FAUSTO ROBERTO FERREIRA

O USO DE REDE NEURAL ARTIFICIAL *MLP* NA PREDIÇÃO DE ESTRUTURAS SECUNDÁRIAS DE PROTEÍNAS

Dissertação apresentada ao Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista, Campus de São José do Rio Preto, para a obtenção do título de Mestre em Biofísica Molecular.

Orientador: **Prof. Dr. Jorge Chahine**

Co-Orientadores: **Prof. Dr. José Roberto Ruggiero**
Prof. Dr. Luís Paulo Barbour Scott

São José do Rio Preto
2004

Ao meu Senhor e Salvador,
Jesus Cristo, toda honra e
toda a glória por este trabalho.

AGRADECIMENTOS

Agradeço de todo o meu coração a minha mãe, por todo apoio e dedicação, pois nunca deixou de acreditar em mim.

A minha esposa, por estar comigo nos momentos difíceis.

Ao professor Prof. Dr. Luís Paulo Barbour Scott;

Aos meus professores e amigos.

LISTA DE FIGURAS

Figura 2.1: Ilustração de uma ligação peptídica 7

Figura 2.2: Ilustração de uma estrutura hélice- a 8

Figura 2.3: Folha- ß antiparalela e paralela 8

Figura 2.4: Fstruturas de proteínas 10

Figura 3.1: Neurônio de McCulloch e Pitts 16

Figura 3.2: Funções de ativações 18

Figura 3.3: Exemplos de arquiteturas de RNAs 19

Figura 3.4: Topologia do modelo proposto por Rosenblatt 20

Figura 3.5: Aprendizado supervisionado 22

Figura 3.6: Aprendizado não-supervisionado 23

Figura 3.7: Fluxo de processamento do algoritmo *back-propagation* 24

Figura 4.1: Exemplo do cálculo de escores 32

Figura 4.2: Processo de alinhamento múltiplo 32

Figura 4.3: Interface do *software conversor* 36

Figura 4.4: A representação de como os aminoácidos são codificados 38

Figura 4.5: Arquitetura de uma RNA 40

Figura 4.6: Gráfico do erro de treinamento e validação 41

Figura 4.7: Implementação de duas RNAs 42

Figura 4.8: Arquivo gerado pelo *software* júri de decisão 44

Figura 4.9: Interface do *software* comparador 46

Figura 5.1 comparação dos resultados 55

LISTA DE TABELAS

Tabela 2.1: Tipos de aminoácidos 6
Tabela 4.1: Classificação das estruturas 34
Tabela 4.2: Número de neurônios na camada de entrada das RNAs 38
Tabela 4.3: Codificação binária das estruturas secundárias 39
Tabela 4.4: Arquiteturas de 18 RNAs 43
Tabela 5.1: proteínas do CASP 51
Tabela 5.2: Porcentagem de acerto R1 (uma rede) 52
Tabela 5.3: Porcentagem de acerto R2 (duas redes) 52
Tabela 5.4: resultado do júri de decisão 53
Tabela 5.5: Acerto geral dos preditores 55
Tabela 5.6: Erro total dos preditores 56
Tabela 5.7: Analise da média de acerto 56

LISTA DE EQUAÇÕES

Equação 3.1: Somatória dos pesos 17
Equação 4.1: Coeficientes de acertos 45
Equação 4.2: Coeficientes observados 45
Equação 4.3: Coeficientes preditos 45

SUMÁRIO

CAPÍTULO 1 **1**

Introdução 1
1.1 Considerações Iniciais 1
1.2 Motivação 2
1.3 Objetivos 3
1.4 Organização da tese 4

CAPÍTULO 2 – Fundamentação Teórica **5**

2.1 Considerações Iniciais 5
2.2 Proteínas 5
2.3 Estrutura das Proteínas 7
2.4 Predição de estrutura secundária de proteínas (1D) 10

CAPÍTULO 3 – Redes Neurais Artificiais (RNA) **13**

3.1 Considerações Iniciais 13
3.2 Introdução às redes neurais artificiais 13
3.3 Motivação para as RNAs 15
3.4 Neurônios artificiais 16
3.5 Funções de ativações 17
3.6 Arquitetura das RNAs 18
3.7 *Perceptron* 20
3.8 Redes *Multi Layer Perceptron (MLP)* 21
3.9 Processos de aprendizado de uma RNA 21
3.9.1 Aprendizado supervisionado 22

3.9.2 Aprendizado não-supervisionado 23
3.9.3 Treinamento de redes *MLP* e algoritmo *back-progation* 23
3.9.4 Desenvolvimento e aplicação 25
3.9.5 Configuração da rede 25
3.9.7 Elaboração do treinamento 26
3.9.7 Fase teste 26
3.10 Aplicação de RNAs na predição de estrutura secundária de proteína 27

CAPÍTULO 4 - Materiais e Métodos 30

4.1 Considerações Iniciais 30
4.2 Elaboração da base de dados 31
4.2.1 Alinhamento de seqüências 32
4.3 Base de dados de estruturas secundárias 34
4.4 Interface do pré-processamento 35
4.5 Projeção das arquiteturas das RNAs 37
4.6 Implementação de uma rede neural artificial 39
4.6.1 Treinamento e o critério de parada 41
4.7 Implementação da arquitetura com duas redes 42
4.8 Implementação do júri de decisão 43
4.9 Coeficientes 45

CAPÍTULO 5 - Resultados 48

5.1 Considerações Iniciais 48
5.2 Simulação 1 – Arquitetura de uma RNA 49
5.3 Simulação 2 – Arquitetura com duas RNAs 50
5.4 Resultados com proteínas do CASP 51
5.5 Resultados com o júri de decisão com 18 RNAs 53
5.6 Comparação dos resultados com outros preditores 54
5.6.1 Resultados 55

CAPÍTULO 6 – Conclusões **58**

6.1 Considerações finais 58
6.2 Contribuições do trabalho 59
6.3 Propostas para futuros trabalhos 60

Resumo

A predição de estruturas secundárias e terciárias pode contribuir para elucidar o problema de enovelamento de proteínas. Para isso, métodos de Redes Neurais Artificiais (RNAs) e Algoritmos Genéticos são utilizados a fim de predizê-las, a partir de determinadas seqüências primárias de aminoácidos. Neste sentido, esta pesquisa visa à utilização de três níveis de RNAs. O primeiro nível é composto por um vetor de entrada representando a seqüência primaria dos aminoácidos, com uma dimensão de 22.n, onde n é o tamanho da janela compreendida entre 7 a 23. O segundo nível possui a implementação dos resultados da primeira rede. Por fim o terceiro nível é composto por um júri de decisão. As RNAs são treinadas no Simulador *MATLAB* 5.0, um *software* composto de vários recursos para a sua implementação (*Neural Network Toolbox*). As RNAs implementadas são do tipo *Multi Layer Perceptron* (MLP), que utilizam o algoritmo *backpropagation* (RPROP) e a função de treinamento *trainrp*. Os dados obtidos são comparados com os preditores *'The Predict Protein Server Default' (www.emblheidelberg.de/predictprotein/submit_def.html), 'The PSA Protein Structure Prediction Server' (http//bmerc-www.bu.edu/psa/request.html)* e *'The PSIPRED Protein Structure Prediction Server' (http://bioinf.cs.ucl.ac.uk/psipred/),* a fim de se obter um modelo de predição.

Abstract

The prediction of (secondaray and tertiary) structures of proteins can contribute to elucidadate the protein-folding problem. In oder to predict these structures we used methods of Artificial Neural Network (ANN) and genetic algorithms starting from the primary sequences of amino acids.

The present work is composed of 3 networks levels. The first level is composed of ANNs of an input vector representing a segment of primary amino acid sequence. Since the encoding scheme uses a local window into the sequence, the input vector is a 22.n dimensional vector where n is the number of positions in the window (between 7 and 23). The outputs of level 1 are the inputs of the second level ANNs. The third level is the jury decision.

The ANNs were trained with the Simulator *MATLAB* 5.0, software with several tools for its implementation (Neural *Network Toolbox*). The implemented ANNs are *Multi Layer Perceptron* (MLP) kind, which use the *backpropagation* algorithms (RPROP) together with training function *trainrp*. The obtained date are compared with the predictors 'The Predict Protein Server Default' (www.emblheidelberg.de/predictprotein/submit_def.html), 'The PSA Protein Structure Prediction Server' (http//bmerc-www.bu.edu/psa/request.html) e 'The PSIPRED Protein Structure Prediction Server' (http://bioinf.cs.ucl.ac.uk/psipred/) in order to heve an idea of the quality of the prediction.

Capítulo 1

Introdução

1.1 Considerações Iniciais

Entre as classes de moléculas biológicas de importância para os seres vivos, encontram-se as proteínas, que constituem grande parte da estrutura de todas as células. Essas, além de desempenharem funções catalíticas como a de controle e a de regulação do metabolismo celular, desempenham funções de defesa e de transporte, dentre outras. As funções de uma proteína estão diretamente relacionadas com a sua estrutura nativa. O processo a partir do qual uma proteína sai de uma conformação aleatória e alcança sua estrutura nativa é conhecido como 'enovelamento'. A importância de se conhecer e solucionar o problema do enovelamento de proteínas acaba refletindo-se de forma considerável sobre a biotecnologia.

Ao se estudar o enovelamento de proteínas pretende-se descobrir quais são as propriedades da proteína que levam a cadeia a adotar uma estrutura única e estável e, também, investigar como a seqüência de aminoácidos de uma proteína (estrutura primária) está relacionada com essas propriedades. Esse processo tem objetivo não apenas de elucidar o problema do enovelamento de proteínas, mas também produzir proteínas que possuam uma estrutura desejada. Para isso, alguns métodos como Redes Neurais Artificiais e Algoritmos Genéticos vêm sendo utilizados, a fim de predizer estruturas secundárias e terciárias, respectivamente, a partir de determinadas seqüências (primárias) de aminoácidos.

1.2 Motivação

O número de seqüências de estruturas primárias de proteínas (seqüência de aminoácidos) conhecidas e depositadas em banco de dados cresce mais rápido do que a habilidade de resolver as estruturas terciárias experimentalmente. Mediante essa informação, o objetivo principal é a predição (tri-dimensional) da proteína, contudo não se têm ferramentas que possam predizer a estrutura (3D) de uma proteína através da sua seqüência primária, tendo-se, apenas, passos intermediários, ou seja, predição da estrutura secundária da proteína (1D) e a predição dos contatos de longa distância, ou seja, contatos inter-resíduos (2D) e assim chegar à estrutura (3D).

Contudo, uma predição de sucesso da estrutura secundária (1D) é pré-requisito para uma predição bem sucedida de uma parte de todos os contatos inter-resíduos (predição 2D) (ROST,1998). Para uma proteína fibrosa, a predição de sua estrutura secundária pode obter informações da sua estrutura (tri-dimensional) (3D) (CAMPBELL, 2001). Portanto, técnicas eficazes para predição de estruturas são importantes para diminuir a diferença entre o número de seqüências (primárias) depositadas e estruturas (terciárias) determinadas.

A predição (1D) é um passo intermediário essencial na predição da estrutura terciária completa, ou seja, estrutura nativa de uma proteína. A formação de estruturas secundárias é importante para a estabilidade de uma proteína, a partir de uma estrutura secundária conhecida pode-se obter um número razoavelmente pequeno de possíveis estruturas terciárias (tri-dimensional). Os métodos para predições de estruturas secundárias, a partir dos anos 90, têm sido constantemente alvo de pesquisas para verificar o comportamento das proteínas e suas funcionalidades. As redes neurais artificiais têm sido o método computacional que vem mostrando os melhores resultados nos algoritmos de predição de estruturas secundárias, por possuírem uma característica fundamental que é o reconhecimento de padrões. Essas informações poderão auxiliar em pesquisas que empregam Algoritmos Genéticos no estudo da predição (3D) das proteínas.

1.3 Objetivos

O trabalho de mestrado, descrito nessa dissertação, possui como objetivo o desenvolvimento de um preditor de estrutura secundária de proteína, usando redes com camadas intermediárias *Multi layer Perceptron (MLP)* e três unidades de saída para classificar o resíduo central da janela[1] em hélice-a, folha-ß ou coil (fita aleatória). As redes são projetadas e treinadas no software simulador MATLAB 5.0, a taxa de aprendizagem é calculada pelo algoritmo de treinamento (*back-propagation)*, que faz a comparação dos resultados preditos com os observados. Esse preditor possui como características, três bancos de dados com proteínas do PDB[2] para o treinamento, validação e teste. Esse método dispõe de janelas de resíduos de aminoácidos deferentes, ou seja, janelas de 7,9,11,13,15,17,19,21 e 23, tendo com filtragem a implementação de uma segunda rede para cada janela e dispõe de um júri de decisão para decidir o resultado final da predição e por fim, a comparação dos resultados com outros preditores existentes no mercado, como o *Predict Secondary Structure* (PSIPRED) (JONES, 1999), o *Predict Protein HeiDelberg* (PHD) (ROST, 1993) e a *Protein Sequence Analysis* (PSA) (STULTZ, 1993).

O primeiro deles, o PSIPRED, é um método de predição com rendimento de 76,5 a 78,3% de acerto. Esse método se utiliza o algoritmo *back-propagation* (com uma camada intermediária e outra de saída), possui uma janela de 15 resíduos de aminoácidos, duas redes neurais artificiais com uma camada intermediária e uma unidade de saída para a classificação de hélice-a, folha-ß ou coil (fita aleatória). O PSIPRED apresenta três estágios para a predição das estruturas secundárias:o primeiro estágio baseia-se na generalização do perfil da seqüência, a partir de uma matriz *scoring*; o segundo estágio fundamenta-se na predição inicial da estrutura secundária; o terceiro estágio consiste na filtragem da predição da estrutura secundária da proteína.

[1] Janelas ou janelas de aminoácidos: dado uma seqüência de aminoácidos, no momento do processo é dividido em janelas com tamanhos determinados, de tal forma que todos os resíduos estejam uma vez no centro da janela a fim de identificar sua estrutura secundária através de informação de resíduos adjacentes.

[2] Banco de dados de estruturas terciárias de proteínas. Este banco está disponível na Internet no seguinte site: www.rcsb.org/pdb.

O segundo método citado, o PHD, apresenta um acerto de 69,7 a 70,2%. Esse método utiliza uma rede do tipo *Mult Layer Perceptron* (MLP), com janela de 13 resíduos de aminoácidos, composta de três camadas de unidades (entrada, intermediária e saída) e uma unidade de saída para a classificação hélice-a, folha-ß ou coil (fita aleatória). O PHD possui três níveis de predição: o primeiro deles consiste em uma rede que prediz a estrutura secundária, a partir da seqüência primária; o segundo nível baseia-se em uma rede de filtragem da predição da estrutura secundária, e o terceiro nível é júri de decisão, onde realiza uma média aritmética sobre todos os resultados obtidos.

O terceiro método mencionado, o PSA, utiliza modelagem por homologia, baseado em um software que calcula a probabilidade de determinados resíduos de aminoácido estarem em formação de hélice-a, folha-ß ou coil (fita aleatória). Computa-se a probabilidade desses resíduos usando um grupo de seqüência pré-definido como classe de modelos que chegam a atingir um rendimento de 55 a 60% de acerto.

1.4 Organização da Tese

Essa Tese está organizada em 6 Capítulos. No Capítulo 1 temos, a introdução. No Capítulo 2 faz-se uma discussão sobre estruturas de proteínas e o problema de predição de estruturas protéicas. No Capítulo 3 discute-se, brevemente, sobre os principais conceitos de redes neurais artificiais e alguns métodos de predição de estrutura secundária de proteínas. No Capítulo 4 encontra-se, uma explanação sobre o desenvolvimento e metodologia usada no trabalho. O Capítulo 5 traz os resultados e as discussões. Por fim, o Capítulo 6 trata sobre a conclusão, contribuição e trabalhos futuros.

Capítulo 2

Fundamentação Teórica

2.1 Considerações Iniciais

Neste capítulo serão descritos conceitos necessários para a compreensão do que foi desenvolvido durante o projeto. Inicialmente, serão abordadas as definições relacionadas às proteínas e ao problema de predição de estruturas protéicas.

