ESCOLA POLITÉCNICA
DE PERNAMBUCO

Departamento de Sistemas e Computação

Aplicação de Redes Neurais Artificiais, Combinadas com Extratores de Características, para Classificação de Famílias de Proteínas.

Trabalho de Conclusão de Curso

Engenharia da Computação

Nome do Aluno: Frederico Duarte de Menezes
Orientador: Prof. Adriano Lorena

UNIVERSIDADE
DE PERNAMBUCO

Frederico Duarte de Menezes

Aplicação de Redes Neurais Artificiais, Combinadas com Extratores de Características, para Classificação de Famílias de Proteínas.

Monografia apresentada como requisito parcial para obtenção do diploma de Bacharel em Engenharia da Computação pela Escola Politécnica de Pernambuco – Universidade de Pernambuco.

Recife, novembro de 2008.

Dedico sem diferenças a todos que acreditaram ou duvidaram que eu seria capaz de chegar até aqui.
Obrigado pelo incentivo.

Agradecimentos

Agradeço a Deus primeiramente, pela paciência e força de vontade a mim dada.

Agradeço a minha família por me aturarem durante estes anos de estudo, com minhas irritações e frustrações.

Agradeço a minha namorada, pois sei como é difícil conviver com uma pessoa como eu.

Agradeço aos meus amigos e colegas, que inconscientemente foram minha bóia de salvação várias vezes durante minha caminhada.

Agradeço também ao professor Adriano Lorena, por me aceitar como aluno para a realização deste trabalho.

Resumo

O crescimento contínuo e exponencial de bases de dados sobre DNA e proteínas tem gerado uma necessidade de desenvolvimento de ferramentas computacionais avançadas para a análise destes dados e para a resolução de problemas de classificação de famílias e funções de proteínas. Neste trabalho, desenvolveu-se uma metodologia simples de extração de características de seqüências de proteínas, para a geração de bases de dados padronizadas para o treinamento de classificadores inteligentes, especificamente redes neurais artificiais do tipo MLP (*Multi Layer Perceptron*) e árvores de decisão. Os resultados obtidos foram bastante satisfatórios, mostrando a aplicabilidade da metodologia desenvolvida para a solução de problemas reais de classificação em bioinformática. A simplicidade dos algoritmos de extração de características desenvolvidos, assim como as etapas de pré-processamento das bases de dados analisadas, são o foco principal deste trabalho.

Abstract

The continuous and exponential growth of DNA and proteins data base have called the attention for the development of advanced computer tools to analyze those data and to resolve problems about function and families classification of proteins. In this work a simple methodology was developed to extract the proteins sequences, in order to generate a standard data base to training smart classifiers, especially artificial neural networks such as MLP (Multi Layer Perceptron) and decision trees. The results were very satisfactory, showing the applicability of the developed methodology to solve the classification real problems in bioinformatics area. The simplicity of the extraction of characteristics algorithms developed, as well as the pre-processing steps of the data base, are the principals aims of this work.

Sumário

Índice de Figuras

Índice de Tabelas

Tabela de Siglas

DNA - ácidos desoxirribonucléicos

RNA - ácidos ribonucléicos

AAs – aminoácidos

AD – árvores de decisão

ANN – redes neurais artificiais

MLP – *Multi Layer Perceptron*

AUC – área sob curva ROC

PCA – análise de componentes principais

Capítulo 1

Introdução

A bioinformática pode ser definida como a ciência que envolve a coleta, manipulação, análise e transmissão de enormes quantidades de dados, a respeito das três principais fontes de armazenamento e compartilhamento de informação de todos os seres vivos: ácidos desoxirribonucléicos (DNA), ácidos ribonucléicos (RNA) e proteínas[1]. Para tanto, tem como ferramentas principais para a execução destas tarefas os computadores e seus sistemas de processamento, capazes de processar informações baseadas nos três alfabetos que regem a criação e funcionamento destas fontes naturais de dados (ver descrição dos alfabetos na seção seguinte).

Como algumas das aplicações para os estudos desenvolvidos pela bioinformática, temos a busca de novos alvos de medicamentos e vacinas para diversas patologias, assim como a modificação de determinados trechos de DNA de certos seres vivos, para a produção industrial de substâncias de interesse terapêutico.

1.1 – Conceitos básicos

O grande Dogma, definido pelo geneticista James Watson [1], que rege os estudos atuais no campo da biologia molecular e que serviu como base para a criação dos diversos projetos genoma criados nos últimos 10 anos, pode ser descrito da seguinte forma: (i) as grandes cadeias dos biopolímeros conhecidos como DNA são as responsáveis pelo armazenamento de toda a informação que permite a **manutenção** e **perpetuação** de todos os seres vivos; (ii) estas duas tarefas são executadas e mantidas de forma correta através da tradução das informações do DNA na forma de proteínas; (iii) e isto só é possível através da utilização de moléculas mensageiras, conhecidas como RNA, que traduzem a informação do DNA para o maquinário celular responsável pela confecção das proteínas. O primeiro passo desta cadeia de processamento de informação, que corresponde à passagem

de informação do DNA para o RNA é denominada **transcrição**, sendo o segundo passo, correspondente a passagem de informação do RNA para proteínas, denominado de **tradução** [2]. A Figura 1 ilustra estas etapas.

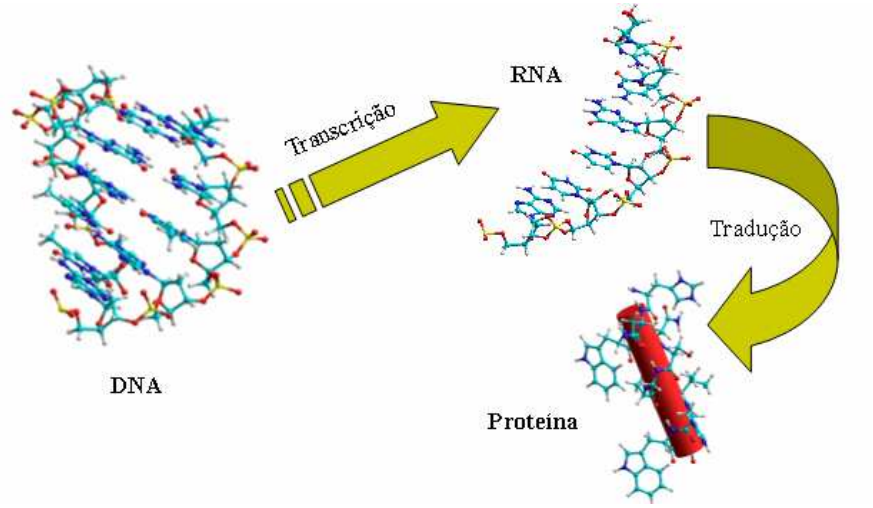

Figura 1 - Esquema geral dos processos de transcrição e tradução.

Cada um dos componentes citados (DNA, RNA e proteínas) possui o seu próprio alfabeto de códigos que representam a composição de cada biopolímero, a saber:

- O DNA é uma composição de moléculas conhecidas como desoxirribonucleotídeos que podem assumir quatro valores possíveis. Estes valores são: A (Adenina), T (Timina), C (Citosina) e G (Guanina). Estas moléculas são alinhadas aos pares, para formar as estruturas em dupla hélice que compõem os filamentos de DNA. As únicas interações de pareamento permitidas são entre A e T ou C e G. Assim, sabendo-se a seqüência de desoxirribonucleotídeos de uma das cadeias de DNA, é simples deduzir a sua cadeia complementar;

- O RNA é uma composição de moléculas conhecidas como ribonucleotídeos que, assim como o DNA, podem assumir quatro valores possíveis: A (Adenina), U (Uracila), C (Citosina) e G (Guanina). Neste caso, o pareamento permitido de

moléculas é entre A e U ou C e G. Raros são os casos de fragmentos de RNA de seqüência dupla, sendo mais freqüente as estruturas de RNA com uma única seqüência pareada internamente;

- As proteínas são compostas, naturalmente, por um conjunto de 20 tipos de aminoácidos (AAs) distintos (ver lista completa no anexo B). Este alfabeto de códigos é capaz de representar proteínas diversas, com estruturas e funções biológicas bastante distintas.

1.2 – Classificação: um problema chave da bioinformática.

