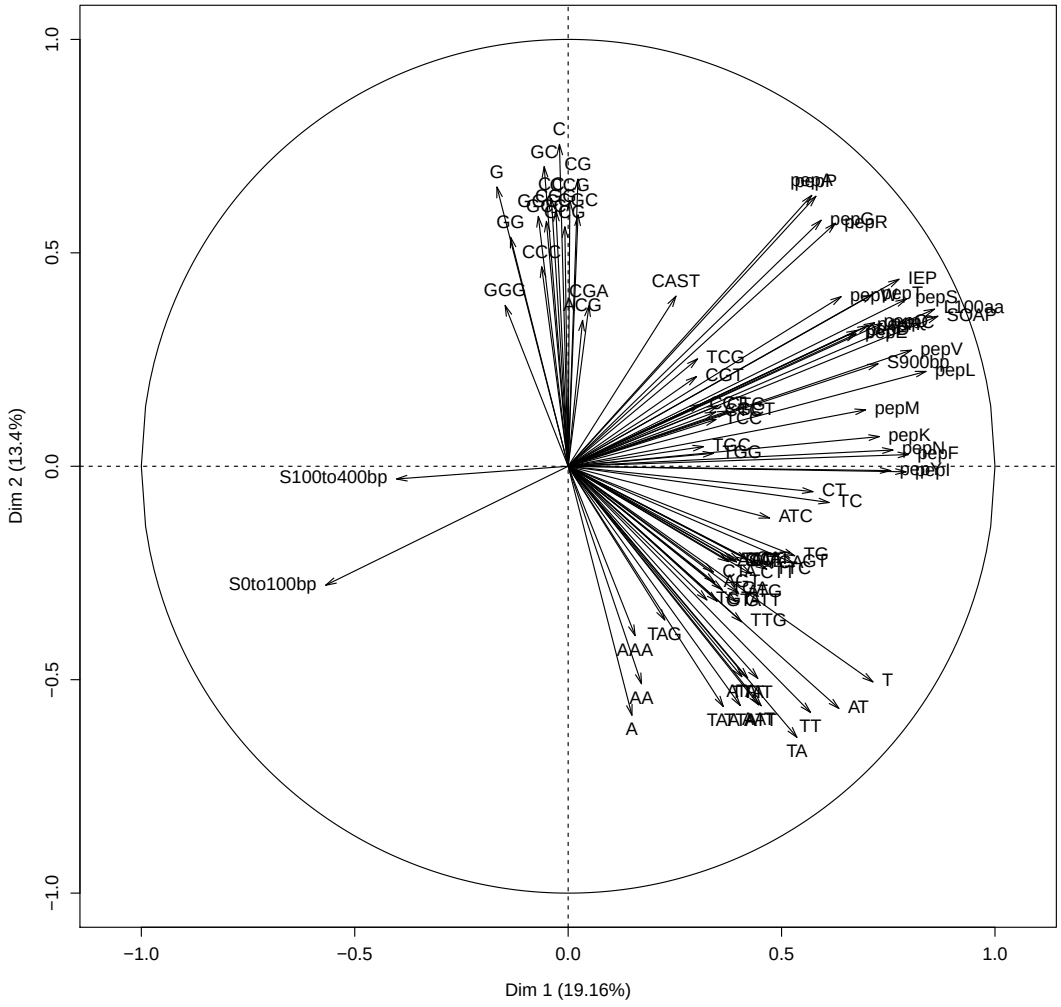


Figura 5.15: Análise de Componente Principal para as duas variáveis com maior contribuição de variância para o conjunto de treinamento $dbTr2.dat$.

O gráfico apresenta três grandes agrupamentos em que os vetores $\psi(q_i)$ concordam entre si, numa situação de alta correlação. Os atributos relacionados aos nucleotídeos G e C, formando o agrupamento A, os atributos relacionados aos nucleotídeos A e T, formando o agrupamento B, e os peptídeos e atributos relacionados a proteínas, formando o agrupamento C.

O agrupamento A, conforme esperado, revela a interação entre as frequências de nucleotídeos, dinucleotídeos e trinucleotídeos que contém majoritariamente as bases G ou C em sua composição. De acordo com o método de extração desse atributo, descrito na Subseção 4.1.1, esse é um resultado esperado. A complementaridade entre bases purinas e pirimidinas, alvo de exemplo do funcionamento da PCA na Subseção 3.1.4, também é resultado esperado para a análise, evidenciado pela correlação entre os agrupamentos A e B. É possível observar, no entanto, algumas variáveis relevantes que interagem de forma mais próxima com frequência de peptídeos, outras com projeção próxima de zero. O interesse dessas variáveis é alvo da análise da matriz de correlação R. Por ser muito

extensa, a matriz é mantida somente como Material Complementar.

O comportamento das variáveis referentes ao atributo 5, tamanho S de sequência, revela uma separação bastante forte entre sequências com menos de $400nt$ e sequências maiores do que $400nt$. Essa distinção é obtida principalmente pela adição de pequenos RNAs, em sua maioria ligados a atributos não codantes. Por isso, o distanciamento de sequências pequenas e sinais caracteristicamente codantes, no gráfico, é um resultado relevante para o entendimento do problema. Outro ponto importante a ser comentado é a ausência de variáveis que representam tamanho de ORF menor do que $100aa$. Esse resultado, condizendo com o fato de que as variáveis $L \leq 20aa$ e $20aa < L \leq 60aa$ não foram exercitadas no conjunto de treinamento, são seguidas da variável $60aa < L \leq 100aa$. Essa variável, muito visada em várias discussões sobre melhores atributos para identificação de ncRNAs (p. ex. nos trabalhos de Eddy, 2001; Liu et al., 2006), acabou sendo descartada por causa de sua baixa variabilidade no conjunto. Fazendo uma análise pontual, descobriu-se que ela atua de forma mais próxima ao agrupamento B, dando evidências de que a presença de pequenas ORFs pode sim contribuir para a identificação de ncRNAs.

O mapa ilustrado na Figura 5.16 mostra as variações das dimensões 1 e 2 tomadas como referência no conjunto de treinamento $dbTr2.dat$. É notável a separação do conjunto em três segmentos bem distintos, e a correspondência óbvia a que isso remonta é o resultado ótimo obtido com redes SOM e ART com classes de múltiplos de 3, em especial a rede SOM de topologia 3×2 retangular. Visualmente, a presença de poucos *outliers*, ou seja, exemplares muito afastados do centro estimado para cada agrupamento, contribui para o treinamento dessas redes não supervisionadas.

Finalmente, de forma condensada, os autovalores obtidos, ordenados de forma decrescente, são exibidos na Figura 5.17. A linha horizontal exibe a posição de autovalor unitário, considerada ponto de corte suficiente para a redução de variáveis. Considerando esse ponto de corte somente, os atributos são reduzidos de 117 variáveis para somente 24 variáveis mais expressivas. Contudo, levando-se em conta também as correlações existentes entre os atributos, de valia fundamental para o conjunto examinado, conforme indícios visuais explanados na Figura 5.15, o número de variáveis do modelo reduzido é fixado em 79.

A Tabela 5.21 relaciona as 79 variáveis numéricas extraídas de sequências no Passo 3 do método proposto. O valor Id refere-se ao identificador dado a cada atributo, conforme descrição na Subseção 4.1.1. O método de extração não foi alterado para nenhum atributo, somente quais variáveis numéricas são selecionadas. Por exemplo, para o atributo 3, frequência de trinucleotídeos, o cálculo não foi alterado, porém, somente os atributos listados constituem o vetor de características.

A redução final obtida pelo procedimento PCA alcançou, utilizando somente informação sobre autovalores, $79,5\%$ do montante inicial de 117 variáveis, enquanto que, incluindo informação sobre o grau de correlação entre as variáveis estudadas, essa redução dimensional atingiu $32,5\%$.

5.2.8 Redução de atributos da rede ART

Após a redução de atributos realizada pela PCA do conjunto $dbTr2.dat$, o passo seguinte consiste em encontrar o melhor número de *clusters* para o conjunto de dados. Por simpli-

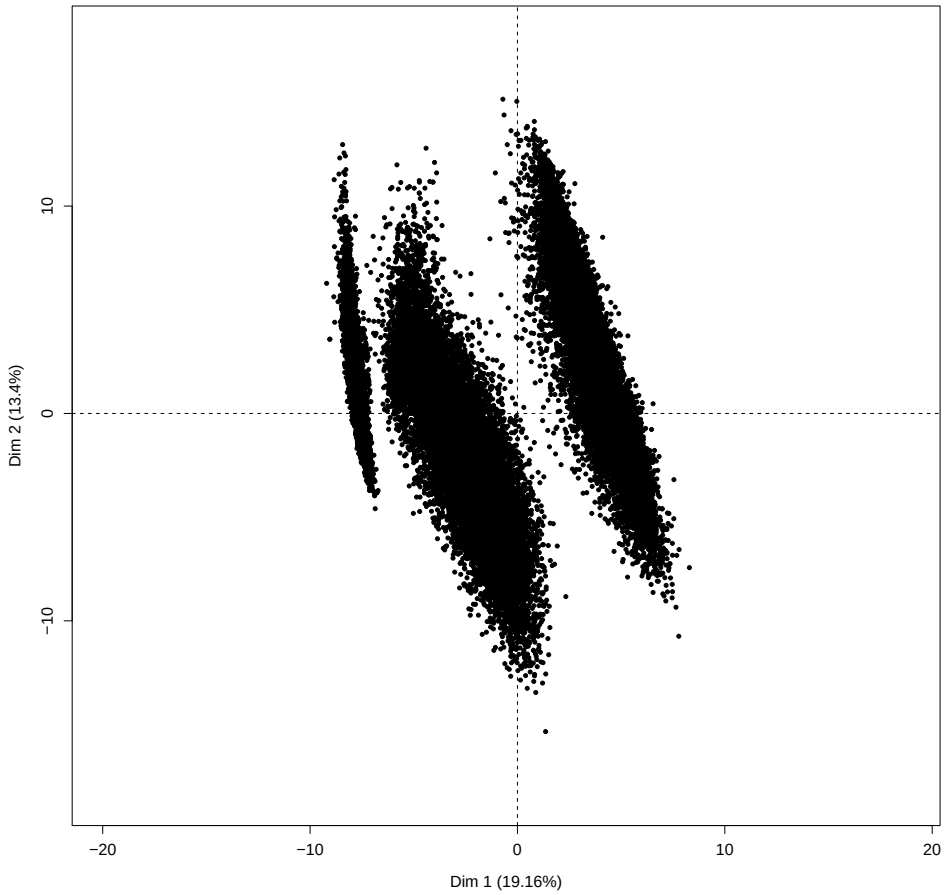


Figura 5.16: Mapa de variabilidade dos exemplares do conjunto $dbTr2.dat$ em relação às duas dimensões de maior contribuição de variância para o conjunto.