2.2 Proteínas

As proteínas, quanto a sua composição química, são polímeros de unidades monoméricas, contendo um grupo amina (NH_2), um grupo ácido carboxílico (COOH) e um radical R ligado a um átomo de carbono. Os monômeros constituídos dessa forma são denominados 'aminoácidos' (DILL, 1995; 1990);(MATTHEUS & HOLLEYDE, 1990). As proteínas são compostas por 20 aminoácidos diferentes que se unem por ligações peptídicas formando uma seqüência primária de aminoácidos (ver Tabela1). Ao sair de uma conformação aleatória, as proteínas alcançam sua estrutura nativa, conhecida como 'enovelada'. Com o estudo desse processo, pretende-se descobrir quais as propriedades da proteína que levam sua cadeia a adotar uma estrutura única e estável, e como sua seqüência primária de aminoácidos se relaciona com essas propriedades.

Nome	Símbolo	Massa (-H_2O)	Cadeia lateral	Occurrência (%)	Natureza Química
Alanina	A, Ala	71.079	CH_3-	7.49	Hidrofóbico
Arginina	R, Arg	156.188	$HN=C(NH_2)-NH-(CH_2)_3$-	5.22	Básico
Asparagina	N, Asn	114.104	$H_2N-CO-CH_2$-	4.53	polar
Ácido aspártico	D, Asp	115.089	HOOC-CH_2-	5.22	Ácido
Cisteína	C, Cys	103.145	HS-CH_2-	1.82	Hidrofóbico
Glutamina	Q, Gln	128.131	$H_2N-CO-(CH_2)_2$-	4.11	polar
Ácido glutâmico	E, Glu	129.116	HOOC-$(CH_2)_2$-	6.26	Ácido
Glicina	G, Gly	57.052	H-	7.10	Hidrofóbico
Histidina	H, His	137.141	N=CH-NH-CH=C-CH_2- |_____|	2.23	Básico
Isoleucina	I, Ile	113.160	CH_3-CH_2-$CH(CH_3)$-	5.45	Hidrofóbico
Leucina	L, Leu	113.160	$(CH_3)_2$-CH-CH_2-	9.06	Hidrofóbico
Lisina	K, Lys	128.17	$H_2N-(CH_2)_4$-	5.82	Básico
Metionina	M, Met	131.199	CH_3-S-$(CH_2)_2$-	2.27	Hidrofóbica
Fenilalanina	F, Phe	147.177	Phenyl-CH_2-	3.91	Hidrofóbico
Prolina	P, Pro	97.117	-N-$(CH_2)_3$-CH- |_____|	5.12	Levemente polar
Serina	S, Ser	87.078	HO-CH_2-	7.34	Contém OH
Treonina	T, Thr	101.105	CH_3-CH(OH)-	5.96	Contém OH
Triptofano	W, Trp	186.213	Phenyl-NH-CH=C-CH_2- |_____|	1.32	Hidrofóbico
Tirosina	Y, Tyr	163.176	4-OH-Phenyl-CH_2-	3.25	Contém OH
Valina	V, Val	99.133	CH_3-$CH(CH_3)$-	6.48	Hidrofóbica

Tabela.1: Tipos de aminoácidos

(glu.fcfrp.usp.br/Curso/Modelagem2003/ EstruturaDeProteinas1.pdf).

Na tabela 1, pode-se visualizar o nome específico de cada aminoácido, o seu símbolo representativo, a sua massa, a cadeia lateral que distingue um do outro, a sua porcentagem de ocorrência nas proteínas e a sua natureza química. Snow (SNOW, 1992) define proteína como uma cadeia de aminoácidos ligados covalentemente, entre o grupo amina de um determinado aminoácido e o grupo carboxila do aminoácido seguinte, formando, assim, ligações peptídicas que definem seu esqueleto conforme a sua liberdade rotacional. (ver Figura 2.1):

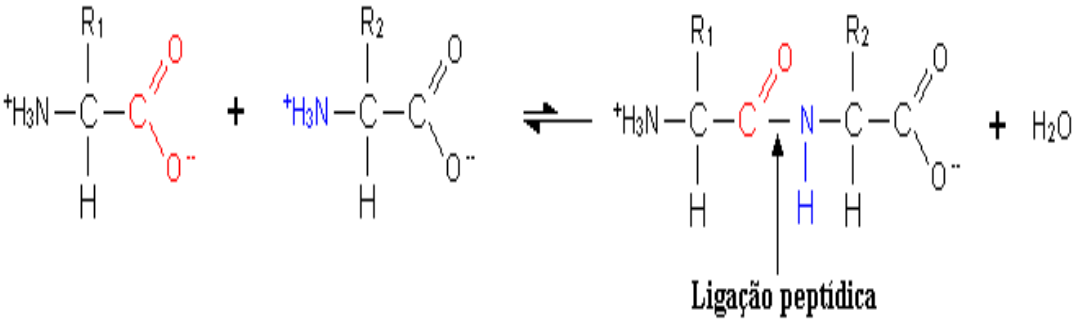

Figura 2.1: Ilustração de uma ligação peptídica (SNOW, 1992).

2.3 Estrutura das proteínas

As proteínas apresentam diferentes níveis de estruturas além da seqüência de aminoácidos que constituem a cadeia polipeptídica (estrutura primária): a estrutura secundaria, terciária e quaternária. A estrutura secundária refere à organização local de partes da cadeia polipeptídica que pode arranjar-se de diferentes formas conhecidas como motifos estruturais. Cada resíduo em uma cadeia polipeptídica possui apenas duas ligações sobre as quais a rotação é permitida nas ligações peptídicas, que permitem as proteínas arranjar-se e formar diferentes estruturas estáveis chamadas hélices-α (ver Figura 2.2), folhas-β (ver Figura 2.3), *coil* e turns[1] que constituem os chamados motivos estruturais que formam a estrutura secundária das proteínas (MATTHEUS, 1990).

[1] *Coil* (fita aleatória) e *turns* (volta): são estruturas irregulares com a não-repetição dos ângulos de torção do esqueleto e, freqüentemente, têm uma ligação de hidrogênio (CHOU & FASMAN, 2000).

Figura 2.2: Ilustração de uma estrutura *hélice-α* (STRYER, 1995)

Figura 2.3: folha- ß antiparalela e paralela (CAMPBELL, 2001)

As hélices-α são os elementos clássicos da estrutura da proteína e são estabilizadas por pontes de hidrogênio paralelas a seu eixo que ocorrem no interior do esqueleto de uma única cadeia polipeptídica (PAULING, 1951). Contando-se a partir da extremidade N-terminal, o grupo C=O de cada resíduo de aminoácido está ligado por pontes de hidrogênio ao grupo N-H do aminoácido posicionado a quatro resíduos adiante, na seqüência linear mantida por ligações covalentes (CAMPBELL, 2001). Assim, a conformação helicoidal permite um arranjo linear dos átomos envolvidos nas pontes de hidrogênio.

Já as folhas-ß diferem-se do arranjo das hélices-α, dado o esqueleto peptídico das folhas-ß estarem quase estendido. Nessas folhas, as pontes de hidrogênio podem ser formadas entre diferentes partes de uma mesma cadeia dobrada sobre si mesma (pontes intracadeia) ou entre diferentes cadeias (pontes intercadeia). Se as cadeias peptídicas estendem-se numa mesma direção, isto é, se todas estiverem alinhadas em suas extremidades N-terminais e C-terminais, será formada uma folha pregueada paralela.

Coil e *turns* são estruturas irregulares com a não-repetição dos ângulos de torção do esqueleto e, freqüentemente, têm uma ligação de hidrogênio (CHOU & FASMAN, 2000). Uma única proteína pode exibir todos os tipos de estrutura secundária, que se arranjam espacialmente, enovelando-se na chamada ‘estrutura terciária (ver Figura 2.4). Quando uma proteína é formada por mais de uma cadeia, suas estruturas terciárias podem arranjar-se, no espaço, formando a estrutura quaternária. Um exemplo conhecido de proteína que possui estrutura quaternária é a Hemoglobina.

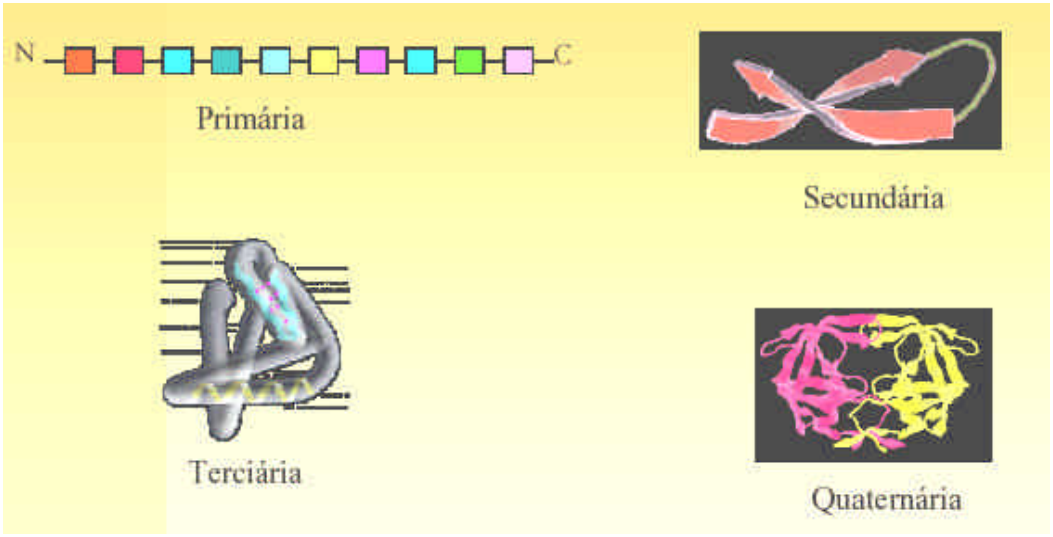

Figura 2.4: Estruturas de proteínas (glu.fcfrp.usp.br/Curso/Modelagem2003/ EstruturaDeProteinas1.pdf).

2.4 Predição de Estrutura Secundária de Proteínas (1D)

Até o momento não se tem conhecimento suficiente para predizer a estrutura terciária (tridimensional) de uma proteína a partir de sua seqüência (estrutura primária). Mas é possível através de alguns métodos predizer aspectos mais simplificados da estrutura (ROST, 1996; 1998). Um aspecto mais simplificado da estrutura seria a estrutura em uma dimensão que são os contatos dos resíduos adjacentes na seqüência (predição da estrutura secundária de proteína ou predição 1D). Alguns métodos utilizam redes neurais artificiais e Algoritmos Genéticos para predizer estruturas secundárias e terciárias a partir de determinadas seqüências primárias de aminoácidos (ROST, 1996; 1998), (SCOTT, 2003). O método de predição de estruturas secundárias de proteínas tem o objetivo de classificar resíduos adjacentes em padrões do tipo H (hélice-α), F (folha-β) e C (*Coil* ou fita aleatória).

Um ponto importante dos métodos de predição de estrutura secundária é o fato de que segmentos de resíduos consecutivos possuem uma preferência por certos estados de estrutura secundária. Então o problema de predição de estrutura torna-se um problema clássico de classificações de padrões, que é tratável por algoritmos de reconhecimento de padrões, por exemplo, redes neurais artificiais.

Os algoritmos destes métodos para predição (1D) podem ser, agrupados da seguinte forma:

- Algoritmos que utilizam informação estática;
- Algoritmos que utilizam propriedades físico-químicas;
- Algoritmos que fazem uso de padrões de seqüências;
- Algoritmos que utilizam redes neurais artificiais;
- Algoritmos que exploram a conservação evolucionária.

Existem algoritmos que combinam idéias, como algoritmos que utilizam redes neurais artificiais e informações evolucionárias (ROST, 1999). Os métodos de predição (1D) podem ser classificados em três categorias: primeira, segunda e terceira geração (ROST, 1996, 1997 e 1998) (BAUCOM,1996).

Os métodos da primeira geração são baseados em propriedades físico-químicas (propensão de certos resíduos estarem formando hélice-a e folha-ß), regras e estatísticas de resíduos isolados, ou seja, não utilizavam informações dos resíduos adjacentes.

Os métodos da segunda geração incorporaram bases de dados maiores e estatísticas, baseadas em segmentos de resíduos adjacentes, tipicamente entre 11 e 21 resíduos. Dessa forma, eles consideram a influência dos resíduos adjacentes sobre o resíduo para o qual a estrutura (1D) foi predita (ROST & SANDER, 1999).

Os métodos da terceira geração são superiores aos métodos da 2ª geração, eles procuram tratar esses três problemas simultaneamente. Possuem como principais características: o fato de utilizar informações evolutivas obtidas a partir de alinhamento múltiplo de seqüências[2], o uso de múltiplos níveis de computação (redes neurais artificiais) e o treinamento balanceado das redes neurais artificiais (ROST & SANDER, 1999). Os métodos de terceira geração são os mais usados na predição de estrutura secundária de proteína através da utilização de redes neurais artificiais, possuindo um acerto considerável em relação aos outros métodos. A identificação com sucesso da estrutura secundária (predição 1D) é um pré-requisito para uma predição bem sucedida de uma parte de todos os contatos inter-resíduos (predição 2D) (ROST, 1998). Contudo, contatos que foram preditos a partir de associação de estrutura secundária são de curto alcance, isto é, entre resíduos próximos na seqüência. É necessário predizer também os contatos de longo alcance, entre resíduos distantes na seqüência. Portanto, a predição (1D) é um passo importante para se obter uma predição 2D confiável. Essas informações poderão auxiliar em pesquisas que empregam Algoritmos Genéticos no estudo da predição 3D das proteínas. O capítulo 3 descreve os conceitos básicos de Redes Neurais Artificiais.

[2] O alinhamento de seqüências faz uma associação explícita entre resíduos de duas ou mais seqüências. Um dos objetivos do alinhamento é determinar a homologia (similaridade) entre seqüências.

Capítulo 3

Redes Neurais Artificiais (RNA)

3.1 Considerações Iniciais

O estudo de redes neurais artificiais é um campo extremamente indisciplinar tanto em relação ao seu desenvolvimento como em relação a sua aplicação. Alguns exemplos de aplicação de redes neurais: a predição de estrutura secundária de proteínas (ROST, 1998); o reconhecimento de caracteres (DENKER, 1995); o diagnóstico médico (BURKE, 1995); a previsão de ações na bolsa (YODA, 1994); a segurança em transações com cartões de crédito (REATEGUI, 1994) e o reconhecimento de assinaturas (MIGHELL1988). Este capítulo tem por objetivo discutir os conceitos básicos de redes neurais artificiais, e suas formas de treinamento e aprendizagem.

3.2 Introdução às Redes Neurais Artificiais

As redes neurais artificiais consistem em um método de solucionar problemas de inteligência artificial, construindo um sistema que tenha circuitos que simulem o cérebro humano, inclusive seu comportamento, ou seja, aprendendo, errando e fazendo descobertas. São mais que isso, são técnicas computacionais que apresentam um modelo inspirado na estrutura neural de organismos

inteligentes e que adquirem conhecimento através da experiência. Apesar da complexidade das redes neurais não permitir uma única definição, as linhas seguintes seguem como uma tentativa das inúmeras definições ou interpretações do que seja realmente uma rede neural. Um grafo[1] direcionado é um objeto geométrico que consiste de um conjunto de pontos, chamados nós, ao longo de um conjunto de segmentos de linhas direcionadas entre eles.

As redes neurais são, estruturas de processamentos de informação distribuída paralelamente na forma de um grafo direcionado, com algumas restrições e definições próprias. Os nós deste grafo são chamados elementos de processamento. Suas arestas são conexões, que funcionam como caminhos de condução instantânea de sinais em uma única direção, de forma que seus elementos de processamento podem receber qualquer número de conexões de entrada. Estas estruturas podem possuir memória local, e também possuir qualquer número de conexões de saída desde que os sinais nestas conexões sejam os mesmos.