O crescimento contínuo e exponencial de bases de dados, contendo informações sobre o seqüenciamento de DNA e proteínas de diversos organismos, tem gerado uma grande necessidade de novos recursos computacionais para a análise e gerenciamento destas informações [3]-[5]. Dentre os vários problemas abordados pela bioinformática, temos que a classificação de proteínas, no que diz respeito a suas estruturas físicas e/ou funções biológicas, apresenta-se como um problema chave [6]. Novas ferramentas capazes de extrair informações relevantes sobre proteínas de interesse ou capazes de realizar a predição de propriedades físico-químicas de novas proteínas, baseando-se em informações disponíveis em diferentes bases de dados, vêm sendo desenvolvidas por diferentes grupos de pesquisa [6]-[10].

Para um grande número de proteínas, o conhecimento básico de sua seqüência de aminoácidos é suficiente para que seja possível a obtenção de informações sobre suas funções biológicas, estrutura tridimensional, etc. Porém, várias técnicas computacionais atualmente utilizadas para esses fins possuem a desvantagem de possuir uma complexidade computacional de ordem $O(n^2)$ [4], no que diz respeito ao tamanho das seqüências de AA analisadas. Isto limita a utilização dessas ferramentas para a análise de grandes massas de dados, visto que várias proteínas podem ser constituídas de centenas de AAs.

Felizmente, há técnicas computacionais de análise, baseadas em métodos de aprendizado de máquinas, cuja complexidade computacional é bem inferior a $O(n^2)$ [4]. É o caso das técnicas de árvore de decisão (AD) e redes neurais artificiais (ANN). As RNAs oferecem uma arquitetura computacional singular, visto que o processamento sobre uma massa de dados é realizado de forma dinâmica e rápida, isto é, as ANNs são capazes de se adaptar a novos dados que lhe sejam apresentados, em tempo real [4][5][7][9]-[11]. Esta adaptação ocorre através da atualização dos pesos relativos ao processamento de cada neurônio da ANN. O objetivo final dessas atualizações é a minimização da soma quadrática dos erros da ANN, considerando-se cada padrão de treinamento apresentado à rede. Porém, um fator crucial para o bom desempenho de uma ANN, ou de qualquer técnica computacional de análise, é a forma como os dados a serem processados são apresentados à ferramenta.

1.3 – Pré-processamento de dados: uma etapa crucial para utilização de classificadores.

Um bom pré-processamento dos dados “brutos” utilizados para o treinamento de qualquer classificador, estatístico ou baseado em inteligência computacional, garante a confiabilidade das respostas geradas pelo classificador utilizado, assim como pode diminuir consideravelmente a quantidade de dados que o sistema necessitará processar. Por isto, como primeira etapa deste processo, é comum o uso de algoritmos que sejam capazes de extrair informações, das mais diversas naturezas, a respeito dos dados que se deseja analisar. Com isso gera-se uma nova base de dados que contém informações representativas sobre os dados originais, isto é, geram-se “metadados”, que nada mais são do que informações sobre informações, que podem ser utilizados, finalmente, para o treinamento e teste de uma ANN, por exemplo [4][7][10]. Além disso, estes extratores devem ser capazes de padronizar o tamanho de cada instância gerada, permitindo que dados originais, com números variáveis de parâmetros entre si, sejam representados por metadados com tamanho fixo. Isto é de suma importância para o treinamento de classificadores como ANNs, que exigem um tamanho fixo de parâmetros de entrada de sua arquitetura.

No caso das classificações de proteínas, isto é um fator que limita a utilização de certos classificadores, visto que proteínas que são rotuladas como tendo a mesma funcionalidade em espécies distintas, podem possuir um número muito diferente de aminoácidos entre si. Logo, faz-se necessário à padronização do tamanho de cada instância que representa estas proteínas para o treinamento de um classificador. A Figura 2 ilustra um exemplo deste tipo de problema: as duas proteínas representadas por um cabeçalho e suas respectivas seqüências de aminoácidos são consideradas como pertencentes a família de proteínas relativa a proteínas de ligação a rRNA, embora possuam composição e tamanho diferentes.

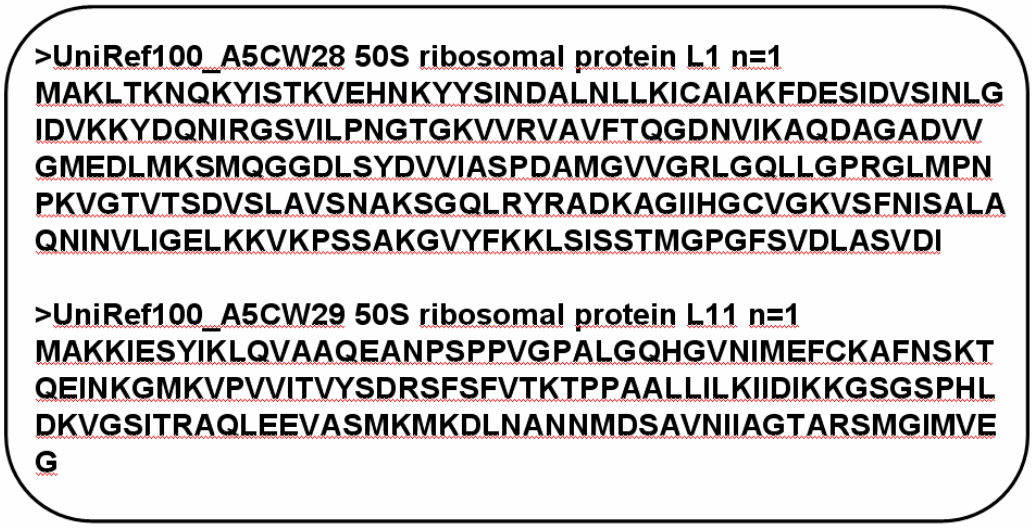

Figura 2 – Exemplos de proteínas com mesma funcionalidade, mas com tamanhos diferentes.

Contudo, caso o extrator de características utilizado para a geração da base de dados padronizada não obtenha parâmetros representativos dos dados originais tratados, a taxa de erro do classificador a ser treinado com esta nova base será consideravelmente alta, visto que os parâmetros obtidos não irão favorecer uma separabilidade entre as instâncias pertencentes a classes distintas. Ou seja, as taxas de falso positivo e falso negativo nas classificações realizadas serão elevadas, invalidando a metodologia de classificação empregada.

Entretanto, não é apenas a utilização de extratores de características que garantirá um bom pré-processamento. Mesmo com a utilização de extratores específicos para cada tipo de natureza dos dados analisados, a base de metadados

obtida ainda poderá conter instâncias de dados que podem se comportar como "*outliers*" dentro de uma dada classe rotulada. Ou seja, instâncias rotuladas como pertencente a uma classe específica poderá se comportar como um elemento estranho desta classe, durante o processo de treinamento de um determinado classificador. Com isto, a convergência da taxa de erro do classificador em direção a um mínimo global poderá ser comprometida, resultando em erros elevados de classificação. Diferentemente do caso citado anteriormente, o comprometimento da classificação neste caso pode ser devido a uma rotulagem errada de algumas instâncias na base de dados original, ou ainda devido a uma elevada diversidade no comportamento dos parâmetros das instâncias pertencentes a uma dada classe.

Para resolver este tipo de problema, pode-se empregar algumas técnicas de "*clustering*", sobre as instâncias pertencentes a uma dada classe, com o intuito de se identificar e eliminar instâncias que se comportem como *outliers*.

1.4 – Motivação do Trabalho.

Atualmente, há uma enorme quantidade de trabalhos existentes na literatura a respeito da resolução de problemas específicos de classificação de proteínas. Entretanto, muitos destes trabalhos não dão a devida ênfase as etapas de pré-processamento das bases de dados utilizadas para o treinamento dos respectivos classificadores adotados, assim como não preconizam a utilização de técnicas de extração de características simples, como forma de reduzir a complexidade da base de dados gerada.

Devido a isto, escolheu-se como linha de desenvolvimento deste trabalho à descrição de um algoritmo geral de pré-processamento de dados e execução do treinamento de ANNs e AD, que se julga ser bastante viável para a formalização da execução de classificações relativas a bases de dados de proteínas. Isto também permite uma melhor rastreabilidade dos fatores que melhoram ou pioram as taxas de acerto dos classificadores, através do desenvolvimento de uma metodologia simples para a extração de características das proteínas analisadas.

1.5 – Objetivos e Metas.

O objetivo principal deste trabalho é o desenvolvimento de algoritmos simples para a extração de características relevantes de seqüências de aminoácidos que representem proteínas diversas, para a geração de bases de dados padronizada a ser utilizada para o treinamento de classificadores inteligentes.