Tabela 5.21: Os 79 atributos numéricos mais significativos extraídos de cada sequência.

Id	Variáveis
1	todos os 4 nucleotídeos
2	todos os 16 dinucleotídeos
3	trinucleotídeos AAA, AAC, AAG, AAT, ACA, ACG, ACT, AGA, AGC, AGG, AGT, ATA, ATC, ATT, CAA, CAC, CAG, CAT, CCA, CCC, CCG, CCT, CGA, CGC, CGG, CGT, CTA, CTC, CTG, CTT, GAA, GAC, GAG, GAT, GCC, GCT, GGA, GGC, GGG, GTA, GTC, GTG, TAA, TAG, TAT, TCA, TCG, TGC, TGG, TTA, TTT
4	Aminoácidos C, D, N e O
5	$S \leq 900bp$; $S > 900bp$
6	$L \leq 100aa$; $L > 100aa$

cidade, nomeia-se o conjunto de treinamento com atributos reduzidos da mesma forma, $dbTr2.dat$. O gráfico de treinamento da rede ART com esse conjunto é representado pela Figura 5.18.

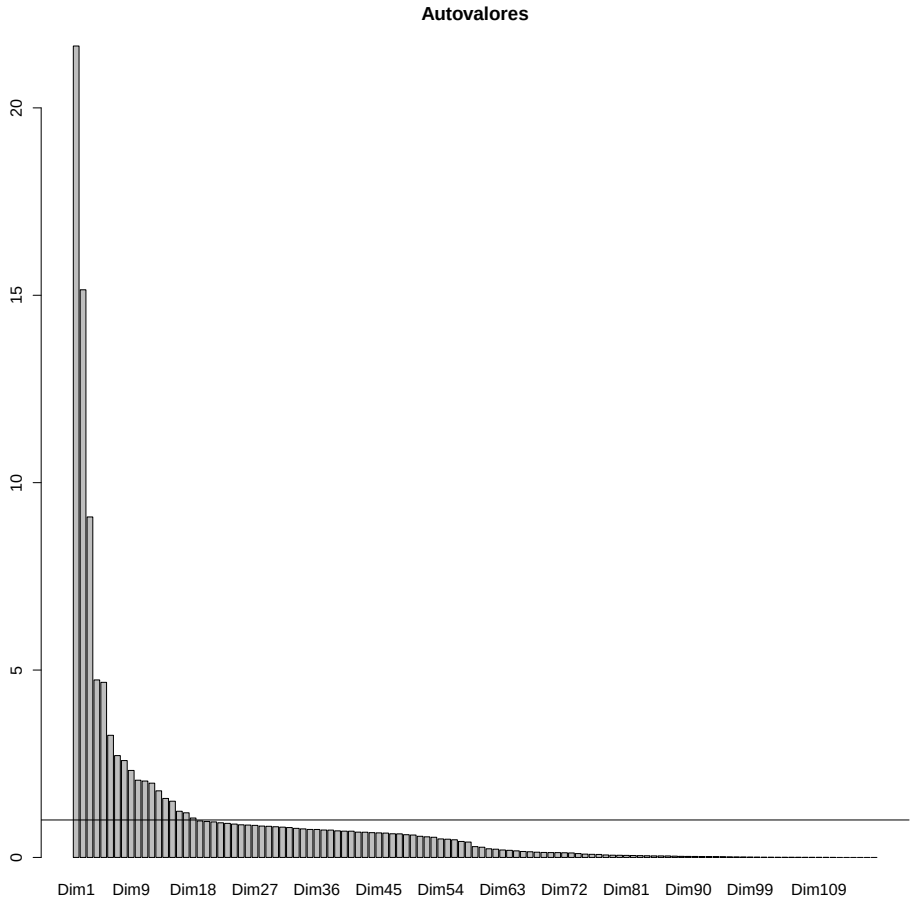


Figura 5.17: Autovalores das 117 dimensões analisadas em relação ao conjunto $dbTr2.dat$. A linha horizontal indica o ponto de autovalor $\lambda = 1$.

Como era esperado, a rede ART não conseguiu convergir em todos os valores de vigilância impostos. O resultado é esperado, dado que a análise não incluiu, propositalmente, nenhuma informação de correlação entre variáveis. Por causa dos resultados ruins apresentados no experimento com 24 variáveis, nenhum outro experimento foi conduzido com esse modelo. Os resultados para a redução no número de atributos para 79 variáveis numéricas é descrito pelo gráfico da Figura 5.19.

Observa-se a melhor convergência da rede para o valor de $\rho = 0,66$, obtendo um valor desprezível para a flutuação e. Nesse ponto, o algoritmo consolidou duas classes bastante distintas, como mostra a matriz de confusão da Tabela 5.22 construída para o conjunto $dbVal2.dat$ com variáveis reduzidas.

As medidas de performance para a rede ART de duas classes é exibida na Tabela 5.23. Os resultados mostram uma redução da especificidade em relação à precisão, proporcionada principalmente pela excisão dos atributos 7, 8 e 9, referentes a proteínas, e grande redução do atributo 4, referente à frequência de aminoácidos. Ressalta-se novamente, por esse resultado, o forte sinal codificante distinguível em conjuntos de treinamento para classificadores de ncRNAs não específicos. O valor de $qerror$ da rede, de $0,87$, sofreu redução, em relação ao modelo de 6 classes, um comportamento esperado, dado a redução

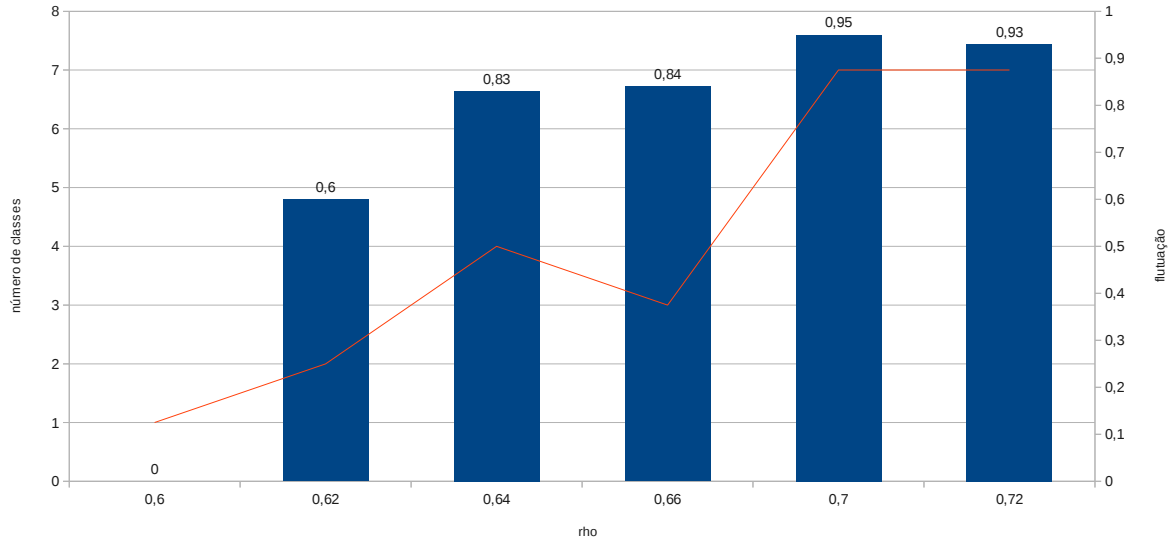


Figura 5.18: Resultados de acurácia do treinamento da rede ART utilizando 24 variáveis numéricas com melhor autovalor calculado pela PCA do conjunto $dbTr2.dat$.