Portanto, estes elementos têm na verdade uma única conexão de saída, que pode dividir-se em cópias para formar múltiplas conexões, sendo que todos carregam o mesmo sinal. Então, a única entrada permitida para a função de transferência[2] (que cada elemento de processamento possui) são os valores armazenados na memória local do elemento de processamento e os valores atuais dos sinais de entrada nas conexões recebidas pelo elemento de processamento. Os únicos valores de saída permitidos a partir da função de transferência são valores armazenados na memória local do elemento de processamento, e o sinal de saída do mesmo. A função de transferência pode operar continuamente ou episodicamente. Sendo que no segundo caso, deve existir uma entrada chamada "ativada" que causa o ativamento da função de transferência com o sinal de entrada corrente e com valores da memória local, e produzir um sinal de saída atualizado (ocasionalmente alterando valores da memória).

[1] Grafo: conjunto cujos elementos são unidos por arcos.
[2] Função de ativações: são funções matemáticas de intervalos determinados onde para transferência de informação de um neurônio a outro na camada seguinte.

E no primeiro caso, os elementos estão sempre ativados, e a entrada "ativada" chega através de uma conexão de um elemento de processamento agendado que também é parte da rede. Sinais de entrada para uma rede neural a partir de fora da rede chegam através de conexões que se originam do mundo externo, saídas da rede para o mundo externo são conexões que deixam a rede. De forma geral, a operação de uma célula da rede se resume em: sinais são apresentados à entrada; cada sinal é multiplicado por um peso que indica sua influência na saída da unidade; é feita a soma ponderada dos sinais que produz um nível de atividade; se este nível excede um limite (threshold) a unidade produz uma saída.

3.3 Motivação para as RNAs

A partir do momento em que as máquinas começaram a evoluir um dos grandes desejos do homem tem sido a criação de uma máquina que possa operar independentemente do controle humano. Uma máquina cuja independência seja desenvolvida de acordo com seu próprio aprendizado e que tenha a capacidade de interagir com ambientes incertos (desconhecidos por ela), uma máquina que possa ser chamada de autônoma, inteligente ou cognitivo[3]. O sucesso de uma máquina autônoma dependeria única e exclusivamente de sua capacidade de lidar com uma variedade de eventos inesperados no ambiente em que opera. Estas máquinas teriam maior capacidade de aprender tarefas de alto nível cognitivo que não são facilmente manipuladas por máquinas atuais. Elas continuariam a se adaptar e realizar tarefas gradativamente com maior eficiência, mesmo que em condições de ambiente imprevisíveis. Organismos humanos são uma fonte de motivação para o desenvolvimento destas máquinas e proporcionam diversas dicas para o desenvolvimento de algoritmos de aprendizado e adaptação . Assim, espera-se que algumas das características de organismos biológicos de aprendizado e adaptação estejam presentes nas mesmas. Enquanto

[3] Cognitivo:relativo ao processo mental de conhecimento, memória, juízo ou raciocínio.

computadores funcionam de modo seqüencial, proporcionando maior eficiência na resolução de tarefas nas quais devem ser seguidas etapas. O cérebro humano funciona de modo paralelo, e sendo extremamente conectado é mais eficiente na resolução de tarefas que exigem várias variáveis. O motivo pelo quais máquinas inspiradas na biologia são diferentes das máquinas atuais se encontra no fato de que as máquinas atuais baseiam seu processamento explicitamente em modelos matemáticos.

Mecanismos de controle baseado em mecanismos neurais entretanto, não são baseados em modelos, utilizam cálculos matemáticos para efetuar suas operações porém podem coordenar diversos graus de liberdade durante a execução de tarefas manipulativas e em ambientes desestruturados. Eles são capazes de lidar com tarefas complicadas sem que tenham que desenvolver um modelo matemático e nem um modelo do ambiente em que operam. Baseado nas características de seres biológicos, acredita-se que surgirá em um futuro próximo, uma geração completa de novos sistemas computacionais, muito mais eficientes e inteligentes que os sistemas atuais.

3.3 Neurônios Artificiais

O Neurofisiologista McCulloch e matemático Walter Pitts (1943), cujo trabalho fazia uma analogia entre células vivas e o processo eletrônico, simulando o comportamento do neurônio natural, onde o neurônio possuía apenas uma saída, que era uma função de entrada (threshold) da soma do valor de suas diversas entradas x_1, x_2, ...,x_n, que representam os dendritos e um terminal de saída y representado pelo axônio. Para imitar o comportamento das sinapses, os terminais de entrada do neurônio têm pesos acoplados w_1, w_2, ...,w_n, cujos valores podem ser positivos ou negativos, dependendo de se essas sinapses são inibitórias ou excitatórias. (ver figura 3.1)

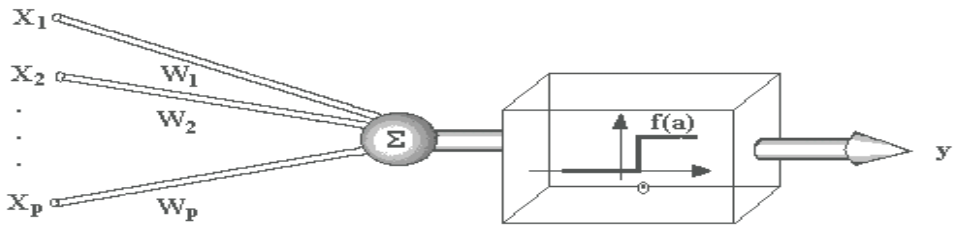

Figura 3.1: Neurônio de McCulloch e Pitts (1943).

Um neurônio biológico dispara quando a soma dos impulsos que ele recebe ultrapassa o seu limiar de excitação (*threshold*). O corpo do neurônio, por sua vez, é estimulado por um mecanismo simples que faz a soma dos valores $x_i w_i$ recebidos pelo neurônio (soma ponderada), e decide se o neurônio deve ou não disparar (saída igual a 1 ou 0) comparando a soma obtida ao limiar do neurônio. No modelo MCP, a ativação do neurônio é obtida através da aplicação de uma "função de ativação", que ativa a saída ou não, dependendo do valor da soma ponderada das suas entradas (PITTS, 1943). Na descrição original do modelo MCP, a função de transferência é dada pela função de limiar descrita na equação 3.1. O neurônio MCP terá então a sua saída ativa quando:

$$\sum_{i=1}^{n} x_i w_i \geq q \quad \text{para i = 1,2, ...,n} \qquad \text{Eq. (3.1)}$$

onde n é o número de entradas do neurônio, w_i é o peso associado à entrada x_i, e q é o limiar (*threshold*) do neurônio. A partir do modelo MCP proposto por McCulloch e Pitts foram derivados vários outros modelos que permitem a produção de uma saída qualquer, não necessariamente zero ou um, e com diferentes funções de ativações.

3.4 Funções de ativações

As operações básicas de uma RNA envolvem somar os sinais que entram multiplicados pelos pesos e aplicar uma função de transferência para determinar a saída do neurônio. Para a camada de entrada de dados a função de transferência geralmente é a função identidade (linear) e as demais camadas utilizam outras funções de transferência. A Figura 3.2 ilustra graficamente quatro funções de ativações diferentes, são elas: a função linear (a), a função rampa (b), a função degrau (c) e a função sigmoidal (d).

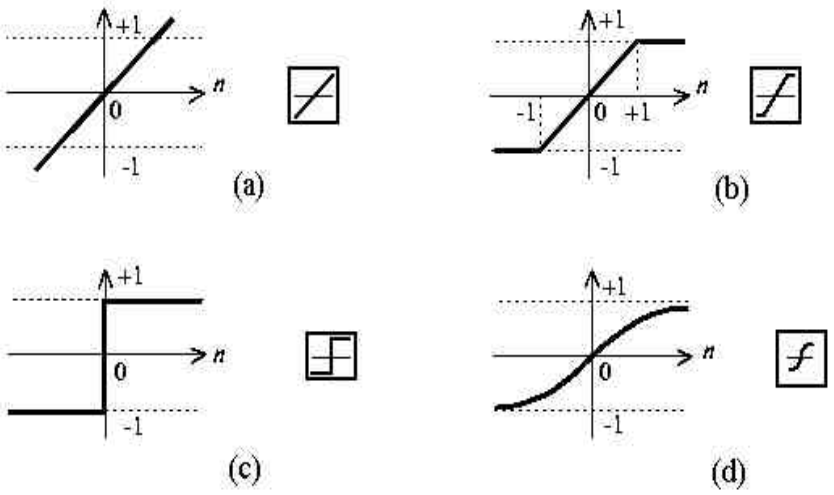

Figura 3.2: Algumas funções de ativação (BRAGA, 1998).

Na figura 3.2a temos a função de transferência linear definida pela equação **y= a. x**, onde **a** é um número real que define a saída linear para valores da entrada **y**. Na figura 3.2 b, temos uma função linear cujo objetivo é produzir valores na faixa de –1 a 1conhecida por função rampa. Na figura 3.2 c conhecida como função degrau ou função sinal, o sinal é transferido quando os valores forem maiores que zero atribuindo o valor 1 e valores menores que zero atribui valor –1. Na figura 3.2 d temos a função de transferência simoidal, essa função é semilinear onde os valores atribuídos compreende-se entre –1 a 1.

3.6 Arquiteturas das RNAs

A arquitetura de uma rede consiste na forma como os neurônios estão arranjados e conectados em camadas (BRAGA,1998). Alguns parâmetros são fundamentais na definição da arquitetura. São eles: o número de camadas da rede, o número de neurônios em cada camada, o tipo de conexão entre os neurônios e a topologia da rede. Alguns exemplos de arquiteturas são apresentados na figura 3.3.

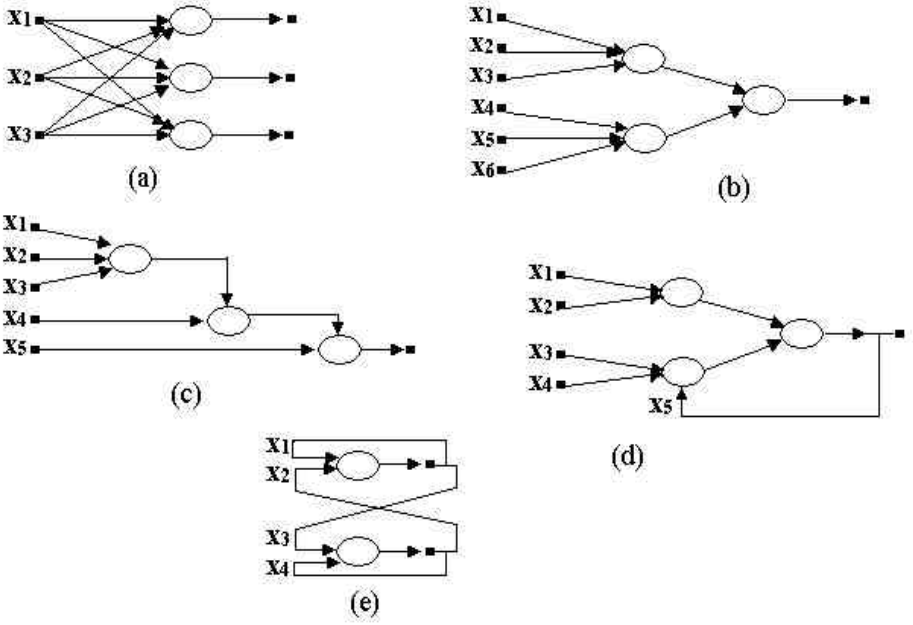

Figura 3.3: Exemplos de arquiteturas de RNAs .

Na ilustração da figura 3.3 temos, as arquiteturas que podem conter todos os neurônios conectados e a recorrência de sua propagação, nas chamadas 'redes recorrentes'. Quanto ao número de camadas, essas apresentam: redes de camada única, em que só há um nó entre quaisquer entradas e saídas da rede (Figura 3.3 (a) e (e)); Redes de múltiplas camadas, em que há mais de um neurônio entre quaisquer entradas e saídas da rede (Figura 3.3 (b), (c) e (d)).

Com relação a suas conexões, os neurônios apresentam: conexão acíclica ou *feed-forward*, em que a saída de um neurônio na *i*-ésima camada não pode ser usada como entrada naquelas de índice menor ou igual a *i*. Finalmente, as RNAs podem ser classificadas quanto a sua conectividade: rede parcial ou fracamente conectada (Figura 3.3 (b), (c) e (d)); Rede completamente conectada (Figura 3.3 (a) e (e)).

3.7 *Perceptron*

O conceito de aprendizado foi introduzido em RNAs com o trabalho de Rosenblatt (ROSENBLATT, 1958), a partir da proposição de um modelo conhecido como *perceptron*[4]. Posteriormente, em 1962, Rosenblatt demonstra o teorema de convergência do *perceptron*, ao mostrar que um neurônio MCP treinado com o algoritmo de aprendizado do *perceptron* sempre converge para uma solução considerada ótima, caso o problema em questão seja linearmente separável. A topologia descrita pelo autor compõe-se de: unidades de entrada ou retina; por um nível intermediário formado pelas unidades de associação e por um nível de saída formado pelas unidades de respostas (ver figura 3.4).

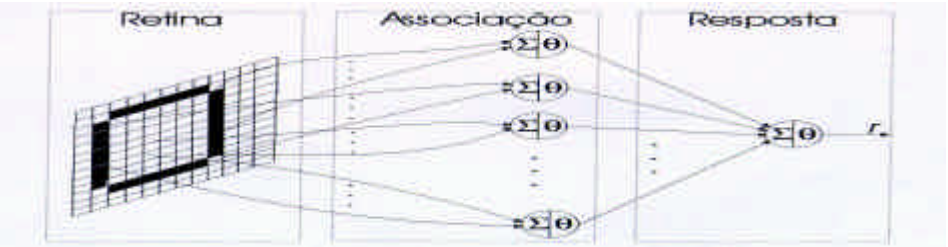

Figura 3.4: Topologia do modelo proposto por Rosenblatt (BRAGA, 1998).

Como ilustrado na figura 3.4, ainda que essa topologia possua três níveis, é conhecida como *perceptron* de uma única camada, visto que somente o nível de saída ou unidade de resposta possui propriedades adaptativas. A retina consiste em unidades sensoras, sendo que as unidades intermediárias de associação,

[4] Perceptron: compõe de uma estrutura de rede com os neurônios MCP, como unidades básicas, e de uma regra de aprendizado.

embora sejam formadas por neurônios MCP, possuem pesos fixos, definidos antes do período de treinamento. Apesar de ter representado um marco na comunidade científica, a aparição do *perceptron* foi efêmera, devido ao fato de ter sofrido críticas com relação a sua capacidade computacional. A mudança desse pessimismo sobre a capacidade do *perceptron* e das RNAs de uma maneira geral só aconteceu nos anos 80 com o surgimento do algoritmo *back-propagation* e das redes *Multi Layer Perceptron* (Redes MLP).

3.8 Redes *Multi Layer Perceptron*

As Redes *Multi Layer perceptron* são formas de projetar redes *perceptrons* em camadas. O *multi layer perceptron* foi concebido para resolver problemas mais complexos, os quais não poderiam ser resolvidos pelo modelo de neurônio básico. Os neurônios internos são de suma importância na rede neural pois se provou que sem estes se torna impossível à resolução de problemas linearmente não separáveis. Em outras palavras pode-se dizer que uma rede é composta por várias unidades de processamento, cujo funcionamento é bastante simples. Essas unidades, geralmente são conectadas por canais de comunicação que estão associados a determinado peso. As unidades fazem operações apenas sobre seus dados locais, que são entradas recebidas pelas suas conexões. A maioria dos modelos de redes neurais possui alguma regra de treinamento, onde os pesos de suas conexões são ajustados de acordo com os padrões apresentados. A rede neural passa por um processo de treinamento a partir dos casos reais conhecidos, adquirindo, a partir daí, a sistemática necessária para executar adequadamente o processo desejado dos dados fornecidos.

3.9 Processos de Aprendizado de uma RNA

A propriedade mais importante das redes neurais é a habilidade de aprender de seu ambiente e com isso melhorar seu desempenho. Isso é feito através de um processo iterativo de ajustes aplicado a seus pesos, o treinamento. O aprendizado ocorre quando a rede neural atinge uma solução generalizada para

uma classe de problemas. Denomina-se algoritmo de aprendizado um conjunto de regras bem definidas para a solução de um problema de aprendizado. Existem muitos tipos de algoritmos de aprendizado específicos para determinados modelos de redes neurais, estes algoritmos diferem entre si principalmente pelo modo como os pesos são modificados. A rede neural se baseia nos dados para extrair um modelo geral. Portanto, a fase de aprendizado deve ser rigorosa e verdadeira, a fim de se evitar modelos espúrios[5].