Um segundo objetivo é o treinamento de classificadores inteligentes, especificamente redes neurais artificiais do tipo *Multi Layer Perceptron* (MLP) e árvores de decisão, utilizando a base de dados padronizada gerada pelos algoritmos de extração de características, para a resolução do problema de classificação de famílias de proteínas, onde cada família é rotulada segundo sua função biológica.

Como meta para o final deste trabalho, espera-se a implementação de um sistema automático de classificação de novas proteínas, baseada nas classes utilizadas para a criação dos classificadores.

1.6 – Estrutura do Trabalho

O trabalho encontra-se dividido nos seguintes capítulos:

- Capítulo 2: Trata dos fundamentos das técnicas de extração de características e dos classificadores utilizados neste trabalho;
- Capítulo 3: Descreve a metodologia utilizada neste trabalho para o pré-processamento e classificação das bases de dados utilizadas neste trabalho;
- Capítulo 4: Trata da apresentação e discussão dos resultados obtidos através do emprego da metodologia descrita no Capítulo 3;
- Capítulo 5: Trata sobre as conclusões finais do trabalho e propostas para trabalhos futuros.

Capítulo 2

Técnicas de extração de características e classificadores utilizados.

Várias são as técnicas empregadas para a extração de características de bases de dados das mais diversas naturezas. Entretanto, o ponto chave para o sucesso do emprego de uma ou mais técnicas para a resolução de um dado problema é o conhecimento aprofundado dos fundamentos matemáticos e/ou estatísticos que regem a execução de cada técnica utilizada. Por isso, neste capítulo trataremos de forma clara, os fundamentos que regem os pontos mais importantes das técnicas de extração de características utilizadas neste trabalho, assim como dos classificadores também utilizados.

2.1 – Noções de distribuição de freqüências

Dado um vetor w qualquer de tamanho n ($w=[x_1\ x_2\ x_3\ ...\ x_n]$), representado por valores x, que obedeçam a seguinte condição:

$$x \in B \text{ , tal que } B = \{b_1,b_2,...,b_m\}$$

Seja B um conjunto discreto de valores reais, iremos assumir que qualquer vetor w será uma combinação aleatória de valores exclusivamente contidos no conjunto B.

Assumindo-se agora um vetor d de tamanho m, cujos valores d_1, d_2, ..., d_m sejam as razões entre o número de vezes que cada valor pertencente ao conjunto B aparece no vetor w e o número total de elementos em w, podemos definir d como o vetor de **distribuição de freqüência** dos valores contidos no vetor w.

As distribuições de freqüências são um elemento essencial para a análise de quaisquer tipos de séries de valores que pertençam a um dado conjunto numérico. Isto ocorre devido ao fato de que séries com tamanhos diferentes, mas cujos valores pertençam a um mesmo conjunto permitido, podem ser comparados, através de certos parâmetros advindos de análises efetuadas sobre as respectivas distribuições de freqüência das séries.

2.1.1 – Parâmetros de forma de distribuições

Para uma distribuição qualquer, temos que alguns parâmetros relativos ao formato da distribuição são de grande interesse para a realização de suas análises.

Dentre os parâmetros de forma utilizados para a análise de distribuições, a **curtose** apresenta-se como um dos mais utilizados [12]. Este parâmetro pode ser definido como uma medida do “achatamento” ou “afunilamento” de uma distribuição, considerando-se esta distribuição com um caráter de uma distribuição normal. A equação que define a curtose é:

$$Curtose = \frac{\mu_4}{\sigma^4} - 3 \qquad \text{Equação 1}$$

Na Equação 1, temos que μ_4 refere-se ao quarto momento central da distribuição analisada e σ é o desvio padrão da distribuição, onde μ_4 é representado pela equação:

$$\mu_4 = \sum_{1}^{n} \left(x_n - \bar{x} \right)^4 d_n \qquad \text{Equação 2}$$

Onde $\bar{x}$ representa a média dos valores valores presentes em um vetor w, x_n representa cada elemento presente em w e d_n representa a distribuição de freqüência de cada elemento b_n que aparece em w.

Para um valor de curtose igual a 0, temos que a distribuição tem um formato exato de uma distribuição normal, sendo denominada de distribuição **mesocúrtica**. Para um valor de curtose maior do que 3, a distribuição apresenta-se mais “afunilada”, sendo mais alta do que a sua distribuição normal equivalente. Neste

caso, denomina-se a distribuição de **leptocúrtica**. Por fim, para um valor de curtose menor do que 3, temos uma distribuição mais “achatada”, sendo denominada de **platicúrtica**.

Outro parâmetro de forma bastante utilizado para a análise de distribuições diz respeito a semelhança entre uma distribuição analisada em uma dada distribuição modelo e denomina-se entropia divergente de Kullback-Leibler [13]. A equação que rege este parâmetro tem a seguinte forma:

$$D_{KL} = \sum_{i=1}^{n} P(i) \log \frac{P(i)}{Q(i)} \qquad \text{Equação 3}$$

Na Equação 2, P(i) são as freqüências relativas da distribuição real que se deseja analisar e Q(i) são as freqüências relativas de uma distribuição modelo. Para que as duas distribuições comparadas sejam consideradas idênticas, o valor de D_{KL} deve ser igual a zero. Quanto mais distantes de zero for o valor de D_{KL}, maior será a diferença entre as formas das distribuições.

O mais interessante de se utilizar estes tipos de parâmetros para a análise de distribuições é a fácil visualização e correlação dos valores obtidos com as interpretações realizadas.

2.2 – Redes neurais artificiais e Árvores de decisão

Redes neurais artificiais são sistemas computacionais inteligentes que utilizam a idéia de funcionamento dos neurônios biológicos para a resolução de problemas de classificação de padrões ou previsão de valores. Em um neurônio real, o somatório de um conjunto de sinais de entrada no mesmo deve possuir um valor maior do que um parâmetro interno da célula, denominado de limiar de ação, para que o neurônio possa passar uma dada informação adiante. Caso o valor do somatório seja menor do que o valor do limiar, a informação não é transmitida.

Em uma ANN, os neurônios da rede possuem pontos de entradas de valores numéricos em sua estrutura, que são marcados com valores denominados pesos.

Quando um conjunto de valores é apresentado a um dado neurônio, cada valor é multiplicado pelo seu respectivo peso. O somatório desses produtos são então comparados com o valor de limiar do neurônio. Caso o valor do somatório seja maior do que o valor de limiar, a saída do neurônio será igual a um. Caso contrário, o valor será igual a zero. Vale ressaltar que estes valores de saída podem variar, mediante o tipo de função de ativação que é utilizada nos neurônios de uma ANN.

Para que possam realizar tarefas de classificação ou de predição, estes sistemas são treinados com bases de dados rotuladas, ou seja, com valores de classificação ou predição conhecidos. Durante a fase de treinamento, os pesos das vias de entrada de sinais em cada neurônio da rede neural são atualizados, mediante um modelo de aprendizado, com o intuito de se minimizar ao máximo a taxa de erro de classificação ou predição do sistema. Quando a taxa de erro da ANN converge para um mínimo global no seu espaço de possibilidades, finaliza-se o treinamento da ANN.

2.2.1 – Redes do tipo *Multi Layer Perceptron*

Dentre as várias arquiteturas de ANNs existentes atualmente, neste trabalho utilizaremos as redes do tipo MLP. Esta é uma arquitetura do tipo *feedfoward* que utiliza neurônios baseados no modelo perceptron, criado por Frank Rosenblatt em 1958 [14]. Como a arquiteturas das redes MLP utilizam pelo menos uma camada de neurônios escondidos, interligados aos neurônios de entrada e de saída da rede, é possível a resolução de problemas não-lineares com este tipo de ANN.

O algoritmo de aprendizado utilizado no treinamento das redes MLP em nosso trabalho é o *backpropagation*, descrito inicialmente por Paul Werbos em 1974 [14]. Neste algoritmo, a correção do erro de cada neurônio é realizada no sentido inverso do fluxo de dados da rede. Dado um padrão apresentado à rede, esta ativa seus neurônios gerando determinados sinais de saída da rede. Caso haja uma diferença entre a saída da rede e o valor esperado do padrão, os pesos dos neurônios da ultima camada da rede são atualizados para a minimização do erro de classificação, ou predição, sendo esta atualização propagada para a camada de neurônios anterior a camada de saída. Esta atualização é propagada em toda a

rede, até que se alcance a camada de neurônios de entrada da rede. A Figura 3 mostra um esquema ilustrando este tipo de arquitetura.