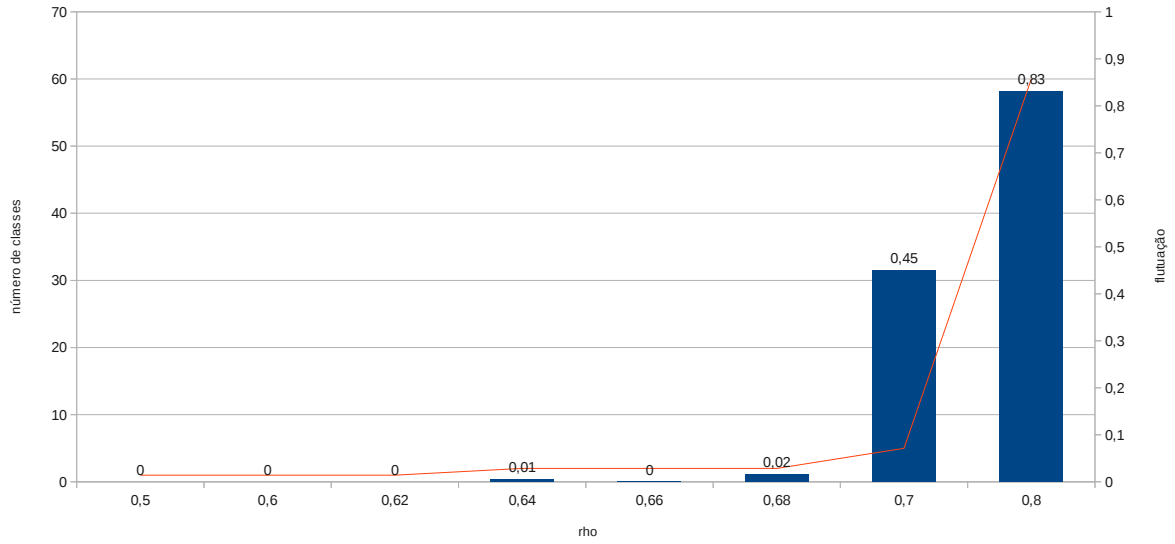


Figura 5.19: Resultados de acurácia do treinamento da rede ART utilizando 79 variáveis numéricas com melhor autovalor e fator de correlação calculados pela PCA do conjunto $dbTr2.dat$.

de variáveis numéricas e consequente redução dos possíveis valores assumidos por $qerror$.

Apesar da redução expressiva de variáveis numéricas, o método ART teve redução de acurácia, comparada ao modelo de 6 classes, de $0,088$, um resultado inicial bastante promissor. Com a adição de novas variáveis referentes a ncRNAs e redução gradual da importância do sinal codificante para classificação, pode-se construir um classificador robusto a partir de um conjunto majoritariamente não codificante.

Tabela 5.22: Matriz de confusão para a rede ART de 2 *clusters* usando 79 variáveis.

		Predito	
		Classe 1	Classe 2
Real	Codificante	19520	5480
	Não Codificante	149	19

Tabela 5.23: Medidas de performance P para a rede ART de 2 *clusters* usando 79 variáveis.

Precisão	*Recall*	Especificidade
$0,893$	$0,803$	$0,880$
Acurácia	**Medida F**	**MCC**
$0,837$	$0,846$	$0,618$

Tabela 5.24: Resultados do estudo de caso sobre ncRNAs de 4 organismos filogeneticamente distantes da rede ART com 2 classes usando 79 variáveis.

	H. sapiens	*E. coli*	*S. cerevisiae*	*A. thaliana*
Sequências não Codantes	133	140	400	415
Codantes	0	14	13	0

A Tabela 5.24 lista os resultados obtidos para o estudo de caso usando os ncRNAs conhecidos. Os resultados alcançam acurácia de $0,978$, mostrando excelente desempenho da rede com esses dados de teste. Entretanto, devido a resultados passados sobre a qualidade da predição, relacionada com o espaço de decisão criado, nova etapa supervisionada é executada para a rede ART de 2 classes.

Etapa supervisionada usando LVQ

A etapa supervisionada obteve valor de $qerror = 0,31$ para o conjunto $dbLvqOpt1.dat$ e $0,39$ para o conjunto $dbLvqOpt2.dat$. Os dados do estudo de caso condicionado ao treinamento supervisionado LVQ são exibidos na Tabela 5.25. Novamente, as primeiras linhas referem-se aos resultados do primeiro conjunto de treinamento, e as subsequentes linhas são o resultado do segundo conjunto.

As acurácias obtidas foram idênticas ao método ART de 2 classes. Esse resultado excelente demonstra que o espaço de decisão treinado pelo algoritmo está bastante consolidado, em termos de acurácia e $qerror$, para o conjunto de testes utilizado.

5.2.9 Redução de atributos da rede SOM

De posse dos ótimos resultados obtidos para a rede ART de 2 classes, o último passo dos experimentos desse trabalho refere-se à construção de uma rede SOM 2×1 equivalente. Antes de prosseguir ao treinamento da rede SOM 2×1, entretanto, o gráfico na Figura 5.20 de treinamento da rede SOM 3×1 utilizando somente 24 variáveis é exibido, para exemplificar uma situação de não convergência da rede SOM.

Tabela 5.25: Resultados do estudo de caso sobre ncRNAs de 4 organismos filogeneticamente distantes da rede ART com 2 classes usando 79 variáveis com treinamento supervisionado por LVQ.

	H. sapiens	*E. coli*	*S. cerevisiae*	*A. thaliana*
Sequências não Codantes	133	140	400	415
Codantes	0	14	13	0
Não Codantes	133	140	400	415
Codantes	0	14	13	0

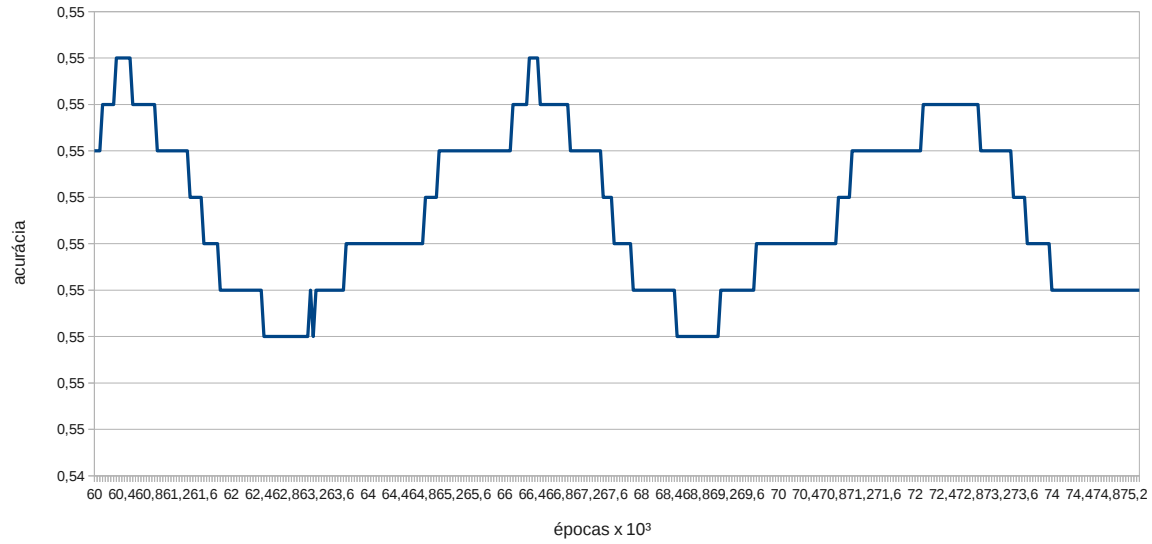


Figura 5.20: Resultados de acurácia do treinamento da rede SOM com topologia 3×1 retangular utilizando 24 variáveis numéricas com melhor autovalor calculado pela PCA do conjunto $dbTr2.dat$.

Como era esperado, dados os péssimos resultados de convergência obtidos pela rede ART discutidos na Subseção 5.2.8, a rede SOM não consegue convergir os valores dos estímulos recebidos em classes bem delimitadas. O resultado são baixos valores de acurácia, próximos de um classificador aleatório de acurácia 50%. Em contrapartida, os resultados para a rede SOM de topologia 2×1, conforme divisão ótima de classes encontrada para a rede ART, apresentaram bom desempenho, ilustrado na Figura 5.21.

A rede treinada teve acurácia estimada de $0,837$, com valor de época para a etapa de ordenação ajustado para 120.000 épocas, e para 1.200.000 épocas para a etapa de convergência. Para efeitos de estudo, outras duas redes SOM foram treinadas, com topologia 2×2 e 3×2, seguindo portanto o consenso estabelecido pelos experimentos anteriores realizados com o conjunto de treinamento $dbTr2.dat$ e todas as 117 variáveis extraídas. Os resultados desses experimentos não foram satisfatórios. Durante a validação do trei-

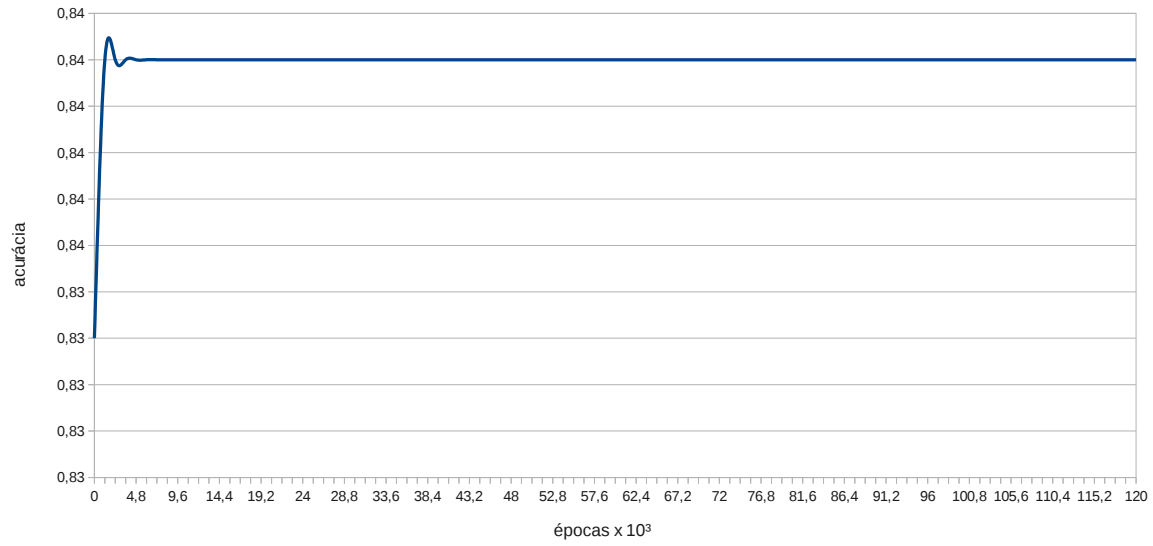


Figura 5.21: Resultados de acurácia do treinamento da rede SOM com topologia 2×1 retangular utilizando 79 variáveis numéricas com melhor autovalor e fator de correlação calculados pela PCA do conjunto $dbTr2.dat$.

namento, respectivamente, 2 e 4 classes das redes treinadas não foram exercitadas por quaisquer exemplares do conjunto. Essa presença de classes nulas indica um exaurimento da capacidade de discriminação de classes, para os parâmetros informados à rede, em relação ao conjunto de treinamento. Por um lado, é um resultado esperado, visto que a única configuração estável para a rede ART encontrada conseguiu distinguir somente duas classes nesse conjunto de treinamento.