Todo o conhecimento de uma rede neural está armazenado nas sinapses, ou seja, nos pesos atribuídos às conexões entre os neurônios. De 50 a 90% do total de dados deve ser separado para o treinamento da rede neural, dados estes escolhidos aleatoriamente, a fim de que a rede "aprenda" as regras e não "decore." O restante dos dados só é apresentado à rede neural na fase de testes a fim de que ela possa "deduzir" corretamente o inter-relacionamento entre os dados. Diversos métodos para treinamento de redes foram desenvolvidos, podendo estes ser agrupados em dois conjuntos principais: Aprendizado Supervisionado e Aprendizado Não-Supervisionado.

3.9.1 Aprendizado Supervisionado

Este método de aprendizado é bastante utilizado no treinamento das RNAs, tanto de neurônios com pesos, como de neurônios sem pesos, sendo chamado aprendizado supervisionado porque a entrada e saída desejadas para a rede são fornecidas por um supervisor (professor) externo. O objetivo é ajustar os parâmetros da rede, de forma a encontrar uma ligação entre os pares de entrada e saída fornecidos. A figura 3.5 ilustra o mecanismo de aprendizado supervisionado.

[5] Espúrio: não genuíno.

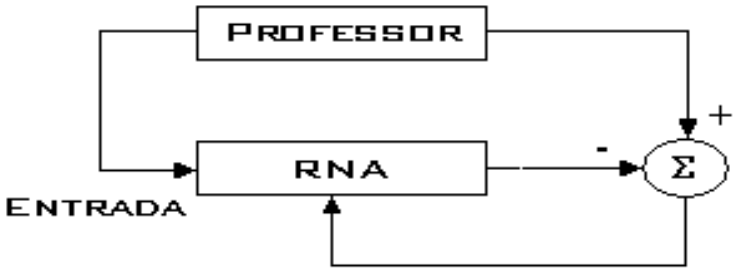

Figura 3.5 – Aprendizado Supervisionado (BRAGA, 1998).

3.9.2 Aprendizado Não-Supervisionado

No aprendizado não-supervisionado, como o próprio nome informa, não há um professor ou supervisor para acompanhar o processo de aprendizado (BRAGA, 1998). Para estes algoritmos, somente os padrões de entrada estão disponíveis para a rede, ao contrário do aprendizado supervisionado, cujo conjunto de treinamento possui *pares* de entrada e saída. Este método é ilustrado na figura 3.6.

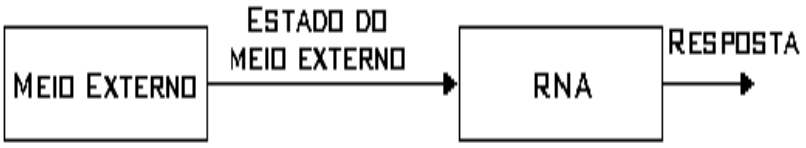

Figura 3.6 – Aprendizado Não-Supervisionado (BRA1998).

3.9.3 Treinamento de Redes *MLP* e o Algoritmo *Back-Propagation*

De acordo com Rumelhart (RUMELHART, 1986), Battiti, (BATTITI, 1991) e Hagan (HAGAN, 1994), existem, atualmente, vários algoritmos para treinar redes *MLP*, do tipo supervisionado e são classificado em estático e dinâmico. Enquanto os algoritmos estáticos não alteram a estrutura da rede, variando apenas os valores de seus pesos, os algoritmos dinâmicos podem tanto reduzir quanto

aumentar o seu tamanho. Quando o aprendizado estático é utilizado, a mesma regra é empregada para redes do tipo *MLP* com diferentes tamanhos e formatos. É interessante observar que topologias diferentes podem resolver o mesmo problema. Segundo Rumelhart, o algoritmo de aprendizado mais conhecido para treinamento dessas redes é o algoritmo *back-propagation.* O treinamento desse algoritmo ocorre em duas fases, conhecidas como *forward* e *backward.* Cada uma dessas fases percorre a rede em um sentido (ver figura 3.7).

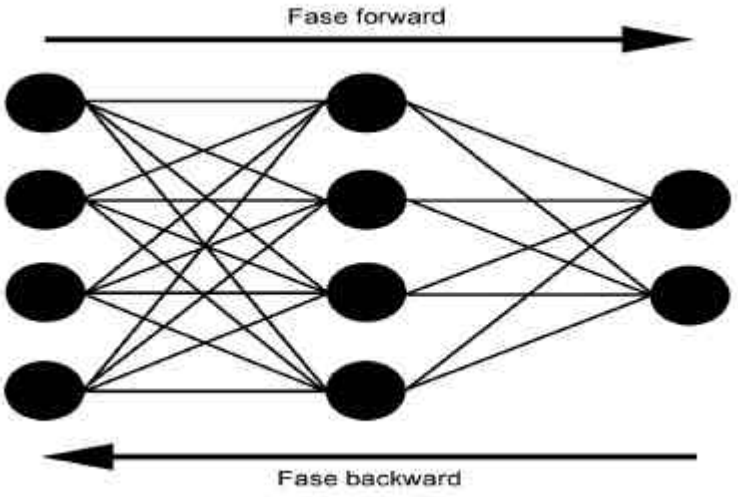

Figura 3.7 : Fluxo de processamento do algoritmo *back-propagation* (BRAGA, 1998).

A figura 3.7 apresenta as fases *forward* e *backward.* Na primeira, tem-se a propagação dos dados de entrada, em que são calculados os pesos até se chegar próximo do valor desejado (por exemplo: saída desejada igual a 1. Então valores processados pela rede só serão resposta ideal quando forem bem próximo de 1, por exemplo 0,987, assim o erro esperado é de aproximadamente 0,01). Na segunda, se o erro for acima do objetivo, ocorre a volta da propagação, isto é, o algoritmo recalcula os pesos até se chegar perto do desejável. Na fase *forward*, a informação é apresentada à primeira camada da rede, conhecida como camada de entrada. Essa informação é passada à próxima camada, intermediária, e, assim, sucessivamente. Nesse processo, as saídas produzidas pelos neurônios da

última camada são comparadas às saídas desejadas. A fase *backward* realiza o processo da volta, a partir da última camada até à camada de entrada. Os neurônios da camada atual ajustam os seus pesos de forma a reduzir seus erros. O erro de um neurônio das camadas intermediárias é calculado utilizando os erros dos neurônios da camada seguinte a ele conectados, ponderado pelos pesos das conexões entre eles.

3.9.4. Desenvolvimento e aplicação

Nesse tópico serão ilustrados os passos necessários para o desenvolvimento de aplicações utilizando redes neurais artificiais. Primeiro temos, a Coleta de dados e segundo separação em conjuntos. Os dois primeiros passos do processo de desenvolvimento de redes neurais artificiais são a coleta de dados relativos ao problema e a sua separação em um conjunto de treinamento e um conjunto de testes. Esta tarefa requer uma análise cuidadosa sobre o problema para minimizar ambigüidades e erros nos dados. Além disso, os dados coletados devem ser significativos e cobrir amplamente o domínio do problema; não devem cobrir apenas as operações normais ou rotineiras, mas também as exceções e as condições nos limitem do domínio do problema.

Normalmente, os dados coletados são separados em duas categorias: dados de treinamento, que serão utilizados para o treinamento da rede e dados de teste, que serão utilizados para verificar sua performance sob condições reais de utilização. Além dessa divisão, pode-se usar também uma subdivisão do conjunto de treinamento, criando um conjunto de validação, utilizado para verificar a eficiência da rede quanto a sua capacidade de generalização durante o treinamento, e podendo ser empregado como critério de parada do treinamento. Além disso, pode ser necessário pré-processar estes dados, através de normalizações, escalonamentos e conversões de formato para torná-los mais apropriados à sua utilização na rede.

3.9.5 Configuração da rede

O terceiro passo é a definição da configuração da rede, que pode ser dividido em três etapas: determinação da topologia da rede a ser utilizada - o número de camadas, o número de unidades em cada camada, etc; determinação de parâmetros do algoritmo de treinamento e funções de ativações – o intervalo da função e o tipo de algoritmo de treinamento. Normalmente estas escolhas são feitas de forma empírica. A definição da configuração de redes neurais é ainda considerada uma arte, que requer grande experiência dos projetistas.

3.9.6 Elaboração do treinamento

O quarto passo é o treinamento da rede. Nesta fase, seguindo o algoritmo de treinamento escolhido, serão ajustados os pesos das conexões. É importante considerar, nesta fase, alguns aspectos tais como a inicialização da rede, o modo de treinamento e o tempo de treinamento. Uma boa escolha dos valores iniciais dos pesos da rede pode diminuir o tempo necessário para o treinamento. Normalmente, os valores iniciais dos pesos da rede são números aleatórios uniformemente distribuídos, em um intervalo definido. A escolha errada destes pesos pode levar a uma saturação prematura.

Quanto ao tempo de treinamento, vários fatores podem influenciar a sua duração, porém sempre será necessário utilizar algum critério de parada. O critério de parada do algoritmo backpropagation pode ser definido como o número máximo de ciclos. Mas, devem ser consideradas a taxa de erro médio por ciclo, e a capacidade de generalização da rede. Pode ocorrer que em um determinado instante do treinamento a generalização comece a degenerar, causando o problema de over-training, ou seja a rede se especializa no conjunto de dados do treinamento e perde a capacidade de generalização. O treinamento deve ser interrompido quando a rede apresentar uma boa capacidade de generalização e quando a taxa de erro for suficientemente pequena, ou seja menor que um erro

admissível. Assim, deve-se encontrar um ponto ótimo de parada com erro mínimo e capacidade de generalização máxima.

3.9.7 Teste

O quinto passo é o teste da rede. Durante esta fase o conjunto de teste é utilizado para determinar a performance da rede com dados que não foram previamente utilizados. A performance da rede, medida nesta fase, é uma boa indicação de sua performance real. Devem ser considerados ainda outros testes como análise do comportamento da rede utilizando entradas especiais e análise dos pesos atuais da rede, pois se existirem valores muito pequeno, as conexões associadas podem ser consideradas insignificantes e assim serem eliminadas. De modo inverso, valores substantivamente maiores que os outros poderiam indicar que houve over-training da rede.

3.9.8 Aplicação de RNAs na Predição da Estrutura Secundária de proteínas

Entre as possíveis aplicações de RNAs no estudo de estruturas de moléculas como proteínas estão: predição de estruturas secundárias através de classificação/reconhecimento de padrões; predição de estruturas terciárias de proteínas através de otimização de uma função potencial de energia; predição de possíveis seqüências de aminoácidos para uma dada proteína, de forma a obter as conformações de mais baixa energia; predição de seqüências de aminoácidos (estruturas primárias) que levem a uma estrutura secundária ou terciária desejada.

Outros aspectos de estruturas de proteínas, tal como a classe estrutural, também pode ser predita utilizando RNAS. Pode-se utilizar redes neurais para associar as proteínas a uma das quatro classes (todas, ou seja, todas as classes em conjunto único, hélices-α, folhas-β e outras, todas folhas-β, todas hélices-α) com uma precisão que pode chegar em alguns trabalhos publicados em 75%. Um dos primeiros trabalhos que utilizou redes neurais na predição de estruturas

secundárias de proteínas a partir de seqüências primárias foi dos pesquisadores Holley e Karphus (HOLLEY & KARPLUS, 1991).

Eles utilizaram RNA do tipo *Multi Layer Perceptron* (MLP) e codificaram os dados de entrada da rede em janelas de resíduos adjacentes. Para cada resíduo, há 21 entradas binárias, ou seja, cada aminoácido foi codificado com 21 unidades. Sendo 20 unidades representando os aminoácidos e 1 unidade representando o heteroátomo. A RNA utilizada possuía uma camada intermediária com duas unidades e uma camada de saída também com duas unidades.

Holley e Karplus utilizaram um conjunto de dados de 48 proteínas para treinar e 14 proteínas para teste. Eles testaram vários tamanhos de janela e a que mostrou os melhores resultado foi à janela de 17 resíduos. Também foram testadas RNA com diferentes tamanhos de camadas intermediárias (variando as camadas de 2 a 20). Apesar da rede com a camada intermediária contendo 20 unidades ter apresentado o melhor resultado para o conjunto de treinamento, a rede com a camada intermediária com 2 unidades apresentou melhor resultado para o conjunto de teste. O desempenho obtido para o treinamento foi de 68,5% e para o de teste foi de 63,2%.

Após Holley e Karplus surgiram outras pesquisas com alguns resultados significativos. Chandonia e Karplus (CHANDONIA & KARPLUS, 1996) aplicaram duas redes neurais, denominadas primária e secundária, e foram utilizados um conjunto de 681 proteínas com estruturas disponíveis no Protein Data *Bank (PDB)*[5]. A rede primária utilizada por Chandonia e Karplus foi similar à rede da pesquisa descrita anteriormente, com 21 unidades de entrada que representam o tipo de aminoácido, sendo o último representando o fim da cadeia. A saída desta rede possui 2 unidades que correspondem à estrutura secundária, hélice-a e folha-ß. A rede neural secundária é utilizada para refinar os resultados produzidos pela primeira rede, classificando as proteínas em classes: todas, conjunto que possui todas as classes: folhas-β, hélices-α e coil (fita aleatória), todas folhas-β, conjunto que possui somente proteínas com características folhas-β e todas

[5] Banco de dados de estruturas terciárias de proteínas. Este banco está disponível na Internet no seguinte *site*: www.rcsb.org/pdb

hélices-α conjunto que possui somente proteínas com características hélices-α. Nessa segunda rede, os dados de entrada são os resultados da primeira rede, hélices-α e folhas-β, para cada resíduo.

Já os pesquisadores Rost e Sander (ROST & SANDER, 1994) utilizaram uma RNA do tipo MLP, com duas camadas intermediárias, sobre uma base de dados não redundante de 130 proteínas. Um elemento chave neste trabalho foi o uso de informações evolucionárias[6] obtidas por alinhamentos múltiplos de seqüências que são utilizadas como entrada ao invés de seqüências simples. Esse foi um dos primeiros trabalhos a incluir informação evolucionária para auxiliar as redes neurais na predição de estruturas e apresentou um aumento de 6% a 8% na capacidade da predição. A combinação de três níveis de redes resultou em uma taxa de acerto de 70.8%. O trabalho utilizou três redes neurais de forma que a primeira rede utiliza 20 unidades de entrada para cada resíduo e uma janela de 13 resíduos. A segunda rede leva em conta a correlação entre os segmentos. Nessa rede, a entrada é uma janela de 17 resíduos adjacentes e possuí como entrada, a saída da primeira rede. Na terceira rede é aplicado um júri na saída de diferentes redes, computando a média aritmética para 8 redes neurais diferentes.

Diferentes estratégias têm sido implementadas para melhorar a performance das redes na predição de estruturas secundárias de proteínas. Essas estratégias podem ser agrupadas em cinco grupos principais: adicionar novos tipos de informações biológicas (como: informações evolucionárias, informações sobre a cadeia principal da molécula e dados sobre contatos), nesse caso não se utiliza apenas à homologia de seqüência primária como fonte de informação para a Rede Neural Artificial; alteração da forma de como apresentar a informação para Rede Neural. (i.e., estratégias que modificam o conjunto de treinamento); utilizar pós-processamento/filtros antes da predição; alterar a arquitetura *feed-forward* padrão utilizada; treinamento balanceado.

[6] informações de segmentos da estrutura primária que se mantêm conservada.

Todas essas estratégias alteram, significativamente, a performance das redes neurais na predição e podem ser combinadas e utilizadas juntamente. Há diferentes linhas quando se discute como melhorar a performance das redes neurais. Alguns autores exploram o problema de melhorar a base de dados, outros autores exploram também o uso de informações evolutivas obtidas pelo alinhamento de seqüências e outros autores exploram ainda que um bom projeto da rede é um fator chave para a performance da predição. O desempenho dessas três linhas é muito importante para obter melhores resultados. No capítulo seguinte descreveremos sobre a metodologia do desenvolvimento da pesquisa.