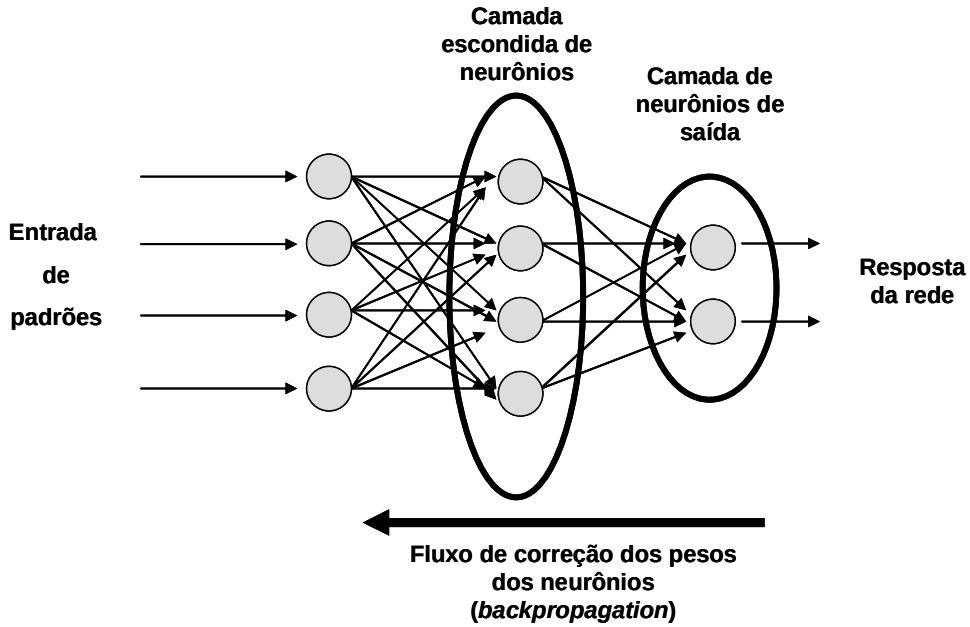

Figura 3 – Esquema geral de uma rede MLP com algoritmo de aprendizado do tipo backpropagation.

2.2.2 – Árvores de decisão

Árvores de decisão constituem uma família de sistemas inteligentes cujo funcionamento baseia-se na idéia de uma árvore contendo nós internos e nós externos, conectados por ramos. Os nós internos são as unidades de suporte a decisão que decidem qual ramo um dado padrão, apresentado a árvore, deve seguir para ser deslocado para um nó filho (descendente), que pode ser um outro nó interno ou um nó externo. Por sua vez, um nó externo não participa do processo de decisão, apenas rotula um dado padrão apresentado. Em termos de importância nas tomadas de decisão, a raiz é o elemento mais importante do sistema, sendo seguido pelos seus nós mais próximos, sendo a importância dos nós subseqüentes decrescente com o aumento da distância em relação à raiz.

O processo de aprendizado destes classificadores se dá através da criação de regras de decisão para cada nó interno, com o intuito de se minimizar o erro de classificação do sistema. Durante este processo, uma árvore criada com uma

arquitetura inicial pode sofrer crescimento, onde novos nós são adicionados à estrutura da árvore como mais um ponto de separação dos padrões, ou pode ser podada, onde um nó criado com uma regra de pouca importância é retirado da estrutura da árvore. Basicamente, cada nó interno representa uma pergunta do tipo "é ou não é?", em relação a um dado parâmetro das instâncias analisadas. Por exemplo, digamos que um parâmetro relativo a uma base de dados seja a idade de um grupo de estudantes. Neste caso, este parâmetro será representado por um nó interno de uma árvore, e a pergunta que dará suporte a decisão de que caminho um padrão deverá seguir na árvore pode ser do tipo: "a idade de um estudante é maior do que 20 anos?". Caso positivo, siga pelo ramo esquerdo do nó, senão, siga pelo ramo direito. Dentre os algoritmos de classificação de árvores de decisão utilizados, temos o ID3 [15] e o C 4.5 [16] dentre os mais utilizados atualmente.

Além de servir para a realização de tarefas de classificação, as AD são também utilizadas para a seleção de parâmetros de entrada para outros classificadores. Esta aplicação advém das etapas de poda durante o treinamento de uma árvore e da hierarquização da importância dos parâmetros relacionados a cada nó. A partir da análise da estrutura de uma árvore de decisão treinada, pode-se identificar que parâmetros são mais importantes para a realização de um processo de classificação de padrões.

Capítulo 3

Metodologia experimental

3.1 - Aquisição da base de dados de proteínas:

Através de *queries* de buscas realizadas na base de dados UNIPROT [17], obteve-se um conjunto de instâncias que representam a família de proteínas conhecida como *rrna-binding*. Obteve-se também um conjunto de proteínas que tem funções biológicas semelhantes a família de proteínas previamente obtida, ao qual denominaremos de grupo contraste. E, finalmente, obteve-se um conjunto de proteínas diversas que representam o grupo negativo de proteínas (ver conjunto completos de *queries* no Anexo A). Ao final, a classe relativa a família rrna-binding foi constituída de 8000 instâncias, seguida do grupo de contraste constituído de 12504 instâncias e do grupo de padrões negativos com 9298 instâncias. As *queries* utilizadas para a obtenção destes conjuntos de instâncias foram retiradas do trabalho de Cay e Lin [10].

Esta é a base de dados “bruta”, contendo apenas as seqüências de aminoácidos que representam cada proteína. Como já mencionado anteriormente, a variabilidade no tamanho das seqüências de aminoácidos impossibilita a utilização direta desta base de dados para a realização de uma análise classificatória utilizando uma ANN ou uma AD. Para isto, necessita-se a realização de um pré-processamento desta base, com o intuito de padronizá-la e aumentar a sua representatividade.

3.2 – Etapas de pré-processamento:

3.2.1 – Obtenção dos parâmetros numéricos dos aminoácidos:

Para representar os aminoácidos que possam aparecer nas seqüências de proteínas analisadas, escolheu-se os parâmetros físico-químicos utilizados por Cay

e Lin[10], em trabalho semelhante de classificação. Os parâmetros para cada aminoácido encontram-se representados na Tabela 1.

Tabela 1 – **Parâmetros utilizados para representar os aminoácidos presentes nas proteínas analisadas.**

Aminoácido (Código)	*Carga do Aminoácido*	*Hidrofobicidade*	*Área superficial acessível*
A	0	1.8	44.1
R	1	-4.5	152.9
N	0	-3.5	80.8
D	-1	-3.5	76.3
C	0	2.5	56.4
Q	0	-3.5	100.6
E	-1	-3.5	99.2
G	0	-0.4	0
H	1	-3.2	98.2
I	0	4.5	90.9
L	0	3.8	92.8
K	1	-3.9	139.1
M	0	1.9	95.3
F	0	2.8	107.4
P	0	-1.6	79.5
S	0	-0.8	57.5
T	0	-0.7	73.4
W	0	-0.9	143.4
Y	0	-1.3	119.1
V	0	4.2	73.0

Além dos caracteres acima, três outros caracteres são comuns de aparecer em seqüências de aminoácidos referentes a proteínas seqüenciadas: X, que representa um aminoácido qualquer que não tenha sido identificado no processo de elucidação de uma dada proteína; B, que representa um aminoácido que foi identificado como sendo ácido aspártico (D) ou asparagina (N); e Z, que representa um aminoácido que foi identificado como sendo ácido glutâmico (E) ou glutamina (Q). Para estes códigos adicionais, atribuiu-se os seguintes valores de parâmetros físico-químicos:

Tabela 2 – **Parâmetros adicionais utilizados**

Aminoácido (Código)	***Carga do Aminoácido***	***Hidrofobicidade***	***Área superficial acessível***
B	-0.5	-0.5	78.55
Z	-0.5	-1.8	49.60
X	3	5.0	150.0

Os valores para B e Z foram obtidos segundo a média aritmética dos parâmetros relativos a D e N, no caso de B, e E e Q, referentes a Z. Como X representa qualquer aminoácido, atribuiu-se valores extremos para seus parâmetros, como forma de facilitar a sua separação dos valores relativos aos aminoácidos identificados.

3.2.2 - *Parsing* das proteínas em parâmetros calculados:

Após a obtenção da matriz de parâmetros químicos, implementou-se um conjunto de métodos em JAVA, com o auxílio da biblioteca BioJAVA [19], para a leitura das seqüências de proteínas organizadas em arquivos do tipo FASTA [20](ver formatação dos arquivos FASTA na seção Anexo B), que representa uma formatação bastante utilizada na área e bioinformática, para representar seqüências de proteínas e ácidos nucléicos.