As matrizes de confusão e medidas de performance obtidas para a rede SOM 2×1 foram idênticas aos valores obtidos para a rede ART com 79 variáveis e 2 classes. Já o valor de $qerror$ medido para a rede foi de $0,66$. A semelhança entre as redes SOM e ART treinadas é resultado da concentração dos sinais relevantes para identificação de ncRNAs em detrimento de sinais secundários. Essa redução auxilia na determinação exata de quais variáveis e atributos são realmente relevantes para o problema abordado pelo trabalho, porém deve ser acompanhada, futuramente, de inclusão de variáveis que permitam criar uma diversidade de classificação de ncRNAs, característica muito útil para anotação de ncRNAs em projetos.

A u-matriz do modelo SOM 2×1 é mostrada na Figura 5.22. Novamente, a separação entre as classes codificante e não codificante é bastante ressaltada, em comparação com a dispersão dos dois agrupamentos.

O estudo de caso da rede SOM com 79 variáveis retornou resultados idênticos aos resultados discutidos para a rede ART de 79 variáveis.

Etapa supervisionada usando LVQ

O procedimento LVQ foi conduzido de igual forma na rede SOM 2×1. O valor de $qerror$ para a rede é de $0,32$ para o conjunto $dbLvqOpt1.dat$ e de $0,39$ para $dbLvqOpt2.dat$. As u-

Figura 5.22: Representações por u-matriz da rede SOM 2×1 com variáveis reduzidas.

matrizes obtidas não diferiram da u-matriz para o mapa 2×1 apresentada na Figura 5.22. Os estudos de caso realizados, de igual forma, obtiveram resultado idêntico à rede ART de 2 classes.

Capítulo 6

Conclusão

Nesse capítulo, as principais atividades e resultados obtidos são recapitulados, para fins de ordenação temporal dos trabalhos realizados. Discutem-se também os resultados obtidos à luz dos objetivos definidos no Capítulo 1. Trabalhos e atividades futuras também são definidos, bem como o acesso e utilização do *software* desenvolvido no trabalho.

6.1 Sumário das Atividades e Resultados

Os experimentos realizados nesse trabalho, em ordem cronológica, são resumidos abaixo:

- Treinamento de rede SOM de topologia hexagonal 2×2, com função de vizinhança $hj(m)$ gaussiana, utilizando, na etapa de ordenação, $e = 2.340$ épocas, raio de vizinhança $V_m(0) = 2$ e taxa de aprendizado $\alpha(0) = 0,1$, e, na etapa de convergência, $e = 23.400$, $V_m(0) = 1$ e $\alpha(0) = 0,01$. A acurácia obtida foi de $0,858$;
- Treinamento de rede SOM de topologia retangular 2×1, $hj(m)$ gaussiana. Na etapa de ordenação, $e = 120.000$, $V_m(0) = 2$ e $\alpha(0) = 0,5$. Na etapa de convergência, $e = 1.200.000$, $V_m(0) = 1$ e $\alpha(0) = 0,05$. A acurácia obtida na validação do treinamento foi de $0,900$;
- Treinamento de rede SOM de topologia retangular 3×1, $hj(m)$ gaussiana. Na etapa de ordenação, $e = 360.000$, $V_m(0) = 3$ e $\alpha(0) = 0,5$. Na etapa de convergência, $e = 3.600.000$, $V_m(0) = 1$ e $\alpha(0) = 0,05$. O melhor resultado da rede obteve acurácia no estudo de caso de $0,971$, sem utilizar etapa supervisionada LVQ;
- Treinamento de rede SOM de topologia retangular 3×2, $hj(m)$ gaussiana. Na etapa de ordenação, $e = 720.000$, $V_m(0) = 3$ e $\alpha(0) = 0,5$. Na etapa de convergência, $e = 7.200.000$, $V_m(0) = 1$ e $\alpha(0) = 0,05$. O melhor resultado da rede obteve acurácia no estudo de caso de $0,961$, utilizando etapa supervisionada LVQ com 7.200 sequências de treinamento, $\alpha(0) = 0,1$ e $e = 360.000$;
- Treinamento de rede ART com 6 classes, utilizando taxa de aprendizado $\eta = 0,25$, taxa de pesquisa por protótipos $\alpha = 0,01$, fator de vigilância $\rho = 0,74$, número máximo de épocas $n = 40$ e valor máximo de flutuação $e = 0,05$. A acurácia obtida no estudo de caso foi de $0,958$, sem utilizar treinamento supervisionado LVQ;

- PCA sobre o conjunto de 117 atributos numéricos extraídos das sequências de RNA. O Estudo resultou na redução para 79 variáveis numéricas. O critério de seleção para boas variáveis escolhido foi seu *eigenvalor* e seu fator de correlação com outras variáveis, medido através da matriz de correlação R;
- Treinamento de rede ART com 79 variáveis numéricas, utilizando $\eta = 0,25$, $\alpha = 0,01$, $\rho = 0,66$, número máximo de épocas $n = 40$ e valor máximo de flutuação $e = 0,05$. A acurácia obtida no estudo de caso foi de $0,978$, utilizando ou não a etapa supervisionada LVQ, aplicada de forma semelhante ao quarto item;
- Treinamento de rede SOM com 79 variáveis numéricas, usando topologia retangular 2×1, com função de vizinhança $hj(m)$ gaussiana e, na etapa de ordenação, $e = 120.000$ épocas, raio de vizinhança $V_m(0) = 2$ e taxa de aprendizado $\alpha(0) = 0,1$, e, para a etapa de convergência, $e = 1.200.000$, $V_m(0) = 1$ e $\alpha(0) = 0,01$. A acurácia obtida foi de $0,978$, utilizando ou não a etapa supervisionada LVQ, aplicada de forma idêntica ao sétimo item.

Dos resultados resumidos acima, frisa-se a relevância das redes SOM 3×2 utilizando o procedimento supervisionado LVQ, cuja criação foi condicionada pela análise da rede ART de 6 classes correspondente. Os ótimos desempenhos de ambas as redes demonstram que a utilização desse procedimento para inferência do número ótimo de classes em conjuntos de treinamento bastante heterogêneos é eficaz.

Em geral, os resultados evoluiram consideravelmente, desde os estudos do Primeiro Experimento, realizado sem o grande auxílio das análises da rede ART, do algoritmo LVQ e do procedimento PCA. Também é importante ressaltar a influência decisiva do novo conjunto de treinamento, fabricado através de métricas e direcionamentos mais criteriosos, para alcançar os objetivos propostos.

Os resultados para redução do número de variáveis mostra, de forma inédita, a capacidade de aplicar o procedimento PCA para avaliar, de forma objetiva e precisa, o grau de contribuição e relevância de cada atributo numérico e informação biológica extraída de sequências de RNA para o problema de identificação e classificação de ncRNA. Essa é uma importante contribuição, visto que não existe um consenso formado sobre quais são as melhores métricas para solucionar esse problema.

A utilização de etapa supervisionada para refinar o espaço de decisão das redes neurais treinadas constrói um entendimento também muito útil para o problema de nomeação das classes encontradas. Pode-se incluir informações específicas de determinados tipos ou famílias de ncRNAs, incluindo até mesmo informações pontuais sobre estruturas secundárias complexas, que de outra forma demandariam vultosos recursos computacionais para serem identificadas. Essa observação é especialmente pertinente para pequenas ncRNAs bem caracterizados, e tem por exemplo a ferramenta DARIO (Fasold et al., 2011), que utiliza florestas de decisão para identificar e classificar pequenos ncRNAs, especialmente miRNAs.

A metodologia para treinamento e validação dos dados é também contribuição do presente trabalho. A construção de um *pipeline* de anotação para ncRNAs envolve sempre a utilização de mais de uma ferramenta de análise. Nesse trabalho, explorou-se a utilização de uma variedade desses *software*, desde métodos por AM, passando por algoritmos consagrados, como o BLAST (Altschul et al., 1997), até algoritmos especializados, como o Infernal (Nawrocki et al., 2009). A ferramenta proposta e aperfeiçoada nesse trabalho

pode ser facilmente integrada a *pipelines* existentes, por exemplo no sistema multiagente BioAgents (Ralha et al., 2008), criado na UnB para anotação automatizada de sequências codantes ou não codantes.

À luz dos objetivos propostos no Capítulo 1, o presente trabalho conclui:

- A construção de um novo conjunto de treinamento;
- A proposta do método SOM-Portrait para identificar ncRNAs;
- A utilização bem sucedida de etapa supervisionada, possibilitando a especialização de classes da rede SOM para classificar grupos específicos de ncRNAs baseados em informação estrutural ou sinal codante existente;
- A proposta do método ART-Portrait para escolha ótima do número de classes existente no conjunto de treinamento;
- A utilização de atributos relevantes para a identificação de ncRNAs pelas redes SOM e ART;
- Estudos de caso bem sucedidos utilizando diversos organismos, filogeneticamente próximos ou não, mostrando ótimo grau de generalização das redes treinadas.