CAPÍTULO 4

Materiais e Métodos

4.1 Considerações Iniciais

Para o desenvolvimento de um modelo ou projeto baseado em redes neurais artificiais para predição da estrutura secundária de proteínas são necessárias diversas etapas:

- Coleta dos dados de treinamento, validação e teste: devem ser reunidos todos os dados pertinentes e potencialmente úteis à tarefa;
- Pré e Pós-processamento dos dados: os dados simbólicos devem ser transformados em dados puramente numéricos, os quais são mais adequados para utilização pela rede;
- Projeto da estrutura da rede: a escolha da configuração adequada da rede tem um impacto substancial no desempenho do sistema;
- Treinamento, teste e validação: o treinamento é realizado em diferentes arquiteturas de rede, e para estas arquiteturas se realiza a validação do treinamento e os testes para avaliar o melhor desempenho;
- Por fim, o Júri de Decisão: fase em que os resultados finais são gerados.

4.2 Elaboração da Base de Dados

O primeiro passo para a implementação do projeto é a coleta e seleção de três conjuntos de seqüências primárias de proteínas. O primeiro conjunto de proteínas foi selecionado através do alinhamento múltiplo (será detalhado no próximo tópico). Foram selecionadas 389 proteínas que são responsáveis pelo treinamento das diferentes arquiteturas de RNA. Estas proteínas foram divididas em 4 subconjuntos (o primeiro conjunto chamado de todas, representando as 389 proteínas, o segundo com 173 proteínas, com características de estruturas hélice-a, o terceiro com 46 proteínas com estrutura de folha-ß e o quarto formado com 115 proteínas, com estruturas de 30 a 50% hélice-a/folha-ß) para o treinamento de RNAs específicas.

O segundo conjunto de proteínas foi obtido através do EVA (Evaluation of Automatic protein structure prediction)[1], um servidor de predição de estruturas de proteínas em tempo real. Prediz as seqüências depositadas no PDB[2] a cada semana, e faz comparações com outros servidores[3]. Em particular foram utilizadas para o segundo conjunto as proteínas publicadas em 27/02/2002. Foi verificado também neste conjunto, o alinhamento múltiplo obtendo na sua totalidade 65 proteínas não homólogas, que foram responsáveis pela validação das diferentes arquiteturas de RNA. O terceiro conjunto de proteínas foi obtido através do trabalho Ning Qian e Terrence J. Sejnowski (QIAN & TERRENCE, 1988). Estas proteínas foram baseadas nas coordenadas atômicas de suas estruturas secundárias disponibilizadas pelo *Brookhaven National Laboratoty*. O conjunto tem 75 proteínas não homólogas utilizadas para testar as diferentes arquiteturas de RNA e obter o desempenho das mesmas através de coeficientes conhecidos na literatura. Para conseguir os conjuntos de treinamento é preciso utilizar a técnica de alinhamento múltiplo.

[1] Site: http://cubic.bioc.columbia.edu/eva/doc/concept.html
[2] Banco de dados de estruturas terciárias de proteínas. Este banco está disponível na Internet no seguinte *site*: www.rcsb.org/pdb
[3] Existem outros servidores de predição de estruturas como o CASP, PREDICTOR e DSPRED.

4.2.1 Alinhamento de Seqüências

O alinhamento de seqüências faz uma associação explícita entre resíduos de duas ou mais seqüências, com o propósito de inferir analogias na estrutura funcional ou evolutiva das moléculas envolvidas. Alinhar duas seqüências consiste em estabelecer uma correspondência entre os aminoácidos dessas seqüências de modo que a ordem não seja violada. Não é necessário que todos os resíduos de uma seqüência estejam associados aos resíduos da outra seqüência, entretanto é desejável que o número de associações seja o maior possível para se obter uma homologia satisfatória (VASCONCELOS, 2001). Para buscar alinhamentos ótimos (uma baixa porcentagem de homologia), é necessário definir um critério, medido na forma de um escore, pelo qual os alinhamentos são quantificados. Um exemplo de critério seria atribuir o valor +1 a um casamento de aminoácidos, -1 a um descasamento e –2 para a ocorrência de g*ap (-),* como ilustrado na figura 4.1.

```
SLNSGYHFC          S L N S G - - - Y H F C
                   |                 | | |
SFQETFLSFHFC       S F Q E T F L S F H F C
                   1-1-1-1-1-2-2-2-1 1 1 1 = -7
```

Figura 4.1 – Exemplo do cálculo de escores.

Embora a comparação de pares de seqüências seja fundamental para o estudo de homologias, a análise de grupo de seqüências que formam uma família de aminoácidos requer a habilidade e capacidade de estabelecer conexões entre mais de dois membros de um grupo com o objetivo de avaliar as características deste grupo, ou seja, o alinhamento múltiplo (VASCONCELOS, 2001). O processo de alinhamento múltiplo pode ser entendido como a melhoria da relação sinal-ruído em um conjunto de seqüências. O objetivo deste processo é transformar, por meio de *gaps*, todas as seqüências em outras do mesmo comprimento, como ilustrado na figura 4.2.

```
Y D G G A V E A L        Y D G G A V - E A L
Y D G G E A L            Y D G G - - - E A L
F E G G I L V A L        F E G G I L E V A L
F D G I L V Q A V        F D - G I L V Q A V
```

Figura 4.2– Processo de alinhamento múltiplo.

Existem várias maneiras de definir os escores, mas duas se destacam:

- Um no qual se define o escore do alinhamento múltiplo como a soma dos escores do alinhamento de todos os pares não ordenados dentre estas seqüências.
- Outro no qual se define uma seqüência denominada de “consenso, ou seja uma seqüência como modelo” e o escore do alinhamento múltiplo são definidos como a soma dos escores entre as seqüências constituintes e a seqüência consenso.

O alinhamento múltiplo e caracterizado como eficiente, quando sua porcentagem de homologia for baixa (30% de homologia caracterizada bem alinhada). Essa porcentagem é obtida pela maximização destes escores, ou seja, quanto maior o escore menor será a porcentagem de homologia. Para realizar o alinhamento dos conjuntos de treinamento e validação das proteínas foi utilizado o software ClustalX[4], que é um dos mais utilizados para alinhamento múltiplo de seqüências, e encontra-se gratuitamente disponível para as plataformas Unix e Windows. Este software baseia-se no conceito de alinhamento progressivo, o qual determina os alinhamentos para cada par de seqüências e constrói uma matriz de “distâncias” que reflete estes alinhamentos.

[4] Site: http://genome.jouy.inra.fr/doc/clustal/clustalx.html

4.3 Base de dados de estruturas secundárias

Depois de construído os três conjuntos de seqüências primárias, é preciso obter os arquivos *PDB* referentes a essas proteínas. Esses arquivos contêm informações sobre a estrutura terciária da proteína, coordenadas atômicas, estruturas secundária e primária. Para se conseguir a estrutura secundária dos três conjuntos, utilizou-se o *software DSSP*[6] que classifica as estruturas secundárias em características geométricas e a exposição dos resíduos ao solvente.

É importante ressaltar que esse *software* não prediz a estrutura secundária, pois a obtenção da mesma é feita por meio de cálculos que utilizam padrões de pontes de hidrogênio. Esse programa classifica cada resíduo em oito classes: H = hélice-a; B = folha-ß isolada; E = folha-ß; G = 3-hélice ou 3/10 hélice; I = 5 hélice *pi*; T = retorno ou volta; S = curva; e '.' = indefinido. Essas classes são, tipicamente, integradas em três classes padronizadas em hélices-α, folha-β e *Coil (fita aleatória)* (ver tabela 4.1).

8 classes	H	G	I	E	B	S	T	"."
3 classes	H	H	H	F	F	C	C	C

Tabela 4.1: Classificação das estruturas.

[6] Site: http://www.sander.ebi.ac.uk/*DSSP*/

4.4 Interface do Pré-Processamento

Com a definição do problema e a coleta dos dados concluída, segue-se à codificação dos dados de entrada das RNAs. Para isso utilizou-se *software* de interface gráfica 'conversor'[7]. Esse *software* desenvolve várias funções, tais como:

- conversão de vários arquivos *PDB* para arquivos *DSSP*, automaticamente, mas deve-se ter o programa *DSSP*cmbi.exe[8];
- obtenção das três classes (hélice-a, folha-ß e Coil (fita aleatória) que serão utilizadas;
- divisão da base de treinamento em 4 subconjuntos de acordo com a sua porcentagem de estruturas: o primeiro conjunto chamado de todas, representando as 389 proteínas; o segundo com 173 proteínas, com características de estruturas hélice-a; o terceiro com 46 proteínas com estrutura de folha-ß e o quarto formado com 115 proteínas, com estruturas de 30 a 50% hélice-a/folha-ß para o treinamento de RNAs específicas
- criação de arquivos que descrevem as matrizes de treinamento e as janelas de aminoácidos, ou seja, a codificação da base de treinamento em matrizes separadas por janelas;
- geração de arquivos estatísticos sobre as proteínas, ou seja, quantos resíduos de aminoácidos compõem a base de treinamento e qual tipo de estrutura é composta essa base;
- geração das matrizes de entrada e saída para a utilização nas RNAs. (ver Figura 4.3).

[7] O software conversor foi executado e planejado por Scott (2002) e modificado por Ferreira (2003), na criação de arquivos que descrevem entradas com janelas diferentes.
[8] O arquivo *DSSP*cmbi.exe é o software *DSSP*, executado em ambiente DOS-Windows e com um arquivo de entrada *PDB*.

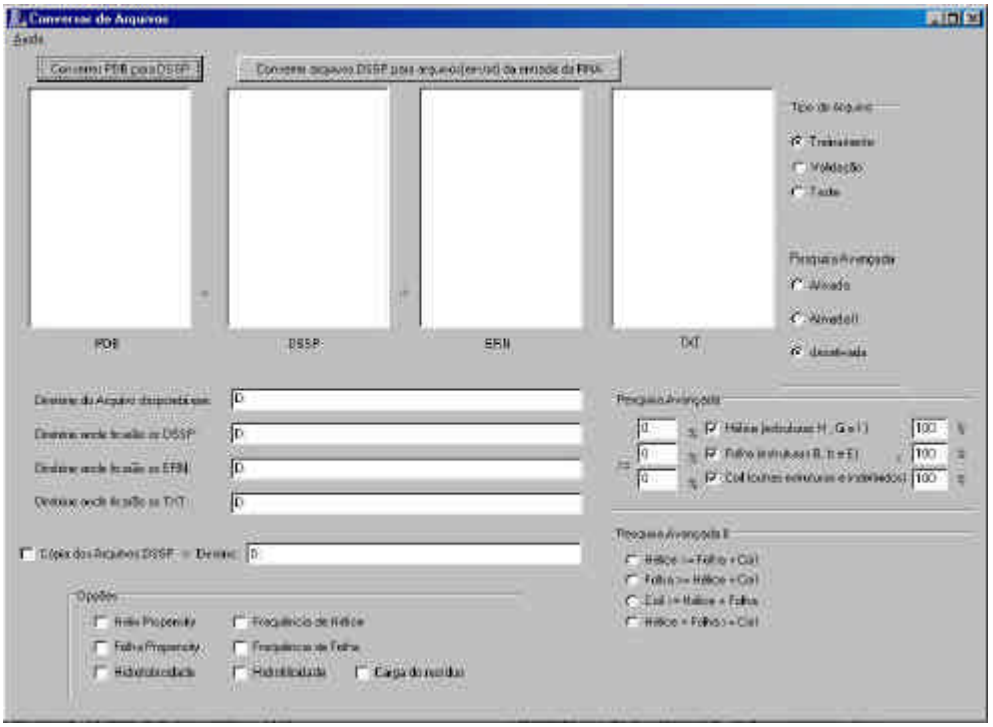

Figura 4.3: Interface do *Software* conversor (SCOTT, 2003).

Na figura 4.3 temos, a Interface do *Software* conversor com suas funções, dentre elas a redução de oito classes para três e a preparação das bases de treinamento por meio da função pesquisa ativada e pesquisa ativada II. A pesquisa ativada possui quadros de porcentagem em que se coloca o percentual da estrutura que deseja para construção da base. A pesquisa ativada II fornece a opção da escolha dessa base, sendo quatro alternativas oferecidas, em que se pode marcar:

- Hélice-a => folha-ß + *Coil (fita aleatória).* A base hélice-a será formada por proteínas que possuem a quantidade de resíduo de aminoácido em formação de hélice-a igual ou superior à soma de folha-ß mais *Coil (fita aleatória)*;
- folha-ß => Hélice-a + *Coil (fita aleatória).* A base folha-ß será constituída de proteínas com resíduos em formação de folha-ß igual ou superior à soma de hélice-a mais *Coil (fita aleatória)*;
- *Coil (fita aleatória)* => Hélice-a + folha-ß. A base *Coil (fita aleatória)* será composta por proteínas, em que o número de resíduos em formação de *Coil (fita aleatória)* é igual à soma de hélice-a e folha-ß;

– Hélice-a + folha-ß => *Coil (fita aleatória).* A base Hélice-a / folha-ß (base mista) se formará com proteínas cujos resíduos em formação de Hélice-a e folha-ß sejam iguais ou superiores aos do *Coil (fita aleatória)* Na próxima seção, abordar-se-á sobre as janelas de entradas das RNAs.

4.5 Projeção das arquiteturas das RNAs

Um dos objetivos deste trabalho constitui-se no uso de janelas de aminoácidos distintas. Foram implementados 9 RNAs, onde cada uma foi projetada com janela diferente. A escolha de tamanho diferenciado de janelas de entradas tem uma contribuição bem satisfatória no trabalho, está contribuição é atribuída pelo fato que as proteínas possui estruturas secundárias iguais, contudo de tamanho diferentes.

Assim, as 9 janelas de entradas diferentes foram desenvolvidas com o objetivo de predizer a estrutura secundária do resíduo presente na posição central das janelas. O software conversor cria arquivos de janelas distintas compreendidas entre (7, 9, 11, 13, 15, 17, 19, 21 e 23), ou seja, para cada seqüência de aminoácidos o software conversor cria 9 matrizes de entrada, sendo cada uma com formato diferente de janela. Os aminoácidos da seqüência primária de cada proteínas são codificados em dados numéricos binários, para cada resíduo de aminoácido foram atribuídas 22 unidades, sendo 20 reservadas para os aminoácidos, uma para o heteroátomo (X) e a última reservada para indicar o fim da janela ou o aminoácido não identificado pelo *DSSP*. Uma dessas 22 unidades é sinalizada com o valor 1 identificando um determinado resíduo, e as outras unidades são zeradas. (ver Figura 4.4).

1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

alanina

0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

arginina

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

Fim da seqüência

Figura 4.4: A representação de como os aminoácidos são codificados.

Na figura 4.4, tem-se um exemplo de como os aminoácidos são codificados, formando assim as matrizes de entrada compreendidas entre 0 e 1. Cada matriz de entrada possui um tamanho fundamentado na codificação dos aminoácidos em 22 unidades multiplicada pelo tamanho da janela correspondente a cada RNA. (ver tabela 4.2).

Unidades	Janelas	N°de neurônios na camada de entrada das RNAs
22	7	22 x 7 = 154
22	9	22 x 9 = 198
22	11	22 x 11 = 242
22	13	22 x 13 = 286
22	15	22 x 15 = 330
22	17	22 x 17 = 374
22	19	22 x 19 = 418
22	21	22 x 21 = 462
22	23	22 x 23 = 506

Tabela 4.2: Número de neurônios na camada de entrada das RNAs.

Para realizar os treinamentos e testes da rede é preciso fornecer a saída para cada padrão (janela). A saída da rede é a estrutura referente ao resíduo central. Para codificar os dados simbólicos (estrutura hélice-a, folha-ß e Coil (fita aleatória) em dados numéricos, foram definidos para cada saída da rede três unidades(ver Tabela 4.3).

Estrutura	Codificação
Hélice-a	1 0 0
Folha-ß	0 0 1
Coil (fita aleatória)	0 1 0

Tabela 4.3: Codificação binária das estruturas secundárias.