Os métodos implementados permitiram a leitura destes arquivos, a extração das seqüências de aminoácidos que representam cada proteína e a tradução de

cada seqüência de aminoácidos nos seus respectivos parâmetros físico-químicos (Tabela 1). Cada tradução das seqüências em um dado parâmetro resultou em um novo arquivo de seqüências de valores, o que resultou em um total de três arquivos para representar o respectivo arquivo FASTA lido.

3.2.3 - Análise de distribuição estatística das seqüências traduzidas:

Após a geração dos arquivos de tradução, realizou-se uma análise estatística quanto a distribuição de freqüência dos valores de cada tipo de parâmetro físico-químico utilizado, em cada seqüência traduzida. Visto que as proteínas analisadas possuem uma grande variação nas suas quantidades de aminoácidos, a análise estatística permite que cada proteína seja representada por um número fixo de parâmetros, o que possibilita a geração de uma base de dados que possa ser utilizada para realizar o treinamento de classificadores inteligentes, como é o caso das ANN e AD.

A análise estatística realizada baseou-se na obtenção dos parâmetros estatísticos moda, desvio padrão, curtose e entropia divergente de Kullback-Leibler relativos ao histograma de distribuição de valores para cada tipo de parâmetro físico-químico, para cada proteína.

Ao final dessa análise, obteve-se a base de metadados referente aos padrões positivo, negativo e de contraste de proteínas. Nesta nova base de dados, cada proteína esta representada por um total de doze parâmetros, podendo agora ser utilizada para o treinamento dos classificadores.

3.2.4 – *Clustering* da base de metadados:

Com o intuito de se aumentar a homogeneidade dos padrões positivos utilizados nos experimentos de classificação, realizou-se um *clustering* através do algoritmo *k-means*, utilizando-se apenas os padrões referentes a classe *rrna-binding*.

O experimento foi realizado através de um *script* implementado em MATLAB, parametrizando-se em cinco o número de *clusters* de agrupamento gerados, utilizando-se como medida de similaridade entre os padrões a equação de distância Euclidiana.

Após a geração dos *clusters*, eliminou-se da base de dados final os padrões referentes aos dois clusters gerados com o menor número de elementos. Assim, partindo-se de uma base com 8000 padrões, após o *clustering* obteve-se uma nova base com 6980 padrões.

Devido a diversidade das bases de padrões negativos e de contraste, não se realizou esta etapa de pré-processamento para estas bases de dados, evitando-se a perda de generalização das mesmas, o que poderia causar *overfitting* dos classificadores gerados.

3.2.5 – Bases de dados utilizada nos experimentos de classificação:

Após a realização das tarefas de pré-processamento descritas anteriormente, gerou-se duas bases de dados referentes a junção de 2000 padrões positivos com 4000 negativos ou 4000 padrões de contraste. Todos os padrões foram selecionados de forma aleatória, a partir de seus conjuntos originais. Com isso, obteve-se as duas bases de dados padronizadas para a realização dos experimentos de classificação.

3.3 – Experimentos de Classificação:

Todos os experimentos de classificação, utilizando RNAs ou AD, foram realizados com o auxílio da ferramenta WEKA [18]. A configuração do computador utilizado nos experimentos é: processador AMD Semprom 2500+ de 1.41 GHz, com 512 MB de memória RAM. Utilizou-se a JVM versão 1.5.0.

3.3.1 – Configurações das MLP utilizadas:

Os experimentos utilizando-se RNA do tipo MLP, forma realizados mediante as configurações do número de neurônios, taxa de aprendizado, número de épocas de treinamento e momento de aprendizado, referentes a Tabela 3.

Tabela 3 – **Parâmetros utilizados para os treinamentos das redes MLP.**

Configuração	*Nº de neurônios*	*Taxa de aprendizado*	*Nº de épocas de treinamento*	*Momento*
1	4	0.05	500	0.0
2	8	0.05	500	0.0
3	4	1.0	500	0.0
4	8	1.0	500	0.0
5	4	0.05	1000	0.0
6	8	0.05	1000	0.0
7	4	1.0	1000	0.0
8	8	1.0	1000	0.0
9	4	0.05	500	0.1
10	8	0.05	500	0.1
11	4	1.0	500	0.1
12	8	1.0	500	0.1
13	4	0.05	1000	0.1
14	8	0.05	1000	0.1
15	4	1.0	1000	0.1
16	8	1.0	1000	0.1

3.3.2 – Configurações das AD utilizadas:

Os experimentos utilizando-se AD com o algoritmo de aprendizado C 4.5, forma realizados mediante as configurações do fator de confidência da árvore e o número mínimo de padrões classificados por folha, referentes a Tabela 4. O método da ferramenta WEKA referente a AD utilizada nos experimentos é o J48.

Tabela 4 - **Parâmetros utilizados para os treinamentos das AD**

Configuração	*Fator de confidência*	*Número mínimo de padrões por folha*
1	0.25	25
2	0.1	25
3	0.05	25
4	0.01	25
5	0.25	50
6	0.1	50
7	0.05	50
8	0.01	50
9	0.25	100
10	0.1	100
11	0.05	100
12	0.01	100

3.3.3 – Validação cruzada (*cross validation*)

Como forma de aumentar a confiabilidade dos experimentos de classificação realizados, utilizou-se o procedimento de validação cruzada para a medição das taxas de acerto das classificações executadas.

Através da validação cruzada, a base de dados utilizada para o treinamento de um classificador é dividida em n subgrupos. Enquanto um destes subgrupos gerados é utilizado para a etapa de teste do classificador, os demais n-1 subgrupos são utilizados para o treinamento do mesmo. Todos os subgrupos são utilizados para teste e treinamento do classificador, realizando-se um total de n experimentos de treinamento e teste. Com isso, a taxa de acerto final do processo de aprendizado

do classificador é uma média das taxas de acerto de cada experimento realizado, o que aumenta a confiabilidade dos experimentos realizados.

Nos experimentos realizados neste trabalho, utilizou-se validação cruzada, dividindo-se as bases de dados utilizadas em 10 *folds*.

3.3.4 – Parâmetros de avaliação das classificações:

Os parâmetros escolhidos para a avaliação dos experimentos de classificação realizados foram a taxa de acerto das classificações, a área sob a curva ROC (AUC) e o tempo de processamento de cada experimento realizado.

Capítulo 4

Resultados e Discussão

4.1 – Etapa de pré-processamento

Como já mencionado no Capítulo 3, as etapas executadas no processo de pré-processamento das bases de dados utilizadas neste trabalho tem como objetivo aumentar a representatividade dos dados utilizados nas classificações, em relação aos dados originais obtidos inicialmente. Além disso, como as seqüências de proteínas coletadas possuem uma grande variação nas suas quantidade de aminoácidos, uma etapa de pré-processamento destas seqüências é crucial para a padronização do número de parâmetros contidos em cada padrão utilizado na etapa de classificação dos mesmos. A Figura 4 exibe um resumo das etapas de pré-processamento realizadas.

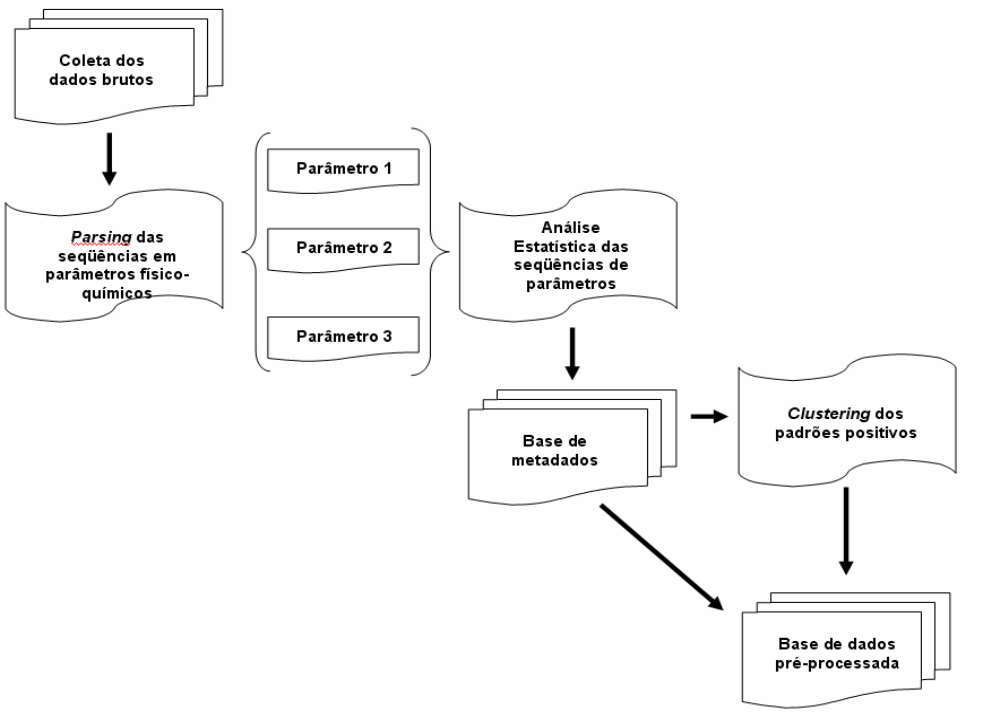

Figura 4 – Etapas de pré-processamento realizadas.