6.2 Trabalhos Futuros

Várias atividades de validação ainda podem ser desenvolvidas, para elucidar questões levantadas durante os experimentos:

- Utilizar a informação de *qerror* para cada classe, de forma a retornar um valor de qualidade associado à predição realizada pela rede. Essa melhoria permite também avaliar o desempenho de classes intermediárias no espaço de decisão;
- Inserir novos atributos indicativos de ncRNAs, para reforçar outros tipos de sinais. Pode-se citar, como exemplo, concentrações de ilhas *CpG*, *z-Score* associado à energia livre de estrutura secundária do RNA (Hofacker et al., 2002), escore de similaridade utilizando o *software* Infernal, entre vários outros;
- Aprimorar o conjunto de treinamento, incluindo mais informações sobre longos ncRNAs não codantes e ncRNAs sem estrutura secundária bem caracterizada;
- Construir o portal para utilização das ferramentas SOM-Portrait e ART-Portrait;
- Integrar as ferramentas ao sistema BioAgents.

6.3 Acesso e *Download*

Atualmente a ferramenta está disponível somente através de contato com o autor. Um pacote compactado com a ferramenta e instruções de compilação e execução é fornecido como Material Complementar. Em breve, uma versão *web* será disponibilizada.

Anexo I

Parâmetros de entrada das bibliotecas e programas utilizados

O presente anexo serve como consulta mais detalhada aos parâmetros de entrada das bibliotecas e ferramentas utilizadas pelo método SOM-Portrait proposto.

I.1 Biblioteca SOM_PAK (Kohonen et al., 1996b)

```
$ ./randinit -help
mapinit/randinit/lininit - initializes the codebook vectors for
SOM learning
Initialization type is determined from program name (randinit
or lininit) or is selected with the -init option.

Required parameters:
  -din filename         input data
  -cout filename        output codebook filename
  -topol type           topology type of map, hexa or rect
  -neigh type           neighborhood type, bubble or gaussian
  -xdim integer
  -ydim integer         dimensions of the map
Optional parameters:
  -init type            initialization type, rand or lin. Overrides
                        the type determined from the program name
  -rand integer         seed for random number generator. 0 is current
   time
  -buffer integer       buffered reading of data, integer lines at a
   time



$ ./vsom -help
vsom - teach self-organizing map
Required parameters:
  -cin filename         initial codebook file
```

```
  -din filename         teaching data
  -cout filename        output codebook filename
  -rlen integer         running length of teaching
  -alpha float          initial alpha value
  -radius float         initial radius of neighborhood
Optional parameters:
  -rand integer         seed for random number generator. 0 is current
   time
  -fixed                use fixed points
  -weights              use weights
  -buffer integer       buffered reading of data, integer lines at a
   time
  -alpha_type type      type of alpha decrease, linear (def) or
   inverse_t.
  -snapfile filename    snapshot filename
  -snapinterval integer interval between snapshots


$ ./vcal -help
vcal - sets the labels of entries by the majority voting
Required parameters:
  -cin filename         codebook file
  -din filename         labeling data
  -cout filename        labeled output codebook filename
Optional parameters:
  -numlabs integer      maximum number of labels to assign to a codebook
                        vector. Default is 1, 0 gives all.
  -buffer integer       buffered reading of data, integer lines at a time


$ ./qerror -help
qerror - calculate quantization error for the data entries
Required parameters:
  -cin filename         codebook file
  -din filename         test data
Optional parameters:
  -qetype 1             another way to calculate the error
  -radius float         radius of neighborhood for alternative above
  -buffer integer       buffered reading of data, integer lines at a time


$ ./umat -help
umat - produce EPS/PS picture of a SOM
Required options:
  -cin              input codebook file
```

```
Optional parameters:
  -o filename       output filename (default is stdout)
  -eps              output EPS file (the default)
  -ps               output PS file
  -portrait         output PS picture in portrait mode
  -landscape        output PS picture in landscape mode
  -border           draw border around map units
  -onlylabs         draw only labels
  -nolabs           don't draw labels
  -W float          white treshold
  -B float          black treshold
  -title string     set title (default is input codebook name)
  -notitle          do not print title line on PS picture
  -font fontname    PS fontname for labels
  -fontsize float   fontsize relative to the radius of an unit
  -paper type       select paper size for PS output, A4 (default) or A3
  -average          average the umatrix
  -median           median filter the umatrix
  -headerfile fname specify alternative postscript header file
```

```
$ ./visual -help
visual - find best matching unit for each data sample
Required parameters:
  -cin filename         codebook file
  -din filename         input data
  -dout filename        output filename
Optional parameters:
  -noskip               do not skip data vectors that have all components
                        masked off
  -buffer integer       buffered reading of data, integer lines at a time
```

I.2 Biblioteca ART distance (Hudik and Zizka, 2011)

```
$ ./art_distance
ART distance (Adaptive Resonance Theory) -- clustering algorithm.
Copyright Tomas Hudik, Jan Zizka
Contact: xhudik@fi.muni.cz


usage: art_distance [-o -s -b -v -a -e -E] -i input_file

The options mean:
-i (a must!) input_file
-o prefix for output files
```

```
-b beta (learning rate, [0,1]). Default 0.5
-v vigilance ( 0<= vigilance <=1 )
-a alpha ( alpha <= #columns^-0.5)
-d distance measure:
1 euclidean: 1 - sqrt( Sum((x-y)^2)/#dimen )
2 modified euclidean: log(#dimen^2)
- sqrt( Sum((x-y)^2) ) NOT WORKING YET
3 manhattan: 1 - Sum(|x - y|)/#dimen
4 correlation distance: 0.5 +
(x-Xmean)*(y-Ymean)'/(2*sqrt((x-Xmean)(x-Xmean)' *
sqrt((y-Ymean)(y-Ymean)')))
5 minkowski distance: 1 - (Sum((|x - y|)^p)/#dimen)^(1/p);
set up -p
6 mahalanobis: 1 - sqrt( (x-mean) * C^-1 * (x-mean)')
NOT WORKING YET
-p x (x is positive integer); it is a power for the minkowski
distance
-s x (x is positive integer) - skip last x columns
-e fluctuation: % of examples which are re-assigned
-E integer: maximum number of passes through the input examples


art distance is based on:
http://www.fi.muni.cz/~xhudik/art

Frank, Kraiss, Kuhlen,"Comparative analysis of Fuzzy ART and
ART-2A network clustering performance" and  Recognition",
Neural Networks, vol. 9, pp. 544--559, 1998
```

I.3 Biblioteca LVQ_PAK (Kohonen et al., 1996a)

```
$ ./olvq1 -help
lvqtrain/lvq1/lvq2/lvq3/olvq1 - teach codebook with one of the
lvq algorithms
 Training algorithm is determined from program name and can be
 overridden
 with the -type option.
Required parameters:
  -cin filename         initial codebook file
  -din filename         teaching data
  -cout filename        output codebook filename
  -rlen integer         running length of teaching
  -alpha float          initial alpha value (optional with olvq1)
  -win float            (lvq2, lvq3) window width
  -epsilon float        (lvq3) training epsilon
```

```
Optional parameters:
  -type lvqtype         select which lvq algoritm to use: lvq1, lvq2,
                        lvq3 or olvq1
  -rand integer         seed for random number generator. 0 is current
   time
  -buffer integer       buffered reading of data, integer lines at a
   time
  -alpha_type type      type of alpha decrease, linear (def) or
   inverse_t.
  -snapfile filename    snapshot filename
  -snapinterval integer interval between snapshots
```

I.4 Rotinas e *scripts* da biblioteca FactoMineR (Lê et al., 2008)

A rotina R descrita nessa Seção deve ser invocada no ambiente gráfico *Rcmdr* **após** carregamento das seguintes bibliotecas:

```
library(FactoMineR)
library(Rcmdr)
library(fields)
```

Dentro do ambiente gráfico *Rcmdr*, a rotina a seguir deve ser copiada, ou simplemente carregada do arquivo *PCAscript.R* disponibilizada como Material Complementar.

```
PCA <- read.table("dbTr.dat", header=TRUE, sep="",
  na.strings="NA", dec=".", strip.white=TRUE)
PCA.PCA<-PCA[, c("A", "C", "G", "T", "AA", "AC", "AG", "AT",
+ "CA", "CC", "CG", "CT", "GA", "GC", "GG", "GT", "TA", "TC",
+ "TG", "TT", "AAA", "AAC", "AAG", "AAT", "ACA", "ACC", "ACG",
+ "ACT", "AGA", "AGC", "AGG", "AGT", "ATA", "ATC", "ATG", "ATT",
+ "CAA", "CAC", "CAG", "CAT", "CCA", "CCC", "CCG", "CCT", "CGA",
+ "CGC", "CGG", "CGT", "CTA", "CTC", "CTG", "CTT", "GAA", "GAC",
+ "GAG", "GAT", "GCA", "GCC", "GCG", "GCT", "GGA", "GGC", "GGG",
+ "GGT", "GTA", "GTC", "GTG", "GTT", "TAA", "TAC", "TAG", "TAT",
+ "TCA", "TCC", "TCG", "TCT", "TGA", "TGC", "TGG", "TGT", "TTA",
+ "TTC", "TTG", "TTT", "S900bp", "S400to900bp", "S100to400bp",
+ "S0to100bp", "L100aa", "L60to100aa", "L20to60aa", "L0to20aa",
+ "pepA", "pepC", "pepD", "pepE", "pepF", "pepG", "pepH", "pepI",
+ "pepK", "pepL", "pepM", "pepN", "pepO", "pepP", "pepQ", "pepR",
+ "pepS", "pepT", "pepU", "pepV", "pepW", "pepY", "SOAP", "IEP",
+ "CAST")]
res<-PCA(PCA.PCA , scale.unit=TRUE, ncp=117, graph = FALSE)
plot.PCA(res1, axes=c(1, 2), choix="ind", new.plot=TRUE,
+ habillage="none", col.ind="black", col.ind.sup="blue",
+ col.quali="magenta", label=c("ind.sup","quali"), title="")
```

```
plot.PCA(res, axes=c(1, 2), choix="var", new.plot=TRUE,
+ col.var="black", col.quanti.sup="blue", label=c("var",
+ "quanti.sup"), lim.cos2.var=0.1, + title="")
write.infile(res$eig, file ="PCA1_e_2.csv",append=FALSE)
write.infile(res$var, file ="PCA1_e_2.csv",append=TRUE)
write.infile(dimdesc(res, axes=c(1, 2)), file ="PCA1_e_2.csv",
+ append=TRUE)
remove(PCA.PCA)
dev.print(pdf, file="PCA1_e_2.pdf", width=10.0, height=10.0,
  pointsize=10)
dev.print(pdf, file="Individuals1_e_2.pdf", width=10.0,
  height=10.0, pointsize=10)
barplot(res$eig[,1], main = "Eigenvalues",names.arg =
+ paste("Dim", 1:nrow(res$eig), sep = ""))
yline(1)
```