A partir dessa tabela 4.3, ressalta-se a codificação dos três padrões, sendo que os resíduos com formação de hélice-α recebem o valor (1 0 0), folha-β, (0 0 1) e *Coil (fita aleatória)*, (0 1 0). No tópico seguinte, discute-se para a implementação das RNAs, com suas arquiteturas e o simulador usado para execução das redes.

4.6 Implementação de uma RNA

Todas as RNAs foram projetadas e treinadas no Simulador MATLAB 5.0. Esse software é conceituado pela comunidade científica e possui várias ferramentas para implementações de redes neurais (*Neural Network Toolbox*). As RNAs implementadas foram do tipo *Multi Layer Perceptron* (MLP) com a função de treinamento *trainrp*, que utiliza o algoritmo *backpropagation* (RPROP). Na elaboração de um projeto de RNA, a tarefa consiste em determinar o número de neurônios de processamento da camada intermediária, bem como o número de camadas ocultas. Embora não existem regras para determinar o número de camadas e de neurônios. Sendo assim, as redes foram projetadas com duas camadas intermediárias e uma camada de saída de forma que a camada de entrada utiliza a função de ativação linear (*purelin*), e a demais a função de ativação *tansig* (ver Figura 4.5).

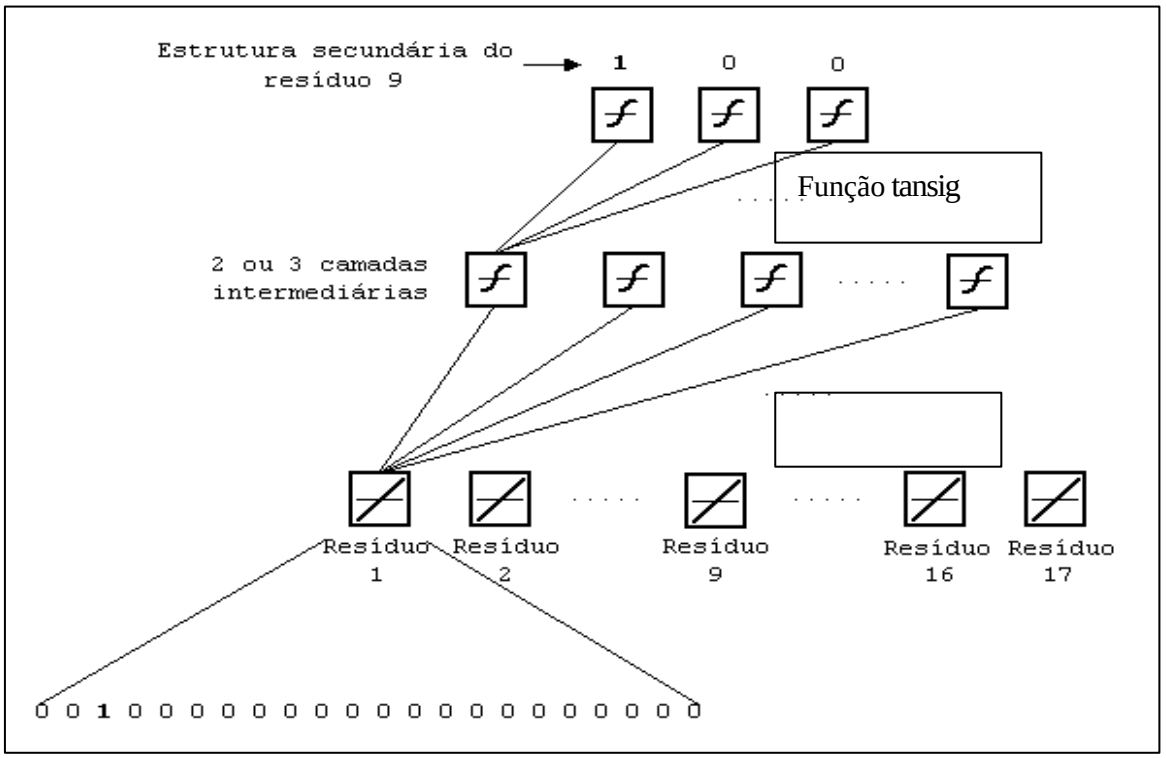

Figura 4.5: Arquitetura de uma RNA.

Na ilustração da figura 4.5 temos um aminoácido codificado sendo processado por uma rede com janela de 17 resíduos. Na primeira camada, tem 374 neurônios, ou seja, 22 unidades vezes 17, em que o processo de cálculo dos pesos nessa camada é realizado pela função linear *purelin*. Os dados, após serem calculados, são transferidos à camada seguinte. A mesma recalcula os pesos e passa para à camada de saída por meio da função *tansing*. Na seção a seguir, versa-se sobre os pesos iniciais e o critério de parada aplicado no treinamento das RNAs.

4.6.1 Treinamento e o critério de parada

Nesse processo, a princípio, os elementos de processamento das RNAs são inicializados aleatoriamente com diferentes limiares internos e com conexões de pequenos pesos, variando entre $-1.0 < w < 1.0$, em que w denota o peso de uma conexão qualquer. Em seguida, os dados existentes nas entradas e saídas do conjunto de treinamento são apresentados as RNAs, repetidamente, e, a cada ciclo de treinamento, os pesos são ajustados a fim de

checar a generalização das redes. São os dois critérios de parada: 1) parada do treinamento no momento em que ele atinja o objetivo e 2) parada do treinamento depois de determinado ciclo. No primeiro, o treinamento é interrompido quando o erro médio quadrático (Anexo1) do treinamento for menor que 0.01 (1% de erro). No segundo, o treinamento é interrompido depois de determinado número de ciclos, a fim de obter uma estimativa do erro da rede sobre o conjunto de validação. A partir do momento em que o erro médio, no conjunto de validação, apresentar crescimento, o treinamento é encerrado como se observa a figura 4.6.

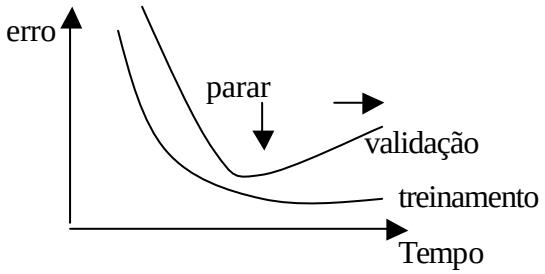

Figura 4.6: Gráfico do erro de treinamento e validação.

Existem vários métodos para a determinação do momento em que o treinamento de uma rede neural deve ser encerrado. A determinação destes critérios é fundamental para um bom treinamento e, conseqüentemente uma boa generalização. Os critérios de parada utilizada no projeto, em ordem de prioridade, são:

- treinar a rede “x” ciclos, com o conjunto de treinamento;
- simular a rede com a entrada do conjunto de validação;
- calcular o resultado contendo o módulo da diferença entre o resultado da simulação da validação e a saída real da validação;
- considerar como um erro de padrão da rede, quando cada linha da matriz de resposta tiver a soma superior a 0.5;
- somar o número de linhas do conjunto de validação que tiverem erro de padrão;
- parar o treinamento ou treinar x ciclos novamente, se no passo atual do conjunto validação contiver um número de linhas com erro ou esse erro for maior do que o número calculado.

4.7 Implementação de duas RNAS

Com o objetivo de melhorar os resultados dos testes realizados com a arquitetura composta por uma RNA, foi implementada uma nova arquitetura utilizando duas redes, sendo que a saída da primeira rede é à entrada da segunda rede. Para essa implementação foi desenvolvido um *software* chamado 'junta'[9], cuja sua função principal é juntar os resultados preditos na primeira rede com a matriz dos aminoácidos (ver figura 4.7).

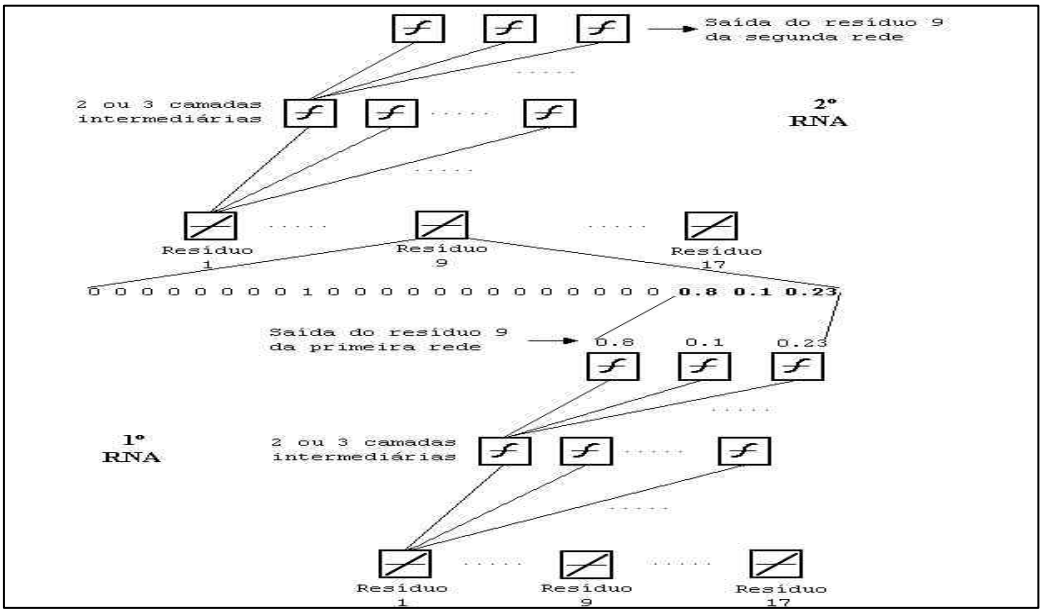

Figura 4.7: implementação de duas RNAS.

Na figura 4.7, o *software* 'junta' faz a conversão dos dados de entrada para segunda rede. Essa tarefa é desenvolvida na inclusão de três colunas, cujos valores são o resultado da predição da estrutura secundária codificado na linha da matriz. Dessa forma, a nova matriz construída possuirá tanto o resíduo como também a sua estrutura secundária identificada pela rede, como, por exemplo, 0.8 0.1 0.23. Na inclusão dessas informações, a matriz original, de 22 unidades, passa a ter 25. Com isso, finalizam-se arquiteturas das RNAs (ver Tabela 4.4).

[9] O software 'junta', planejado e executado por mim. Esse foi desenvolvido no compilador Fortran 90.
.

Janelas de resíduo de aminoácidos	Arquiteturas 1º RNAs Janela vezes 22 unidades	Arquiteturas 2° RNAs Janela vezes 25 unidades
7	154 x 24 x 12 x 3	175 x 24 x 12 x 3
9	198 x 24 x 12 x 3	225 x 24 x 12 x 3
11	242 x 24 x 12 x 3	275 x 24 x 12 x 3
13	286 x 24 x 12 x 3	325 x 24 x 12 x 3
15	330 x 24 x 12 x 3	375 x 24 x 12 x 3
17	374 x 24 x 12 x 3	425 x 24 x 12 x 3
19	418 x 24 x 12 x 3	475 x 24 x 12 x 3
21	462 x 24 x 12 x 3	525 x 24 x 12 x 3
23	506 x 24 x 12 x 3	575 x 24 x 12 x 3

Tabela 4.4: Arquiteturas de 18 RNAS.

Na tabela 4.4, em que 18 arquiteturas de RNAs são formadas por suas respectivas janelas, sua estrutura padrão é composta por 24 e 12 neurônios na camada intermediária e 3, na saída. Essas três unidades de saída classificam-se em padrões do tipo hélice-α, folha-β ou *Coil* (fita aleatória*)*. Na seção seguinte, discuti-se a implementação do júri de decisão.

4.8 Implementação do júri de decisão

A inclusão do júri de decisão destina-se a fazer uma leitura da predição final (ROST & SANDER,1993). A partir dessa motivação, desenvolveu-se um *software* chamado 'júri'[10], com a função de realizar a média aritmética sobre os resultados da predição das 18 redes. Um exemplo do resultado gerado pelo programa pode ser checado na (figura 4.8) .

[10] O software 'júri' foi planejado, executado e desenvolvido por mim (2003), no compilador Fortran 90, com base em Ros e Sander (1993).

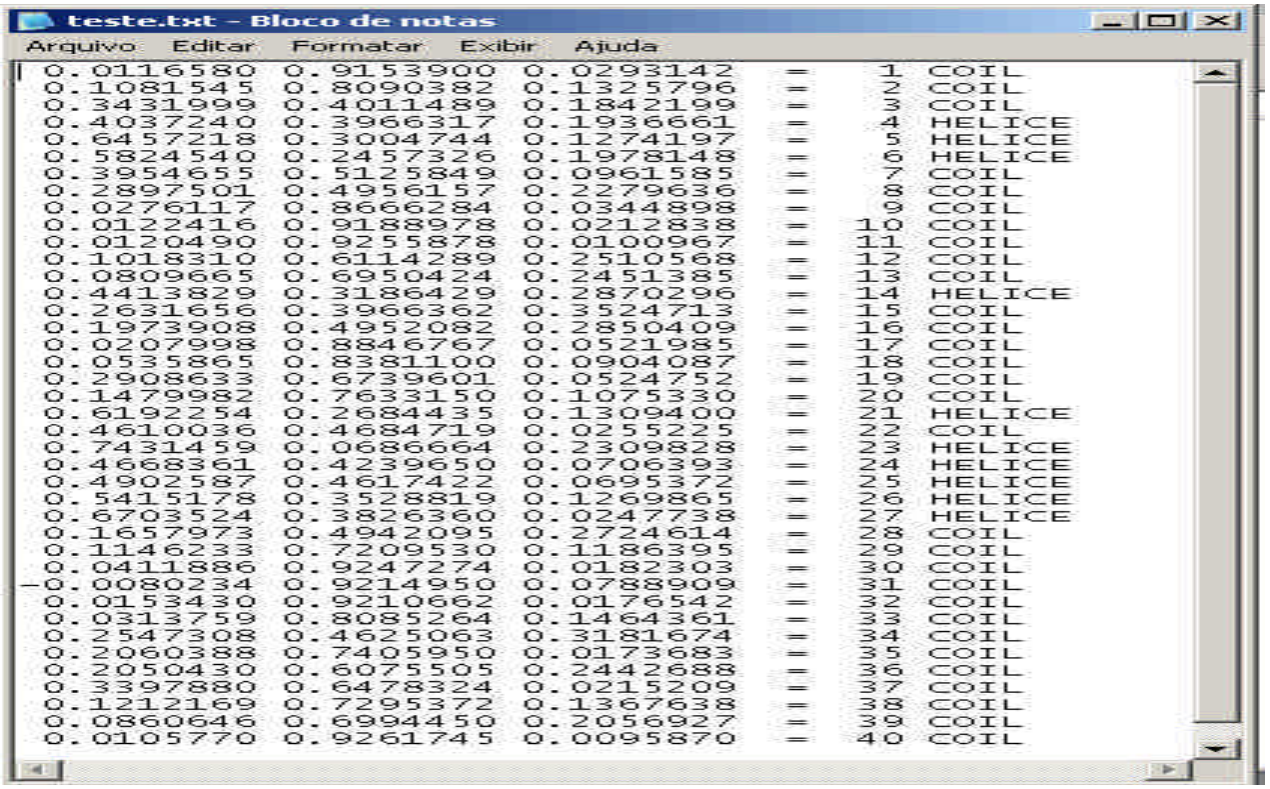

Figura 4.8: Arquivo gerado pelo *software* Júri de decisão

Na figura 4.8 temos, um arquivo gerado pelo programa júri, em que cada coluna contém o resultado da média aritmética das 18 redes. Logo após a média, executa-se o critério de classificação. Essa classificação se baseia na analise do maior valor em cada coluna. Se a primeira coluna for o maior valor, classifica-se como hélice-a, se for a segunda maior, classifica-se como *Coil (fita aleatória)* e a terceira, como folha-ß, gerando, assim, o resultado da predição da estrutura secundária do resíduo em questão. No tópico seguinte será abordado o coeficiente para analise dos resultados.

4.9 Coeficientes

Na literatura existem alguns coeficientes para verificar o desempenho de uma RNA na predição de estruturas secundárias. Dentre estes, o mais utilizado pelos pesquisadores é conhecido como (Q3), que fornece a porcentagem dos resíduos preditos corretamente considerando os três tipos de estruturas secundárias: hélice-a, folha-ß e Coil (fita aleatória).