4.2 – Classificação com as redes MLP:

A Tabela 5 e a Tabela 6 exibem os resultados dos experimentos de classificação utilizando as redes MLP, para as bases constando de padrões positivos e negativos ou padrões positivos e de contraste, respectivamente.

Tabela 5 – **Resultados dos experimentos de classificação utilizando MLP, para a base de dados contendo padrões positivos e negativos.**

Nº do experimento	*Taxa de acerto (%)*	*AUC*	*Matriz de confusão*	*Tempo de processamento (min)*
			(Acerto P/Erro P) (Acerto N/Erro N)	
1	92,67	0,9669	(1758/242) (3802/198)	8,00
2	93,02	0,9682	(1777/223) (3804/196)	13,00
3	92,55	0,9674	(1755/245) (3798/202)	8,00
4	92,72	0,9681	(1794/206) (3769/231)	13,00
5	92,81	0,9683	(1772/228) (3797/203)	15,00
6	93,17	0,9698	(1798/211) (3801/199)	27,00
7	92,78	0,9677	(1767/233) (3800/200)	16,00
8	92.83	0.9678	(1779/221) (3791/209)	17,00
9	92,58	0,9670	(1756/244) (3799/201)	8,00
10	92,90	0,9679	(1773/227) (3801/199)	23,00
11	92,73	0,9672	(1762/238) (3802/198)	17,00
12	92,83	0,9687	(1801/199) (3769/231)	27,00
13	92,77	0,9670	(1770/230) (3796/204)	15,00
14	93,23	0,9708	(1790/210) (3804/196)	53,00
15	92,98	0,9687	(1772/228) (3807/193)	15,00
16	**93,47**	**0,9703**	**(1805/195) (3803/197)**	**58,00**

Tabela 6 - **Resultados dos experimentos de classificação utilizando MLP, para a base de dados contendo padrões positivos e de contraste.**

Nº do experimento	*Taxa de acerto (%)*	*AUC*	*Matriz de confusão*	*Tempo de processamento (min)*
			(Acerto P/Erro P) (Acerto C/Erro C)	
1	97,38	0,9952	(2208/86) (3635/71)	27,00
2	97,57	0,9959	(2210/84) (3644/62)	34,00
3	97,48	0,9951	(2215/79) (3634/72)	25,00
4	97,73	0,9950	(2216/78) (3648/58)	45,00
5	97,45	0,9948	(2212/82) (3635/71)	45,00
6	97,72	0,9956	(2213/81) (3650/56)	53,00
7	97,62	0,9941	(2223/71) (3634/72)	33,00
8	**97,78**	**0,9947**	**(2222/72) (3645/61)**	**46,00**
9	97,43	0,9952	(2211/83) (3635/71)	17,00
10	97,62	0,9959	(2210/84) (3647/59)	27,00
11	97,58	0,9950	(2222/72) (3633/73)	17,00
12	97,65	0,9948	(2213/81) (3646/60)	27,00
13	97,43	0,9945	(2214/80) (3632/74)	36,00
14	97,75	0,9954	(2215/79) (3650/56)	47,00
15	97,62	0,9940	(2226/68) (3631/75)	27,00
16	97,67	0,9944	(2215/79) (3645/61)	27,00

Como se pode observar nos resultados obtidos nos dois conjuntos de experimentos, a taxa de acerto média das classificações realizadas foram de 92,88 e 97,59% para as bases contendo padrões negativos e de contraste, respectivamente. O desvio padrão para os dois conjuntos de dados foram de 0,25 e 0,12 para as bases contendo padrões negativos e de contraste, respectivamente. Levando-se em

consideração os valores médios e os desvios padrões dos experimentos realizados, deduz-se que o conjunto de configurações utilizado convergiu para o mínimo global de erro para as duas bases de dados analisadas.

As taxas elevadas de acerto para ambos os conjuntos mostraram-se bastante atraentes, para a metodologia estabelecida, visto que Cay e Lin [10] reportam que a família de proteínas escolhida como padrões positivos possui uma baixa homologia entre as seqüências de suas proteínas, o que, em tese, dificultaria os processos de classificação utilizando estas proteínas, frente a padrões negativos diversos e, principalmente, frente a padrões de contraste. Os valores de taxa de acerto obtidos nos experimentos realizados não encontram-se tão distantes dos valores obtidos em trabalho semelhante na literatura [10]. Entretanto, os parâmetros utilizados neste trabalho de referência possuem uma complexidade de interpretação e obtenção bem maiores do que os parâmetros aqui calculados e obtidos para a realização dos experimentos com as redes MLP. Além disso, obteve-se uma taxa de acerto satisfatória utilizando-se uma quantidade relativamente pequena de parâmetros para representar cada padrão analisado, o que, em termos de processamento de dados, diminui consideravelmente o tempo necessário para a realização das classificações.

Os valores elevados do parâmetro AUC estão correlacionados com a baixa taxa de falso positivos e falso negativos nas classificações realizadas [21]. Em termos de aplicação destas arquiteturas para classificações reais para este tipo de problema, um alto valor de AUC garante uma alta confiabilidade e precisão nas respostas destes classificadores, viabilizando a possibilidade de utilização destes sistemas como ferramenta de suporte a decisão na área de bioinformática.

Os valores satisfatórios das taxas de acerto e de AUC em todos os experimentos realizados podem ser atribuídos a utilização do procedimento de validação cruzada utilizado em todos os experimentos, assim como a boa representatividade dos parâmetros escolhidos para representar os padrões analisados.

Outro parâmetro importante analisado foi o tempo de processamento para a realização dos experimentos. Pode-se observar nas tabelas de resultados que todos os experimentos tiveram um tempo de processamento abaixo de uma hora, sendo

estes tempos satisfatórios para a realização de uma análise exploratória das bases de dados, onde procura-se encontrar a melhor configuração de uma rede MLP para resolver este problema específico de classificação.

4.3 - Classificação com as AD J48:

A Tabela 7 e a Tabela 6 exibem os resultados dos experimentos de classificação utilizando as AD, para as bases constando de padrões positivos e negativos ou padrões positivos e de contraste, respectivamente.

Tabela 7 - **Resultados dos experimentos de classificação utilizando AD, para a base de dados contendo padrões positivos e negativos.**

Nº do experimento	*Taxa de acerto (%)*	*AUC*	*Matriz de confusão*	*Tempo de processamento (min)*
			(Acerto P/Erro P) (Acerto N/Erro N)	
1	87.71	0.9230	(1583/417) (3680/320)	0.18
2	87.92	0.9200	(1599/401) (3676/324)	0.17
3	87.87	0.9190	(1584/416) (3688/312)	0.24
4	87.52	0.9160	(1585/415) (3666/334)	0.21
5	86.72	0.9160	(1539/461) (3664/336)	0.15
6	86.35	0.9140	(1537/463) (3644/356)	0.17
7	86.47	0.9130	(1527/473) (3661/339)	0.15
8	86.63	0.9140	(1540/460) (3658/342)	0.15
9	86.05	0.9030	(1483/517) (3680/320)	0.13
10	85.95	0.9040	(1486/514) (3671/329)	0.12
11	85.83	0.9010	(1506/494) (3644/356)	0.12
12	85.93	0.8940	(1539/461) (3617/383)	0.17

Tabela 8 - **Resultados dos experimentos de classificação utilizando AD, para a base de dados contendo padrões positivos e contraste.**

Nº do experimento	*Taxa de acerto (%)*	*AUC*	*Matriz de confusão*	*Tempo de processamento (min)*
			(Acerto P/Erro P) (Acerto C/Erro C)	
1	96.23	0.9780	(1910/90) (3864/136)	0.11
2	96.15	0.9760	(1908/92) (3861/139)	0.11
3	96.12	0.9760	(1906/94) (3861/139)	0.11
4	95.98	0.9730	(1903/97) (3856/144)	0.11
5	95.80	0.9690	(1884/116) (3864/136)	0.10
6	95.78	0.9690	(1886/114) (3861/139)	0.10
7	95.78	0.9690	(1886/114) (3861/139)	0.15
8	95.75	0.9690	(1878/122) (3867/133)	0.22
9	94.72	0.9650	(1846/154) (3837/163)	0.12
10	94.58	0.9600	(1866/134) (3809/191)	0.12
11	94.58	0.9600	(1866/134) (3809/191)	0.12
12	94.65	0.9590	(1876/124) (3803/197)	0.12

Como pode ser visto nos resultados obtidos nas classificações utilizando AD, embora o tempo de processamento das classificações tenham sido bastante inferiores, quando comparados aos tempos das redes MLP, as taxas de acerto e o valores de AUC apresentaram-se menores.