I.5 Parâmetros de entrada e opções da ferramenta ANGLE (Shimizu et al., 2006)

```
$ ./ANGLE-linux64DP --help
Usage: ANGLE.exe [OPTION]

OPTIONS:
--help: Show this help.
-i: Input filename (default name is sample.txt.)
-d: Paramater directory (default directory is
./param-human)
-w: Display width (default width is 60 bases.)
-h: Output frame layout to a html file.
-s: Output score of each base to a file.

----
ex) ANGLE.exe -w 70 -h out.html -s score.cvs -i
inputFasta.txt

REFERENCE:
Shimizu, K., Adachi, J., and Muraoka, Y. (2006) ANGLE:
a sequencing errors resistant program for predicting
protein coding regions in unfinished cDNA.
J Bioinform Comput Biol 4, 649-664
```

O método SOM-Portrait utiliza os valores padrão da ferramenta.

I.6 Parâmetros de entrada e opções da ferramenta CAST (Promponas et al., 2000)

```
Usage: ./cast SequenceFile [options]

 -help    ... print this text
 -thr t   ... set the threshold score for reported
 regions
              default is 40
              t should be an integer number
 -stat    ... outputs statistics information to file cast.stat
 -matrix  ... use different mutation matrix (.mat) file
 -verbose ... verbose mode prints filtering information to
 standard output
 -stderr  ... verbose mode prints filtering information to
 standard error
```

O método SOM-Portrait utiliza os valores padrão da ferramenta.

Anexo II

Configuração do ambiente de trabalho

O objetivo desse anexo é detalhar os procedimentos para instação e execução do método numa máquina com configurações semelhantes à máquina descrita na Seção 4.4, ressaltando principalmente a instalação e compilação das bibliotecas e programas auxiliares utilizados.

II.1 Instalação de ferramentas auxiliares

A ferramenta CAST está disponível no arquivo compactado, na pasta *externalLibs*. O binário foi compilado para máquina 32*bits*. Até a data de confecção desse documento, não era possível descarregar os arquivos fonte para compilação correta. Sendo assim, para executar em ambiente 64*bits*, a biblioteca de compatibilidade *compat-libstdc++-33* deve ser instalada.

Utilize o método preferido para instalar a biblioteca.

```
$ yum install compat-libstdc++-33
Transação realizada com:
    Atualizados    rpm-4.9.1.2-1.fc16.x86_64
    Instalados     yum-3.4.3-7.fc16.noarch
    Instalados     yum-metadata-parser-1.1.4-5.fc16.x86_64
Pacotes alterados:
    Instalar compat-libstdc++-33-3.2.3-68.1.x86_64
```

A ferramenta ANGLE possui versões de 32*bits* e 64*bits*. Os autores do *software* não disponibilizaram o código fonte, porém retornaram os binários compilados com opção de ligação dinâmica de bibliotecas. Entretanto, ainda é necessário incluir uma biblioteca essencial C/C++, *libstdc++.so.5*.

Utilize o método preferido para instalar a biblioteca.

```
$ yum install libstdc++.so.5
Transação realizada com:
    Atualizados    rpm-4.9.1.2-1.fc16.x86_64
```

```
    Instalados     yum-3.4.3-7.fc16.noarch
    Instalados     yum-metadata-parser-1.1.4-5.fc16.x86_64
Pacotes alterados:
    Instalar    compat-libstdc++-33-3.2.3-68.1.i686
    Dep-Install libgcc-4.6.2-1.fc16.i686
```

II.2 Instalação das bibliotecas utilizadas

A biblioteca SOM_PAK não possui configurações detalhadas para instalação. Basta executar o comando *make*, conforme instruções no arquivo *README* fornecido. De forma similar, a biblioteca ART é compilada com o comando *make art*.

A biblioteca LVQ_PAK contém uma declaração conflitante da função *getline* com a biblioteca de sistema *stdio.h*. A correção é feita alterando a assinatura da função declarada no arquivo *fileio.h*, linha 69:

```
char *getline(struct file_info *fi);
```

Para:

```
char *get_line(struct file_info *fi);
```

E corrigir as referências a essa função adequadamente nos módulos *fileio.c*, linha 282, *datafile.c*, linha 125, 160 e 611.

Para a biblioteca dos métodos SOM e ART-Portrait, nomeada "SP", a versão de *perl* utilizada deve ser:

```
$ perl --version

This is perl 5, version 14, subversion 2 (v5.14.2)
```

A biblioteca BioPerl (Stajich et al., 2002) versão *1.6.901* deve ser instalada, bem como o suporte à função *Switch* para Perl, pela instalação do módulo *Switch.pm*. Utiliza-se o aplicativo *mpan* para instalação diretamente do repositório *CRAN*.

```
$ perl -MCPAN -e shell
cpan>install Bundle::CPAN
cpan>q

$ cpan
cpan>install Module::Build
cpan>o conf prefer_installer MB
cpan>o conf commit
cpan>q

$ yum install expat

$ cpan
```

```
cpan> d /bioperl/
cpan> force install CJFIELDS/BioPerl-1.6.901.tar.gz
cpan>q

$ yum install perl-Switch.noarch
Transação realizada com:
    Atualizados    rpm-4.9.1.2-1.fc16.x86_64
    Instalados     yum-3.4.3-7.fc16.noarch
    Instalados     yum-metadata-parser-1.1.4-5.fc16.x86_64
Pacotes alterados:
    Instalar perl-Switch-2.16-1.fc16.noarch
```

A instalação da biblioteca BioPerl requererá a instalação de vários módulos auxiliares. Recomenda-se a instalação utilizando opção *[all]*.

Referências

Saccharomyces Genome Database. `http://downloads.yeastgenome.org/`, dezembro 2011. 53

BioPerl Wiki Main Page. `http://www.bioperl.org/wiki/Main_Page`, fevereiro 2012. 46

UniProt. `http://www.uniprot.org/`, fevereiro 2012. 49, 67

S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, 25(17):3389–3402, 1997. 2, 51, 58, 67, 96

R. V. Andrade. *Análise do Transcriptoma e da Expressão Diferencial de Genes de Micélio e Levedura de* Paracoccidioides brasiliensis. PhD thesis, Universidade de Brasília, junho 2006. 52

R. Arrial, R. Togawa, and M. Brigido. Screening non-coding RNAs in transcriptomes from neglected species using PORTRAIT: case study of the pathogenic fungus Paracoccidioides brasiliensis. *BMC Bioinformatics*, 10(1):239, 2009. 2, 4, 37, 67, 74, 79

R. T. Arrial. Predição de RNAs não-codificadores no transcriptoma do fungo *Paracoccidioides brasiliensis* usando aprendizagem de máquina. Master's thesis, Universidade de Brasília, 2008. in portuguese. 3, 17, 49

R. Backofen, S. H. Bernhart, C. Flamm, C. Fried, G. Fritzsch, J. Hackermüller, J. Hertel, I. L. Hofacker, K. Missal, A. Mosig, S. J. Prohaska, D. Rose, P. F. Stadler, A. Tanzer, S. Washietl, and S. Will. RNAs everywhere: genome-wide annotation of structured RNAs. *Journal of Experimental Zoology Part B: Molecular and Developmental Evolution*, 308B(1):1–25, 2007. 18, 38

D. A. Benson, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and D. L. Wheeler. GenBank. *Nucleic Acids Research*, 33(suppl 1):D34–D38, 2005. 48

H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28(1):235–242, 2000. 15

C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995. 21, 23, 39

A. Bleasby. IEP. `http://emboss.bioinformatics.nl/cgi-bin/emboss/help/iep`, 1999. xiii, 45

Boaz and Shaanan. Structure of human oxyhaemoglobin at 2·1resolution. *Journal of Molecular Biology*, 171(1):31–59, 1983. x, 10

B. Boeckmann, A. Bairoch, R. Apweiler, M. Blatter, A. Estreicher, E. Gasteiger, M. J. Martin, K. Michoud, C. O'Donovan, I. Phan, S. Pilbout, and M. Schneider. The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Res.*, 31(1):365–370, outubro 2002. 1, 15, 48, 67

M. Brígido. Laboratório de Biologia Molecular - UnB. `http://vsites.unb.br/ib/cel/biomol/`, jan 2012. 4

G. A. Carpenter and S. Grossberg. A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine. *Computar Vision, Graphics, and Image Processing*, 37:54–115, 1987. xvi, 4, 34, 36