Coeficiente Q3:

Seja i = { hélice-a, folha-ß, Coil (fita aleatória)};

TOTi = número de “i” existentes nas proteínas de teste;

PREDi = número total de “i” que a RNA identificou;

CORRi = número de “i” que a RNA identificou corretamente;

$$Q3 = \frac{\sum_{(i=H,E,C)} CORRi}{\sum_{(i=H,E,C)} TOTi} * 100 \qquad \text{Eq.(4.1)}$$

Outros coeficientes importantes e que também são utilizados no projeto para ajudar na avaliação dos testes são o Q(obs) e Q(pred). O primeiro dá a porcentagem do número de resíduos preditos corretamente em relação ao número real observado, em um estado particular. Já o Q[prd] dá a porcentagem do número de resíduos preditos corretamente em relação ao número que a RNA identificou, em um estado particular.

Seja i = {hélice, folha, Coil (fita aleatória)}

$$Qi[obs] = \frac{CORRi}{TOTi} * 100 \quad \text{Eq.(4.2)} \qquad Qi[prd] = \frac{CORRi}{PREDi} * 100 \quad \text{Eq.(4.3)}$$

Para a implementação destes coeficientes que avaliam o desempenho da rede, foi utilizado o software "Comparador"[11]. Este software fornece informações como porcentagem de acerto, linhas incompatíveis, estruturas preditas e coeficientes, como atesta a figura 4.9.

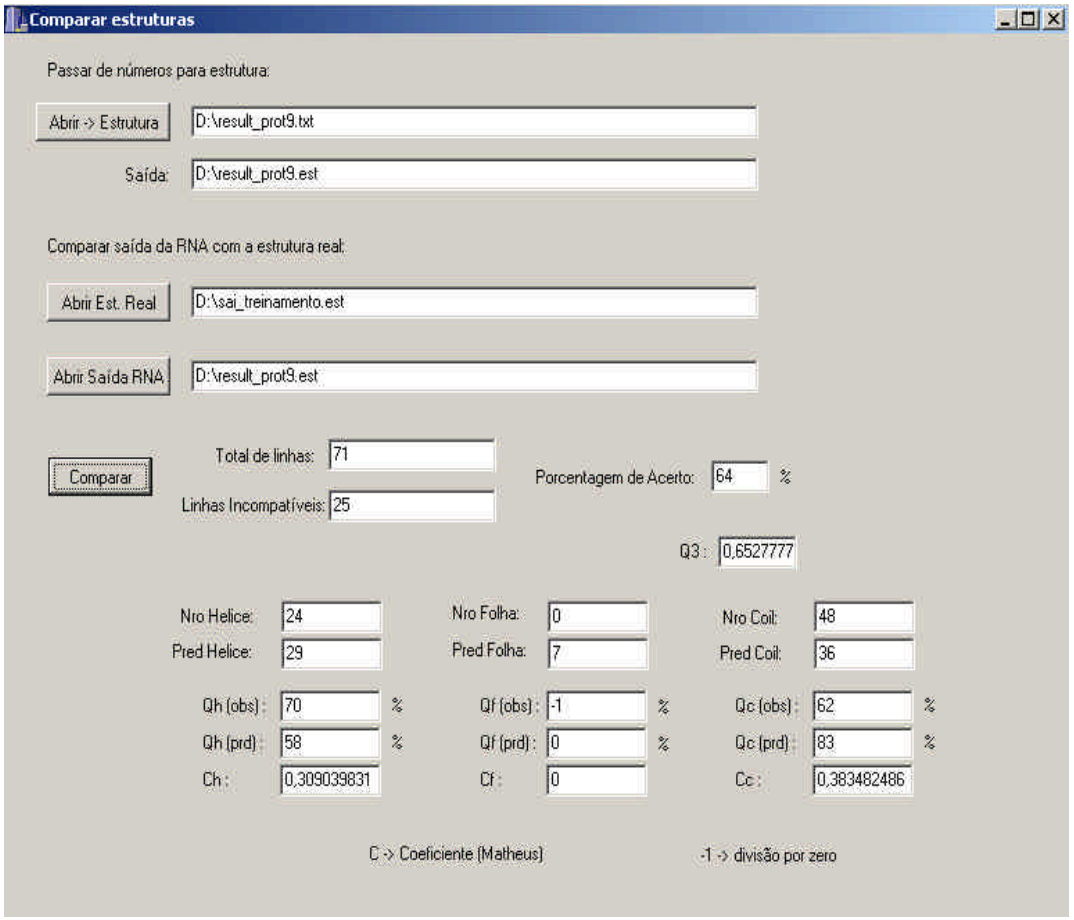

Figura 4.9: A interface do software Comparador.

Na ilustração da figura 4.9 temos, o ambiente gráfico do programa comparador. No botão "Abrir =>Estrutura", executa a leitura de arquivos codificados em estruturas secundárias. No botão "Abrir Est. Real", seleciona a estrutura real da proteínas na qual está sendo simulada e no "Abrir Saída RNA", a estrutura gerada pela rede, afim de comparar.

[11] O software foi planejado e executado pelo Scott (2003), em C++ builder 5.

Em linhas gerais, esta pesquisa processa as informações de entrada em três níveis: Nível 1, seqüência-estrutura; Nível 2, estrutura-estrutura e Nível 3, júri de decisão. No primeiro, encontram-se redes que predizem a estrutura secundária a partir da seqüência primária. Essas redes, do tipo MLP, são não-recorrente e compostas de três camadas (entrada, intermediária e saída).

No segundo, tem-se a implementação de uma segunda rede, em que os dados de saída da primeira são reaproveitados como entrada para a segunda. O último nível realiza uma média aritmética dos resultados da predição, obtidos de 18 redes distintas e treinadas independentemente. Na prática, é como se o júri recebesse a predição de diferentes redes e realizasse uma média para decidir qual a estrutura secundária está associada para o resíduo central. O próximo capítulo abordará os testes, a comparação dos resultados com outros preditores e a discussão.

Capítulo 5

Resultados

5.1 Considerações Iniciais

Neste capítulo são discutidos os testes e os resultados. As redes neurais foram treinadas com bases de treinamento diferentes: primeira base de treinamento chamada 'todas', composta por 389 proteínas, ou seja, todas as proteínas do banco de treinamento, com 137185 resíduos de aminoácidos. As outras bases de treinamentos foram construídas com o auxilio do software conversor, realizando uma filtragem na base treinamento 'todas'.

Essa filtragem permitiu a escolha de três bases de treinamento: base de treinamento folha-ß, formado com 56 proteínas, cuja a soma de resíduos em formação de folha-ß é maior que hélice-a e coil, com 14772 resíduos de aminoácidos; base de treinamento hélice-a, constituído com 173 proteínas do tipo hélice-a, com 70160 resíduos de aminoácidos e a base de treinamento hélice-a / folha-ß, formado com 115 proteínas com 30 a 50% de estruturas do tipo hélice-a e folha-ß (conhecido como base de treinamento mista), com 39707 resíduos de aminoácidos. Todas as simulações com as bases de treinamento utilizaram para validação as proteínas disponíveis no conjunto de validação, 78 proteínas com 41466 resíduos de aminoácidos e para o teste foram utilizadas as proteínas coletadas a partir do artigo publicado por Qian (QIAN, 1988), 65 proteínas com 12419 resíduos de aminoácidos.

5.2 Simulação 1 – Arquitetura com uma RNA

Nestas simulações foram usadas redes treinadas com as bases de treinamento descritas anteriormente e projetadas com 4 camadas (entrada, duas intermediárias e uma saída). A camada de entrada possui 374[1] neurônios, as intermediárias com 24 e 12 neurônios e a saída 3 neurônios. A função de ativação da camada de entrada é a *purelim* e as demais camadas *tansig*. Nesta fase comparamos o erro de treinamento após a fase teste, ou seja, as redes depois de treinadas com suas respectivas base de treinamento elas são testadas com o conjunto de 65 proteínas chamado 'conjunto teste'.

A porcentagem de acerto (Q3), foi obtido com a comparação dos resíduos em formação de hélice-a, folha-ß e coil (fita aleatória) observados no conjunto das proteínas teste, identificado como (Q[obs])[2] e os preditos pela rede (Q[prd])[3]. Todas as simulações mantiveram o mesmo ciclo de treinamento igual a 800. O objetivo dessas simulações é verificar qual base de treinamento possui o melhor desempenho e qual das arquiteturas possui o melhor rendimento. Obs: O calculo de erro de treinamento segue anexo1.

Testes:

Base de Treinamento: Todas – refere-se a todas as proteínas (389 proteínas).
Arquitetura (camadas): 4 camadas (374 – purelim; 24 - tansig; 12 - tansig; 3 –)
Ciclos: 800 **Erro de Treinamento:** 9,8%
Porcentagem de Acerto (Q3): 50,62%
Hélice: Q[obs]: 37% Q[prd]: 51% ; **Folha:** Q[obs]: 36% Q[prd]: 45% ; **Coil:** Q[obs]: 64% Q[prd]: 63%

Base de Treinamento: Hélice (173 proteínas). 2 matrizes de 54 proteínas e uma de 65.
Arquitetura (camadas): 4 camadas (374 – purelim; 24 - tansig; 12 - tansig; 3 –)
Ciclos: 800 **Erro de Treinamento:** 7,5%
Porcentagem de Acerto (Q3): 50,93%
Hélice: Q[obs]: 62% Q[prd]: 42% ; **Folha:** Q[obs]: 19% Q[prd]: 50% ; **Coil:** Q[obs]: 58% Q[prd]: 64%

Base de Treinamento: Folha (56 proteínas)
Arquitetura (camadas): 4 camadas (374 – purelim; 24 - tansig; 12 - tansig; 3 –)
Ciclos: 800 **Erro de Treinamento:** 1,6%
Porcentagem de Acerto (Q3): 47,41%
Hélice: Q[obs]: 8% Q[prd]: 58% ; **Folha:** Q[obs]: 70% Q[prd]: 35% ; **Coil:** Q[obs]: 58% Q[prd]: 65%

[1] 374 neurônios são equivalentes a uma janela de 17 aminoácidos, ou seja 17 vezes 22unidades.
[2] Q[obs]: quantidade em percentual observada na proteína.
[3] Q[prd]: quantidade em percentual predita pela rede.

Base de Treinamento: Hélice-Folha (173 proteínas)
Arquitetura (camadas): 4 camadas (374 – purelim; 24 - tansig; 12 - tansig; 3 –)
Ciclos: 800 **Erro de Treinamento:** 6,25%
Porcentagem de Acerto (Q3): 52,84%
Hélice: Q[obs]: 44% Q[prd]: 50% ; **Folha:** Q[obs]: 39% Q[prd]: 45% ; **Coil:** Q[obs]: 63% Q[prd]: 64%

5.3 Simulação 2 – Arquitetura duas RNAs

Nesta fase de testes, o principal objetivo foi verificar o desempenho entre os resultados obtidos com uma arquitetura composta com duas redes e aqueles obtidos anteriormente utilizando apenas uma rede. Nos quadros abaixo temos os erros de treinamento de uma única rede, representada pela arquitetura (374-purelim; 24-tansig; 12-tansig; 3) e os erros de treinamento de duas redes, representados por (425 - purelim; 24 - tansig; 12 - tansig; 3).

Base de Treinamento: Todas (216 proteínas).
1ª RNA - Arquitetura (camadas): 4 camadas (374 - purelim; 24 - tansig; 12 - tansig; 3 –)
1ª RNA – Ciclos: 800 **Erro de Treinamento:** 9,8%
2ª RNA - Arquitetura (camadas): 4 camadas (425 - purelim; 24 - tansig; 12 - tansig; 3 –)
2ª RNA - Ciclos: 800 **Erro de Treinamento:** 3,47%
Porcentagem de Acerto (Q3): 51%
Hélice: Q[obs]: 38% Q[prd]: 55% ; **Folha:** Q[obs]: 48% Q[prd]: 42% ; **Coil:** Q[obs]: 59% Q[prd]: 65%

Base de Treinamento: Hélice (173 proteínas).
1ª RNA - Arquitetura (camadas): 4 camadas (374 - purelim; 24 - tansig; 12 - tansig; 3 –)
1ª RNA - Ciclos: 800 **Erro de Treinamento:** 7,5%
2ª RNA - Arquitetura (camadas): 4 camadas (425 - purelim; 24 - tansig; 12 - tansig; 3 –)
2ª RNA - Ciclos: 800 **Erro de Treinamento:** 2,68%
Porcentagem de Acerto (Q3): 52,11%
Hélice: Q[obs]: 58% Q[prd]: 45% ; **Folha:** Q[obs]: 25% Q[prd]: 51% ; **Coil:** Q[obs]: 60% Q[prd]: 64%

Base de Treinamento: Folha (56 proteínas)
1ª RNA - Arquitetura (camadas): 4 camadas (374 - purelim; 24 - tansig; 12 - tansig; 3 –)
1ª RNA - Ciclos: 800 **Erro de Treinamento:** 1,6%
2ª RNA - Arquitetura (camadas): 4 camadas (425 - purelim; 24 - tansig; 12 - tansig; 3 –)
2ª RNA - Ciclos: 800 **Erro de Treinamento:** 0,9%
Porcentagem de Acerto (Q3): 49,56%
Hélice: Q[obs]: 16% Q[prd]: 53% ; **Folha:** Q[obs]: 67% Q[prd]: 37% ; **Coil:** Q[obs]: 60% Q[prd]: 65%

Base de Treinamento: Hélice-Folha (173 proteínas)
1ª RNA - Arquitetura (camadas): 4 camadas (374 - purelim; 24 – tansig; 12 - tansig; 3 –) **1ª RNA - Ciclos:** 800 **Erro de Treinamento:** 6,25%
2ª RNA - Arquitetura (camadas): 4 camadas (425 - purelim; 24 – tansig; 12 - tansig; 3 –) **2ª RNA - Ciclos:** 800 **Erro de Treinamento:** 1,5%
Porcentagem de Acerto (Q3): 57% **Hélice:** Q[obs]: 45% Q[prd]: 55% ; **Folha:** Q[obs]: 45% Q[prd]: 47% ; **Coil:** Q[obs]: 64% Q[prd]: 64%

Percebe-se com estes resultados que o erro de treinamento de redes projetadas com duas redes diminui em até 5% em relação ao erro de treinamento de uma única rede. Observamos que a projeção de arquiteturas composta por duas redes, melhora o desempenho da predição. A melhor porcentagem de acerto verificada com arquitetura de uma rede foi com a base de treinamento hélice-a/folha-ß alcançando 52,84%. Utilizando a arquitetura formada por duas redes com a mesma base, chegou-se a um acerto de 57%.

Pode-se observar que redes treinadas com diferentes bases de dados (todas, hélice-a, folha-ß e hélice-a/ folha- ß) possuem performances diferentes para o mesmo conjunto de teste. Portanto, no interesse de escolher a melhor base de treinamento, foram realizadas testes com todas as bases, os resultados obtidos apontaram a base hélice-a/folha-ß como a melhor. Para efeito de comparação os testes posteriores foram executados com redes treinadas com a base hélice-a/folha-ß. A seguir são discutidos os resultados realizados com proteínas do CASP[2] .

5.4 Resultados com proteínas do CASP

Para a avaliação do preditor foram selecionadas 15 proteínas. Essas proteínas são usadas pelo *Critical Assessment of Structure Prediction* (CASP)(Zemla *et al*., 2001) na avalição dos métodos de predição de estrutura secundária (ver tabela 5.3).

Proteínas	nome	Nº=a	Proteínas	nome	Nº=a
1QLQ	Pancreatic Trypsin inhibitor	58	1DT4	Neuro o.ventral antigen 1	73
1EIG	Eotaxin - 2	73	1EDS	Rhodopsin	31
1C56	Butantoxin	40	1G6X	Trypsin inhibitor	58
1DAQ	Endoglucanase SS	71	1DOI	Ferredoxin	128
1EHD	Aggutinim isolectin VI	89	1FD8	Atxi copper chaperone	73
1E5B	Xylanase D	87	1FE5	Phospholipase A2	118
1EJG	Crambin	46	1EHJ	Crambin	46
1ES1	Cytochrome B5	82			

Tabela 5.1 Proteínas do CASP.