Isto pode ser devido a forma como o algoritmo de aprendizado da AD utilizada trata os dados apresentados as árvores. Uma explicação plausível é que as redes MLP possuem uma maior facilidade de tratar padrões cuja solução de classificação seja um problema não-linear, enquanto que o tipo de AD utilizado neste trabalho

pode não ter a mesma facilidade de tratamento de problemas não-lineares, o que poderia explicar os menores valores de taxas de acerto e de AUC.

Contudo, levando-se em consideração o tamanho das bases de dados analisadas, assim como a diversidade de padrões presentes nos conjuntos negativo e de contraste, as taxas de acerto obtidas utilizando-se AD ainda são satisfatórias, podendo esta abordagem ser utilizada como uma etapa de pré-processamento para a seleção de atributos, visto que as árvores geradas só possuem em sua estruturas os atributos mais importantes para a realização de suas classificações.

4.4 – Comparação entre as bases de dados:

Comparando-se os resultados de taxas de acerto, em todas as classificações realizadas, em relação as bases de dados utilizadas, pode-se observar que os resultados foram melhores para os experimentos envolvendo padrões positivos com padrões de contraste. Contudo, como já mencionado na Seção 4.1, era de se esperar uma menor taxa de acerto para todos os casos, visto que os padrões positivos possuem uma baixa homogeneidade entre suas proteínas, o que dificultaria sua classificação frente a outros padrões.

Como tentativa de explicar este resultado, realizou-se uma análise de componentes principais (PCA) nas duas bases de dados utilizadas, com o intuito de se observar a variabilidade dos valores referentes aos padrões, nas componentes principais 1 (PC1) e 2 (PC2) da análise. As Figuras 6 e 7 exibem os gráficos de *scores* das análises de PCA realizadas nas bases contendo padrões positivos com negativos e padrões positivos com contraste, respectivamente.

Como pode-se observar nos gráficos exibidos, os padrões positivos e negativos possuem uma maior sobreposição de seus *scores*, enquanto os padrões positivos e de contraste possuem uma menor sobreposição.

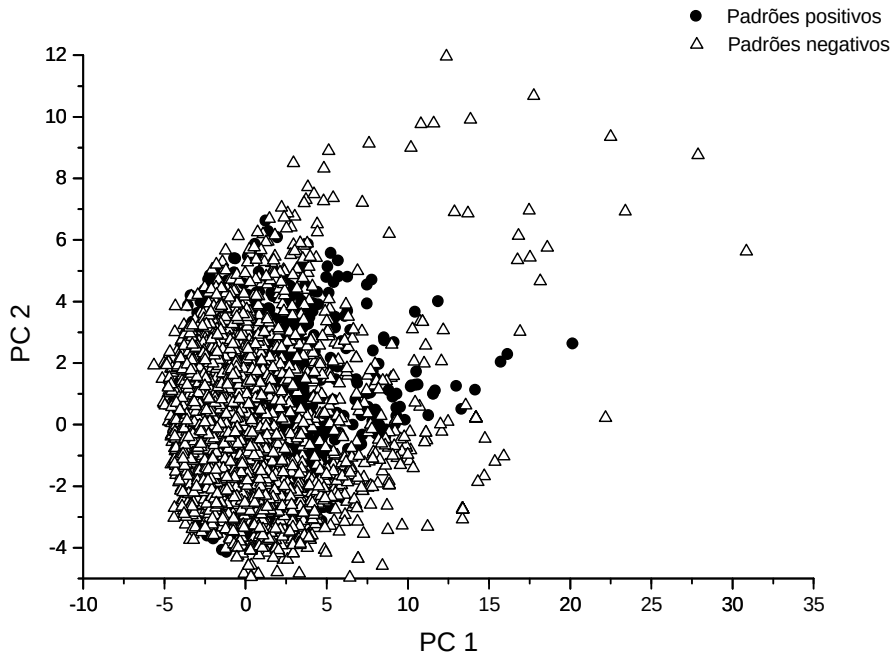

Figura 5 – Análise de PCA, mostrando o gráfico de *scores*, dos padrões positivos e negativos, referentes as componentes PC1 e PC2.

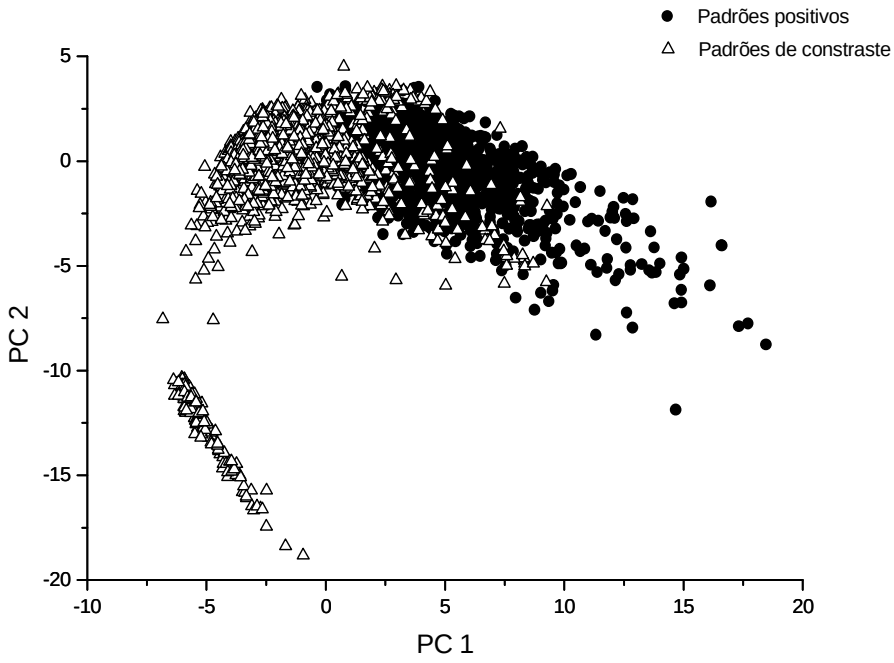

Figura 6 - Análise de PCA, mostrando o gráfico de *scores*, dos padrões positivos e contraste, referentes as componentes PC1 e PC2.

Tendo em vista que a PCA é realizada levando-se em consideração a variabilidade dos parâmetros que representam cada padrão analisado, concluí-se que a base contendo padrões positivos e de contraste possuem uma menor variabilidade interclasse dos parâmetros, isto é, há uma menor sobreposição dos valores destes parâmetros em relação as classes. Isto resulta em uma maior separabilidade dos padrões, resultando em uma maior taxa de acerto de classificação, para os classificadores utilizados neste trabalho. O mesmo não acontecendo para a classe contendo padrões positivos e negativos, visto a maior variabilidade inerente ao conjunto de padrões negativos.

Este resultado pode ser correlacionado a etapa de *clustering* realizada durante o pré-processamento das bases, o que mostra a utilidade deste tipo de metodologia para a melhoria de classificações.

Capítulo 5

Conclusão e Trabalhos Futuros

Apresentou-se neste trabalho uma metodologia simples para a realização de experimentos de classificação com RNAs do tipo *MLP* e *AD*, envolvendo problemas na área de bioinformática. Os resultados obtidos mostraram-se satisfatórios, visto as taxas de acerto e valores de AUC, obtidas com uma metodologia simples desenvolvida para a extração de características. Além disso, as etapas de pré-processamento executadas foram descritas de forma clara e objetiva, visando a utilização deste trabalho como base para a realização de trabalhos futuros de classificação, na área de bioinformática.