C. Chang and C. Lin. *LIBSVM: a library for Support Vector Machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`. 41

P. Clote and R. Backofen. *Computational Molecular Biology: An Introduction.* John Wiley & Sons Ltd, Chichester, England, 2000. 7, 9, 11, 12

G. Cochrane, R. Akhtar, J. Bonfield, L. Bower, F. Demiralp, N. Faruque, R. Gibson, G. Hoad, T. Hubbard, C. Hunter, M. Jang, S. Juhos, R. Leinonen, S. Leonard, Q. Lin, R. Lopez, D. Lorenc, H. McWilliam, G. Mukherjee, S. Plaister, R. Radhakrishnan, S. Robinson, S. Sobhany, P. T. Hoopen, R. Vaughan, V. Zalunin, and E. Birney. Petabytescale innovations at the European Nucleotide Archive. *Nucleic Acids Res.*, pages 1–7, 2008. 15, 49

J. H. R. D. Correia and A. A. D. Correia. Funcionalidades dos RNA não codificantes (nc-RNA) e pequenos RNA reguladores nos mamíferos. *Revista Eletrônica de Veterinária*, 8(10):1–22, 2007. 2

K. Crammer, Y. Singer, N. Cristianini, J. Shawe-Taylor, and B. Williamson. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001. 66

M. E. Dinger, K. C. Pang, T. R. Mercer, and J. S. Mattick. Differentiating Protein-Coding and Noncoding RNA: Challenges and Ambiguities. *PLoS Computational Biology*, 4(11), November 2008. 3, 16, 17, 22, 42, 43

Projeto DOGAN. *Aspergillus oryzae* RIB40. `http://www.bio.nite.go.jp/dogan/project/view/AO`, 2012. 52

S. R. Eddy. Non-coding RNA genes and the modern RNA world. *Nature Reviews Genetics*, 2:919–929, 2001. xiii, 1, 9, 12, 13, 14, 15, 16, 22, 53, 87

M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *PNAS*, 95(25):14863–14868, 1998. 2, 35

M. Fasold, D. Langenberger, H. Binder, P. F. Stadler, and S. Hoffmann. DARIO: a ncRNA detection and analysis tool for next-generation sequencing experiments. *Nucleic Acids Research*, 39(suppl 2):W112–W117, 2011. 3, 96

M. S. S. Felipe and M. M. Brígido. Projeto Genoma *Pb*. `https://www.biomol.unb.br/Pb/`, 2009. 52

National Center for Biotechnology Information. Fasta format description. `http://www.ncbi.nlm.nih.gov/BLAST/fasta.shtml`, 2011. 40

T. Frank, K. Kraiss, and T. Kuhlen. Comparative Analysis of Fuzzy ART and ART-2A Network Clustering Performance. *IEEE Transactions on Neural Networks*, 9(3): 544–559, 1998. x, 34, 36, 59

M. C. Frith, T. L. Bailey, T. Kasukawa, F. Mignone, S. K. Kummerfeld, M. Madera, S. Sunkara, M. Furuno, C. J. Bult, J. Quackenbush, C. Kai1, J. Kawai1, P. Carninci1, Y. Hayashizaki1, G. Pesole, and J. S. Mattick. Discrimination of non-protein-coding transcripts from protein-coding mRNA. *RNA Biology*, 3(1):40–48, 2006. 14, 16, 22

P. P. Gardner, J. Daub, J. G. Tate, E. P. Nawrocki, D. L. Kolbe, S. Lindgreen, A. C. Wilkinson, R. D. Finn, S. Griffiths-Jones, S. R. Eddy, and A. Bateman. Rfam: updates to the RNA families database. *Nucleic Acids Research.*, 37:D136–D140, 2009. x, xi, xiii, 13, 16, 49, 51, 54, 67

E. Gibb, C. Brown, and W. Lam. The functional role of long non-coding RNA in human carcinomas. *Molecular Cancer*, 10(1):38, 2011. 2, 14, 81

A. R. Gruber, R. Neuböck, I. L. Hofacker, and S. Washietl. The RNAz web server: prediction of thermodynamically stable and evolutionarily conserved RNA structures. *Nucleic Acids Research*, 35(suppl 2):W335–W338, 2007. xi, 54

A. M. Gustafson, E. Allen, S. Givan, D. Smith, J. C. Carrington, and K. D. Kasschau. ASRP: the Arabidopsis Small RNA Project Database. *Nucleic Acids Research*, 33(suppl 1):D637–D640, 2005. 53

S. Haykin. *Neural Network - a comprehensive foundation.* Prentice Hall, New Jersey, USA, 1999. x, 3, 19, 26, 27, 29, 30, 31, 33, 34, 35, 61

M. B. Hoagland, M. L. Stephenson, J. F. Scott, L. I. Hecht, and P. C. Zamecnik. A soluble ribonucleic acid intermediate in protein synthesis. *J. Biol. Chem.*, 231(1):241–257, 1958. 1

I. L. Hofacker, M. Fekete, and P. F. Stadler. Secondary Structure Prediction for Aligned RNA Sequences. *Journal of Molecular Biology*, 319(5), 2002. 2, 97

J. Hollmen. U-matrix. `http://www.cis.hut.fi/~jhollmen/dippa/node24.html`, 2009. 23, 25

C. Hsu, C. Chang, and C. Lin. *A Practical Guide to Support Vector Classification.* Department of Computer Science of the National Taiwan University, Taipei 106 Taiwan, abril 2010. Disponível em `http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf`. 29

T. Hudik and J. Zizka. Adaptive Resonance Theory. `http://users.visualserver.org/xhudik/art/`, 2011. ix, 6, 35, 58, 59, 100

F. Jossinet, T. E. Ludwig, and E. Westhof. RNA structure: bioinformatic analysis. *Science Direct*, 2007. 13

N. K. Kasabov. *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering.* The MIT Press, Cambridge, Massachusetts, USA, 1998. x, 3, 19, 20, 21, 30, 32, 34, 35, 36, 59

W. J. Kent, C. W. Sugnet, T. S. Furey, K. M. Roskin, T. H. Pringle, A. M. Zahler, Haussler, and David. The Human Genome Browser at UCSC. *Genome Research*, 12 (6):996–1006, 2002. 15

T. Kin, K. Yamada, G. Terai, H. Okida, Y. Yoshinari, Y. Ono, A. Kojima, Y. Kimura, T. Komori, and K. Asai. fRNAdb: a platform for mining/annotating functional RNA candidates from non-coding RNA sequences. *Nucleic Acids Research*, 35(suppl 1):D145–D148, 2007. 16

T. Kohonen. *Self-Organizing Maps.* Springer, Berlin, Germany, 2001. 3, 21, 29

T. Kohonen, J. Hynninen, J. Kangas, J. Laaksonen, , and K. Torkkola. LVQ_PAK: The Learning Vector Quantization Program Package. Technical Report A30, Helsinki University of Technology, Laboratory of Computer and Information Science, FIN-02150 Espoo, Finland, 1996a. ix, x, xvi, 3, 6, 22, 32, 33, 34, 42, 61, 101

T. Kohonen, J. Hynninen, J. Kangas, and J. Laaksonen. SOM_PAK: The Self-Organizing Map Program Package. Technical report, Helsinki University of Technology, Espoo, Finland, 1996b. ix, xiii, xvi, 6, 25, 26, 31, 41, 48, 53, 54, 64, 98

L. Kong, Y. Zhang, Z. Ye, X. Liu, S. Zhao, L. Wei, and G. Gao. CPC: assess the protein-coding potential of transcripts using sequence features and support vector machine. *Nucleic Acids Res.*, 35:345–349, 2007. 2, 17, 37, 66, 67, 68, 79

J. Kyte and R. F. Doolittle. A Simple Method for Displaying the Hydropathic Character of a Protein. *J. Mol. Biol.*, 157:105–132, 1982. xiii, 43, 44

S. S. Lakshmi and S. Agrawal. piRNABank: a web resource on classified and clustered Piwi-interacting RNAs. *Nucleic Acids Research*, pages 1–5, 2007. xiii, 15

A. M. Lesk. *Introduction to Bioinformatics.* Oxford University Press, Great Clarendon Street, Oxford, UK, 2002. 9

L. Lestrade and 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. 53

W. Li and A. Godzik. CD-HIT: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, 22(13), 2006. 49

C. Liu, B Bai, G. Skogerbo, L. Cai, W. Deng, Y. Zhang, D. Bu, Y. Zhao, and R. Chen. NONCODE: an integrated knowledge database of non-coding RNAs. *Nucleic Acids Research.*, 33:D112–D115, 2005. xiii, 2, 3, 16, 48, 51

J. Liu, J. Gough, and B. Rost. Distinguishing protein-coding from non-coding RNAs through support vector machines. *PLoS Genet*, 2(e):29–36, 2006. 17, 37, 43, 48, 63, 66, 74, 79, 87

S. Lê, J. Josse, and F. Husson. FactoMineR: An R Package for Multivariate Analysis. *Journal of Statistical Software*, 25(1):1–18, 2008. ix, 27, 60, 102

A. Machado-Lima, Portillo H. A., and A. M. Durham. Computational methods in non-coding RNA research. *Mathematical Biology*, 56:15–49, 2008. 13, 14, 16, 17, 18

M. Machida, K. Asai, M. Sano, T. Tanaka, T. Kumagai, G. Terai, K. Kusumoto, T. Arima, O. Akita, Y. Kashiwagi, K. Abe, K. Gomi, H. Horiuchi, K. Kitamoto, T. Kobayashi, M. Takeuchi, D. W. Denning, J. E. Galagan, W. C. Nierman, J. Yu, D. B. Archer, J. W. Bennett, D. Bhatnagar, T. E. Cleveland, N. D. Fedorova, O. Gotoh, H. Horikawa, A. Hosoyama, M. Ichinomiya, R. Igarashi, K. Iwashita, P. R. Juvvadi, M. Kato, Y. Kato, T. Kin, A. Kokubun, H. Maeda, N. Maeyama, J. Maruyama, H. Nagasaki, T. Nakajima, K. Oda, K. Okada, I. Paulsen, K. Sakamoto, T. Sawano, M. Takahashi, K. Takase, Y. Terabayashi, J. R. Wortman, O. Yamada, Y. Yamagata, H. Anazawa, Y. Hata, Y. Koide, T. Komori, Y. Koyama, T. Minetoki, S. Suharnan, A. Tanaka, K. Isono, S. Kuhara, N. Ogasawara, and H. Kikuchi. Genome sequencing and analysis of Aspergillus oryzae. *Nature*, 438(7071):1157–1161, 2005. 52

C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008. Disponível em `http://nlp.stanford.edu/IR-book/html/htmledition/irbook.html`. 66

N. R. Markham and M. Zuker. UNAFold: software for nucleic acid folding and hybriziation. *Methods in Molecular Biology*, II(453):3–31, 2008. 37, 49

B. W. Matthews. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 405(2): 442–451, 1975. 23, 24, 25

D. Q. Mendes and M. F. S. Oliveira. Tutorial de Redes Neurais. Aplicações em Bioinformática. `http://www.lncc.br/~labinfo/tutorialRN/`, 2009. 19

T. R. Mercer, M. E. Dinger, and J. S. Mattick. Long non-coding RNAs: insights into functions. *Nat Rev Genet*, 10(3):155–159, 2009. 14, 81

T. M. Mitchell. *Machine Learning*. McGraw-Hill International Editions, Ohio, USA, 1997. 3, 21

S. M. Mount, V. Gotea, C-F. Lin, K. Hernandez, and W. Makalowski. Spliceosomal Small Nuclear RNA Genes in Eleven Insect Genomes. *RNA*, 13(1):5–14, 2007. 2

E. P. Nawrocki, D. L. Kolbe, and S. R. Eddy. Infernal 1.0: Inference of RNA alignments. *Bioinformatics*, 25:1335–1337, 2009. 2, 4, 37, 50, 66, 96

Broad Institute of Harvard and MIT. *Coccidioides* group Database. `http://www.broadinstitute.org/annotation/genome/coccidioides_group/MultiHome.html`, 2012a. 52

Broad Institute of Harvard and MIT. *Paracoccidioides brasiliensis* Sequencing Project. `http://www.broadinstitute.org/annotation/genome/paracoccidioides_brasiliensis/MultiHome.html`, 2012b. `http://www.broadinstitute.org/`. 52

K. C. Pang, S. Stephen, P. G. Engstrom, K. Tajul-Arifin, W. Chen, C. Wahlestedt, B. Lenhard, Y. Hayashizaki, and J. S. Mattick. RNAdb - a comprehensive mammalian noncoding RNA database. *Nucleic Acids Research.*, 33:D125–D130, 2005. xiii, 15, 48, 51

K. Pearson. On Lines and Planes of Closest Fit to Systems of Points in Space. *Philosophical Magazine*, 2:559–572, 1901. 3, 26

V. J. Promponas, A. J. Enright, S. Tsoka, Kreil D. P., C. Leroy, S. Hamodrakas, S. Sander, and C. Ouzonis. CAST: an iterative algorithm for the complexity analysis of sequence tracts. *Bioinformatics*, 16(10):915–922, 2000. ix, 6, 40, 41, 43, 44, 59, 104

C. Ralha, H. Schneider, L. Fonseca, M. Walter, and M. Brígido. Using lt;igt;BioAgentslt;/igt; for Supporting Manual Annotation on Genome Sequencing Projects. In *Advances in Bioinformatics and Computational Biology*, volume 5167 of *Lecture Notes in Computer Science*, pages 127–139. Springer Berlin / Heidelberg, 2008. 97

P. Rice, I. Longden, and A. Bleasby. EMBOSS: The European molecular biology open software suite. *Trends Genet.*, 16:276–277, 2000. 43, 44

E. Rivas and S. Eddy. Noncoding RNA gene detection using comparative sequence analysis. *BMC Bioinformatics*, 2(1):8, 2001. 2

RNA, 2011. RNA codons. `http://www.mysciencebox.org/`, outubro 2011. x, 8, 9

J. C. Setubal and J. J. Meidanis. *Introduction to Computational Molecular Biology.* PWS Publishing Company, Boston, MA, 2000. xiii, 1, 7, 9, 11, 12, 13

K. Shimizu, J Adachi, and Y. Muraoka. ANGLE: a sequencing errors resistant program for predicting protein coding regions in unfinished cDNA. *Journal of Bioinformatics and Computational Biology*, 4(3):649–664, 2006. ix, 6, 40, 41, 42, 58, 103

T. C. C. Silva. SOM-PORTRAIT: um método para identificar RNA não codificador utilizando Mapas Auto Organizáveis. `http://hdl.handle.net/123456789/186`, 2009. Monografia de Conclusão de Curso. 3, 39

T. C. C. Silva, P. A. Berger, R. Arrial, R. Togawa, M. M. Brigido, and M. E. M. T. Walter. SOM-PORTRAIT: Identifying Non-coding RNAs Using Self-Organizing Maps. In *Advances in Bioinformatics and Computational Biology*, volume 5676 of *Lecture Notes in Computer Science*, pages 73–85. Springer Berlin / Heidelberg, 2009. 2, 3, 5, 17, 40, 53

S. Sinha, T. Singh, V. Singh, and A. Verma. Epoch determination for neural network by self-organized map (SOM). *Computational Geosciences*, 14:199–206, 2010. 54, 55

T.F. Smith and M.S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197, 1981. 45

D. Soll and U. L. RajBhandary. *tRNA: Structure,Biosynthesis, and Function.* ASM Press, Washington DC, 1995. 1

D. Song, Y. Yang, B. Yu, B. Zheng, Z. Deng, B. Lu, X. Chen, and T. Jiang. Computational prediction of novel non-coding RNAs in Arabidopsis thaliana. *BMC Bioinformatics*, 10 (Suppl 1):S36, 2009. 53

M. C. P. Souto, A. C. Lorena, A. C. B. Delbem, and A. C. P. L. F. Carvalho. *Técnicas de Aprendizado de Máquina em Problemas de Biologia Molecular.* Porto Alegre, 2003. Apostila de minicurso II JAIA - XXIII Congresso da SBC. 22

J. E. Stajich, D. Block, K. Boulez, S. E. Brenner, S. A. Chervitz, C. Dagdigian, G. Fuellen, J. G. R. Gilbert, I. Korf, H. Lapp, H. Lehväslaiho, C. Matsalla, C. J. Mungall, B. I. Osborne, M. R. Pocock, P. Schattner, M. Senger, L. D. Stein, E. Stupka, M. D. Wilkinson, and E. Birney. The Bioperl Toolkit: Perl Modules for the Life Sciences. *Genome Research*, 12(10):1611–1618, 2002. 40, 106

S. V. Stehman. Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment*, 62(1):77–89, 1997. 23, 24

M. Szymanski, J. Barciszewski, and V. A. Erdmann. *Noncoding RNAs: Molecular Biology and Molecular Medicine, chapter Riboregulators: An Overview.* Springer, 2003. 8

M. Szymanski, V. A. Erdmann, and J. Barciszewski. Noncoding RNAs database (ncRNAdb). *Nucleic Acids Research*, 35(suppl 1):D162–D164, 2007. 1, 2, 13, 14, 22

P. Sætrom, R. Sneve, K. I. Kristiansen, O. Snøve, T. Grünfeld, T. Rognes, and E. Seeberg. Predicting non-coding RNA genes in Escherichia coli with boosted genetic programming. *Nucleic Acids Research*, 33(10):3263–3270, 2005. 53

V. Taira, T. C. C. Silva, M. E. M. T. Walter, P. A. Berger, and M. M. Brígido. Reducing the number of attributes in a non-coding identifier based on Support Vector Machine. In *BSB Digital Proceedings*, 2011. x, 29, 30

S. Thrun and P. Norvig. Introduction to Artificial Intelligence. `https://www.ai-class.com/`, outubro 2011. Material de Curso da KnowIt em parceria com a Universidade Stanford (Out-Dez 2011). xiii, 22, 23

D. Ulveling, C. Francastel, and F. Hubé. When one is better than two: RNA with dual functions. *Biochimie*, 93(4):633–644, 2011. 23

Valley-Fever.org. Valley fever. http://www.valley-fever.org/index.html, 2006. 52

C. Wang, C. Ding, R. F. Meraz, and S. R. Holbrook. PSoL: a positive sample only learning algorithm for finding non-coding RNA genes. *Bioinformatics*, 22(21):2590–2596, 2006. 2, 79

S. Washietl, I. L. Hofacker, and P. F. Stadler. Fast and reliable prediction of noncoding RNAs. *Proceedings of the National Academy of Sciences of the United States of America*, 102(7):2454–2459, 2005. 2

J. D. Watson and F. H. C. Crick. Molecular structure of nucleic acids. *Nature*, 171(4356): 737–738, 1953. 1, 8, 11, 13

M. Zuker and P. Stiegler. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research*, 9(1), November 1981. 17

M. Zuker, D. H. Matthews, and D. H. Turner. Algorithms and thermodynamics for RNA secondary structure prediction: A practical guide. In *RNA Biochemistry and Biotechnology*, NATO ASI Series, pages 11–43. Kluwer Academic Publishers, 1999. 2