Tendo em vista os resultados obtidos e a metodologia empregada, espera-se que este trabalho sirva como marco para o desenvolvimento de novos trabalhos na área de bioinformática, dentro da estrutura do Departamento de Sistemas Computacionais da UPE.

Como trabalhos futuros, propõe-se a utilização de novos parâmetros físico-químicos para a representação dos aminoácidos e comparar os resultados de classificação obtidos com estes novos parâmetros com os parâmetros utilizados neste trabalho, utilizando-se a mesma metodologia de extração de características. Além disso, propõe-se o desenvolvimento de novos parâmetros estatísticos referentes a distribuições de freqüência, assim como validar a metodologia aqui desenvolvida para outras bases de dados referentes a novas classes de proteínas.

Por fim, com a execução de algumas melhorias nas rotinas utilizadas neste trabalho, espera-se que brevemente um sistema automatizado para a classificação de proteínas esteja em funcionamento, alcançando-se a meta proposta para esta monografia.

Bibliografia

[1]. Bergeron, B. **Bioinformatics Computing**. Prentice Hall, 2002.

[2]. Stryer, L. **Bioquímica**. 4ª ed. Guanabara Koogan S. A., Rio de Janeiro, 1996.

[3]. Rost, B., and Sander, C. (1996). **Bridging the protein sequence-structure gap by structure predictions**. *Annu.Rev. Biophys. Biomol. Struct*, 25: 113.

[4]. Wu, C., Whiston, G. et al. (1992) **Protein classification artificial neural system**. *Protein Sci*. 1: 667.

[5]. Wu, C.H. (1997). **Artificial neural networks for sensor neural network approach**. *Proc. Natl. Acad. Sci. USA*, 88: 11261.

[6]. Weston, J., Leslie, C. et al (2005). **Semi-supervised protein classification using cluster kernels**. *Bioinformatics*. 21(15): 3241.

[7]. Ding, C. H. Q. and Dubchak, I. (2001). **Multi-class protein fold recognition using suport vector machines and neuralnetworks**. *Bioinformatics*, 17 (4): 349.

[8]. Enright, A. J., Van Dongen, S. and Ouzounis, C. A. (2002). **An efficient algorithm for large-scale detection of protein families**. *Nucleic Acids Research*, 30(7):1575.

[9]. Murvai, J., Vlahovicek, K., et al. (2001). **Prediction of Protein Functional Domains from Sequences Using Artificial Neural Networks**. *Genome Research*, 11: 1410.

[10]. Cai, Y. e Lin, S. L. (2003). **Support vector machines for predicting rRNA ,RNA and DNA-binding proteins from amino acid sequence**. *Biochimica et Biophysica Acta*, 1648: 127.

[11]. Baldi, P., and Brunak, S. (1998). *Bioinformatics: The machine learning approach*. Cambridge: MIT Press.

[12]. Balakrishnan, N. and Nevzorov, V. B. (2003). *A primer on statistical distributions*. New Jersey, John Wiley & Sons.

[13]. Freyhult, E., Cui, Y., Nilsson, O. and Ardell, D. H. (2007). **New computational methods reveal tRNA identity element divergence between Proteobacteria and Cyanobacteria**. *Biochimie*. 89: 1276.

[14]. Rojas, R. (1996). *Neural Networks*. Berlin, Springer-Verlag.

[15]. Dietterich, T. G. (2000). **An experimental comparasion of three methods for construcing ensembles of decision trees: bagging, boosting and randomization**. *Mach. Learn*. 40: 139.

[16]. Quinlan, J. R. (1986). **Induction of decision trees**. *Mach. Learn*. 1: 81.

[17]. UNIPROT database. URL:http://www.uniprot.org. Acessado em 09/09/2008.

[18]. WEKA Tool. URL: http://www.cs.waikato.ac.nz/ml/weka/. Acessado em 15/09/2008.

[19]. Biojava. URL: http://biojava.org/wiki/Main_Page. Acessado em 12/05/2008.

[20]. About FASTA file format. URL: http://www.proteomecommons.org/ data/fasta/fasta.jsp. Acessado em 15/11/2008.

[21]. Bradley, A. P. (1997). **The use of the área under the ROC curve in he evaluation of machine learning algorithms**. *Patter. Recog*. 30 (7):1145.

Apêndice A

Queries utilizadas para a obtenção das bases de dados

Como já descrito no Capítulo 3, as seqüências de proteínas utilizadas neste trabalho foram extraídas da base de dados UNIPROT [17], utilizando-se as seguintes queries para os padrões negativos e de contraste, respectivamente:

- Padrões negativos: NOT (Activator or adp-ribosylation or chroma-chromatin regulator or chromosomal protein or chromosome partition or core protein or dna damage or dna excision or dna integration or dna packaging or dna priming or dna recombination or dna repair or dna replication or dna replication inhibitor or dna synthesis or dna-directed dna polymerase or dna-directed rna polymerase or endonuclease or excision nuclease or exonuclease or helicase or intron homing or isomerase or mrna processing or mrna splicing or mrna transport or nuclear protein or nuclease or nucleocapsid or nucleoprotein or ribonucleoprotein or ribosomal protein or ribosome biogenesis or rna repair or rna replication or rna-directed dna polymerase or rna-directed rna polymerase or rrna processing or t-dna or topoisomerase or trans-acting factor or transcription or transcription regulation or transcription termination or translation regulation or trna processing or trna-binding or dna-binding or rna-binding or rrna-binding);
- Padrões de contraste: Activator or adp-ribosylation or chroma-chromatin regulator or chromosomal protein or chromosome partition or core protein or dna damage or dna excision or dna integration or dna packaging or dna priming or dna recombination or dna repair or dna replication or dna replication inhibitor or dna synthesis or dna-directed dna polymerase or dna-directed rna polymerase or endonuclease or excision nuclease or exonuclease or helicase or intron homing or isomerase or mrna processing or mrna splicing or mrna

transport or nuclear protein or nuclease or nucleocapsid or nucleoprotein or ribonucleoprotein or ribosomal protein or ribosome biogenesis or rna repair or rna replication or rna-directed dna polymerase or rna-directed rna polymerase or rrna processing or t-dna or topoisomerase or trans-acting factor or transcription or transcription regulation or transcription termination or translation regulation or trna processing or trna-binding or dna-binding or rna-binding

Apêndice B

Formato de dados FASTA

O formato FASTA refere-se a uma formatação textual que serve para representar seqüências de ácidos nucléicos (DNA ou RNA) ou de peptídeos (conjuntos de aminoácidos), através da utilização de códigos únicos para cada componente (ácido nucléico ou aminoácido). Este formato também permite que seqüências depositadas em bases de dados, públicas o ou privadas, sejam precedidas pelos nomes das seqüências e alguns comentários a respeito desta.

De um forma geral, as seqüências escritas no formato FASTA sempre iniciam-se com o símbolo de maior que (“>”) seguido por um cabeçalho com o nome da seqüência e comentários. Abaixo, temos um exemplo de seqüência neste formato:

```
>sp|P00171|CYB5_BOVIN Cytochrome b5 OS=Bos taurus GN=CYB5A PE=1 SV=3
MAEESSKAVKYYTLEEIQKHNNSKSTWLILHYKVYDLTKFLEEHPGGEEVLREQAGGDAT
ENFEDVGHSTDARELSKTFIIGELHPDDRSKITKPSESIITTIDSNPSWWTNWLIPAISA
LFVALIYHLYTSEN
```

Neste caso, por tratar-se de uma proteína, logo após o cabeçalho, temos a seqüência de letras que representam cada aminoácido presente na seqüência. Este é um padrão de formatação aceito mundialmente, o que facilita a troca de informações entre diferentes instituições.

No caso específico das proteínas, temos que o alfabeto utilizado para representar os aminoácidos no formato FASTA é:

Tabela 9 – Código dos aminoácidos na formatação FASTA

Aminoácido	**CÓDIGO**
Alanina	A
Arginina	R
Asparagina	N

Ácido aspártico	D
Cisteína	C
Glutamina	Q
Ácido glutâmico	E
Glicina	G
Histidina	H
Isoleucina	I
Leucina	L
Lisina	K
Metionina	M
Fenilalanina	F
Prolina	P
Serina	S
Treonina	T
Triptofano	W
Tirosina	Y
Valina	V
Ácido aspártico ou Asparagina	B
Ácido glutâmico ou Glutamina	Z
Qualquer aminoácido	X