[2] CASP: Critical Assessment of Structure Prediction

A fim de verificar a porcentagem de acerto de cada janela foi realizados testes com nove redes com janelas entre 7 a 23. Primeiramente as projetadas com apenas uma rede e depois as projetadas com duas redes. No final de cada teste aplicou-se o júri de desição (júriR1_% -aplicado em resultados de apenas uma rede e júriR2_% aplicado nos resultados de duas redes) (ver tabela 5.2 e 53).

Proteínas	Janela 7_%	Janela 9_%	Janela 11_%	Janela 13_%	Janela 15_%	Janela 17_%	Janela 19_%	Janela 21_%	Janela 23_%	Júri R1_%
1QLQ	67	68	60	58	50	67	60	60	53	75
1EIG	54	63	60	50	58	60	57	50	54	67
1C56	45	37	35	42	37	40	37	35	35	45
1DAQ	61	66	57	69	50	63	58	46	60	74
1EHD	48	53	47	51	48	42	44	46	49	52
1E5B	57	54	52	49	43	45	56	49	47	59
1EJG	60	69	76	73	65	69	51	50	47	71
1ES1	58	63	57	57	58	63	51	50	63	74
1DT4	64	52	54	49	58	49	53	56	54	63
1EDS	51	54	58	51	41	48	35	38	41	58
1G6X	67	68	60	58	50	67	60	60	53	75
1DOI	53	44	42	42	50	47	45	39	44	55
1FD8	52	49	53	54	54	58	67	57	59	60
1FE5	59	64	56	54	56	62	44	46	48	64
1EHJ	64	60	52	67	54	54	54	57	60	70

Tabela 5.2: porcentagem de acerto R1(uma rede).

Proteínas	Janela 7_%	Janela 9_%	Janela 11_%	Janela 13_%	Janela 15_%	Janela 17_%	Janela 19_%	Janela 21_%	Janela 23_%	Júri_ R2_%
1QLQ	70	58	63	63	50	72	58	74	53	78
1EIG	67	57	57	60	60	69	60	58	69	71
1C56	42	40	32	47	45	50	35	42	45	47
1DAQ	66	69	60	60	54	64	61	57	64	77
1EHD	52	56	43	49	47	38	55	46	47	56
1E5B	60	47	59	58	45	56	55	56	60	63
1EJG	54	67	65	76	60	62	56	40	50	76
1ES1	67	63	67	67	60	69	60	50	74	73
1DT4	64	65	58	58	58	53	54	60	58	64
1EDS	54	67	54	38	32	41	19	45	45	67
1G6X	70	58	63	63	50	72	58	74	53	77
1DOI	58	53	44	51	46	52	42	41	48	57
1FD8	68	56	61	60	68	63	73	68	79	83
1FE5	55	61	52	50	55	62	45	44	51	61
1EHJ	61	63	63	57	50	60	47	60	61	73

Tabela 5.3: porcentagem de acerto R2(duas redes).

Os resultados apresentados nas tabelas mostraram que aplicação do júri de decisão nas nove redes projetadas de uma única rede (tabela 5.2) comprovaram um acerto significativo entre 45 a 75%. Na implementação da segunda rede (tabela 5.3) a eficiência do júri de decisão aumenta para 47 a 83% de acerto. Isso mostrou que o júri de decisão possui um aumento considerável para o problema de predição.

5.5 Resultado com o Júri de Decisão com 18 redes

Este resultado tem a finalidade de avaliar a predição das proteínas do CASP, usando o júri de decisão com 18 redes. Dessas redes 9 são projetadas de uma única rede, representada pelo percentual final (Júri R1_%) e 9 são de duas redes (Júri R2_%). No final foi aplicado o júri de decisão na soma dos resultados gerados por R1(uma rede) e R2(duas redes), resultando a porcentagem final de acerto representada por (Júri_18_%) (ver tabela 5.4).

Proteínas	Júri R1_%	Júri R2_%	Júri_ 18_%	Proteínas	Júri R1_%	Júri R2_%	Júri_ 18_%
1QLQ	75	78	84	1ES1	74	73	74
1EIG	67	71	73	1DT4	63	64	64
1C56	45	47	57	1EDS	58	67	61
1DAQ	74	77	85	1G6X	75	77	84
1EHD	52	56	56	1DOI	55	57	61
1E5B	59	63	63	1FD8	60	83	83
1EJG	71	76	80	1FE5	64	61	67
				1EHJ	70	73	76

Tabela 5.4: resultado do júri de decisão.

Os resultados obtidos com redes treinadas com 115 proteínas com características hélice-a/folha-ß e projetadas com janelas de 7 a 23. Mostram que o júri aplicado nas predições das proteínas do CASP, usando 18 redes aumenta o acerto chegando atingir um percentual de até 85% de acerto. A aplicação do júri de decisão tem sido satisfatório por vários métodos de predição de estrutura secundária de proteínas, os mais citados são o de (Rost & Sander, 1994) e (John & Martin, 1999). O primeiro, publicado em 1994 é uma extensão do trabalho anterior de Rost e Sander, adicionando informações derivadas do alinhamento múltiplo como dados de entrada para a rede neural. Eles utilizaram o peso de

conservação relativo a posição específica da seqüência para aumentar a performance e o número de inserções e remoções para evitar sobre-predição e melhorar a exatidão. A rede final obteve uma exatidão de 71,6% em um teste *cross-validation* sobre um conjunto de 126 proteínas. Com a implementação do júri de decisão a predição obteve um rendimento de 5% a mais, chegando atingir um acerto de 76%. O júri realiza uma média aritmética sobre os resultados (a predição) obtidos por 12 redes distintas, essas redes distintas possuem a mesma arquitetura porém são treinadas com conjunto de dados diferentes.

No trabalho de John e Martin, os autores usam um júri de decisão em 15 RNAs diferentes projetadas com janelas de 15 a 27, dessas quinze, cinco possuem 5 camadas internas , cinco com 7 camadas internas e cinco com 9. Os resultados de cada rede são reunidos em júri de decisão, onde o mesmo realiza a predição do resíduo central em questão, atingindo um acerto de até 86%. Na prática, é como se o júri recebesse a predição de diferentes RNAs e realizasse uma média para decidir qual a estrutura secundária é associada para o resíduo central na janela da primeira rede.

5.6 Comparação dos resultados com outros preditores

Para efeito de avaliar a qualidade do preditor desenvolvido, batizado com PREDCASA. As 15 proteínas citadas na tabela 5.1, foram submetidas em 3 preditores disponíveis no mercado: o primeiro é o *Predict Secondary Structure* (PSIPRED) (JONES, 1999), o segundo *Predict Protein HeiDelberg* (PHD)(ROST 1993) e o terceiro *Protein Sequence Analysis* (PSA) (STULTZ, 1993). Esses possuem um preditor de estrutura secundária de proteínas on-line, disponível na internet. Os resultados de cada proteína seguem em anexo2

5.6.1 Resultados

Os resultados obtidos através das predições foram reunidos em gráfico. Na figura (comparação dos resultados 5.1) temos, as proteínas no eixo da abscissa e a porcentagem de acerto no eixo das ordenadas.

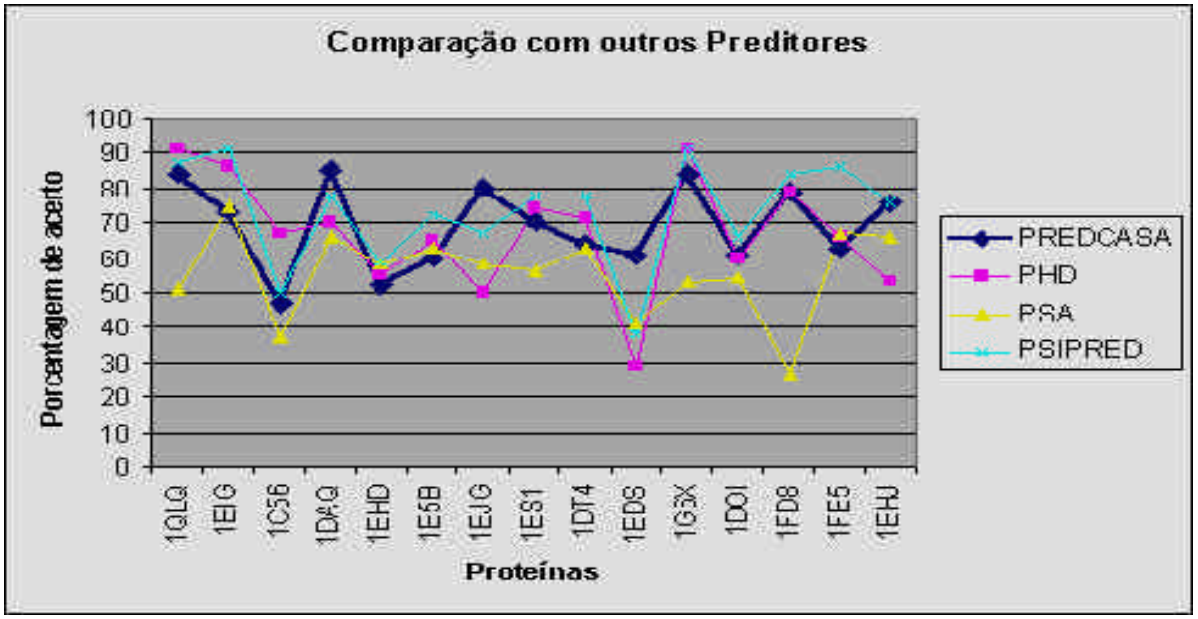

Figura 5.1 Comparação dos resultados

Na figura 5.1 é apresentada a comparação do preditor PREDCASA com o PSIPRED, o PHD e o PSA. Percebe-se uma certa regularidade de percentual de acerto de 47% até 84% para o PREDCASA em relação aos outros . Nesta fase foi analisado o acerto das estruturas individuais. Foi somado o total de resíduos em formação hélice-a, folha-ß e coil (fita aleatória) das proteínas do CASP e o total previstos pelos preditores (ver tabela 5.5).

Estruturas	CASP	PREDCASA	PHD	PSA	PSIPRED
Hélice-a	318	284	204	302	303
Folha-ß	211	181	275	124	208
Coil	566	627	612	671	584

Tabela 5.5. acerto geral dos preditores.

Fazendo a subtração dos resultados dos preditores em relação ao número de aminoácidos formando as estruturas hélice-a, folha-ß e *coil* das proteínas do CASP tem-se o total de erro obtido com cada preditor (ver tabela 5.6):

Estruturas	PREDCASA	PHD	PSA	PSIPRED
Hélice-a	34	114	16	15
Folha-ß	30	64	87	3
Coil	61	46	105	18

Tabela 5.6. erro total dos preditores.

Os resultados na tabela 5.6 temos que a base de treinamento hélice-a/ folha-ß possui um influência positiva nas predições. O erro do preditor PREDCASA em relações as estruturas hélice-a e folha-ß são praticamente iguais, o PHD possui um erro menor com estruturas do tipo folha-ß e *coil* (fita aleatória) . O método usado pelo preditor PSA, possui um erro menor com as estruturas do tipo hélice-a. Por fim temos o preditor PSIPRED, o seu erro é praticamente iguais em todas estruturas submetidas, mostrando então a eficiência de sua base de treinamentoAo fim de todas predições foi feita uma média aritmética para analisar o percentual de acerto total dos preditores (ver tabela 5.7).

PROTEÌNAS	PREDCASA %	PHD %	PSA %	PSIPRED %
1OLO	84	91	51	87
1EIG	73	86	75	91
1C56	47	67	37	50
1DAO	85	70	66	78
1EHD	56	55	58	59
1E5B	63	65	63	72
1EJG	80	50	58	67
1ES1	74	74	56	78
1DT4	64	71	63	78
1EDS	61	29	41	38
1G6X	84	91	53	91
1DOI	61	60	54	66
1FD8	83	79	27	84
1FE5	67	66	67	86
1EHJ	76	53	66	76
MÈDIA	**69.13**	**67.13**	**55.66**	**73.4**

Tabela 5.7: análise das médias de acerto

Os resultados mostraram que o júri de decisão aplicado nos 18 resultados gerado por 9 redes R1(uma rede) e 9 redes R2(duas redes) mostraram que a média de acerto do PREDCASA é de 69,13%, esse valor é superior ao do PHD e perdendo apenas para o PSIPRED com 73,4%. Foram analisadas as proteínas (1C56, 1EDS e 1EJG), com o PREDCASA o acerto é de 57, 67 e 80%, PHD o acerto foi de 67, 29 e 50%, PSA obteve um acerto de 37, 41 e 58% e o PSIPRED obteve um acerto de 50, 38 e 67%. Isso comprova que o preditor PREDCASA também tem uma média de acerto melhor com proteínas, cujo número de aminoácidos é menor que 50 resíduos. Portanto, a média geral atingida pelo preditor PREDCASA comprova a importância de redes treinadas com janelas diferentes, e o júri de decisão aplicado na soma dos resultados de R1(uma rede) e R2 (duas redes).

Capitulo 6

Conclusões

6.1. Considerações Finais

Nesse trabalho, foram explorados os efeitos do projeto da arquitetura, a base de treinamento, o uso de varias janelas de aminoácidos e um júri de decisão. O projeto de arquitetura teve como objetivo a implementação de camadas intermediárias, visando ao menor tempo de treinamento e ao maior aprendizado das redes. Os resultados obtidos demonstraram que redes projetadas com duas camadas intermediárias possuem uma boa performance. Com o desenvolvimento do software 'conversor', foi possível criar as bases de treinamentos hélice-a, folha-ß e hélice-a/folha-ß. Essas bases de treinamentos contribuíram favoravelmente na predição de estrutura secundária da proteína e na compreensão melhor das informações obtidas através dos bancos de dados.

O trabalho teve como ponto principal a implementação de várias janelas de entradas diferentes, chamadas de janelas de aminoácidos e o júri de decisão. O uso de janelas diferenciadas, revelaram através de muitos testes que proteínas com características de resíduos em formação de hélice-a são melhores preditos por redes com janelas compreendidas entre 7 a 11. As proteínas que possuem grande parte de sua estrutura com resíduos em formação de hélice-a e folha-ß, são melhores preditos por redes com janelas entre 13 a 17 e as proteínas com grande parte dos resíduos em formação de folha-ß as redes com janelas de 19 a 23. Podemos também verificar que o uso de janelas diferentes e a aplicação do júri de decisão são extremamente importantes para a performance da predição 1D usando RNAs.

Portanto, os resultados apresentados e discutidos no capítulo 5 reforçam a idéia defendida por (HEAD & GORDON, 1993) de que o projeto da arquitetura de uma rede neural é tão importante quanto o da base de dados para a performance dela na predição 1D. Uma rede projetada com muito cuidado e com entradas adequadamente codificadas pode suprir as dificuldades de uma base de dados pobre ou pequena e melhorar de forma significativa. Essa melhora pode ser verificada, com a implementação de uma arquitetura com duas camadas intermediária, o projeto de entradas diferentes (janelas de aminoácidos) e o júri de decisão, mostraram que os resultados obtidos com a base de treinamento hélice-a/folha-ß (base mista) com 115 proteínas foram melhores do que os resultados obtidos com a base de treinamento todas com 389 proteínas.

Por fim, para esses resultados serem satisfatório é preciso também tomar muito cuidado com o tipo de dados fornecido como entrada para a rede, para que não haja padrões sobrepondo, o que dificulta o problema de classificações de padrões. O treinamento de algumas redes neurais chegou a levar aproximadamente 24 horas em um Pentium IV. Já os testes e operacionalização das redes, após serem treinadas, são de 10 minutos para cada proteínas, este é o tempo gasto pela predição em todas as janelas e depois passarem pelo júri de decisão.

6.2 Contribuições do Trabalho

Sem dúvida, a principal contribuição do trabalho está na obtenção de conhecimento da técnica Predição 1D e o desenvolvimento de um preditor com redes projetadas com janelas diferentes e o uso do Júri de Decisão. Existem muitos trabalhos na literatura correlacionados com este projeto, e muitos são os pesquisadores que utilizam a técnica de redes neurais artificiais na predição de estruturas protéicas. Entretanto, tais desenvolvimentos são realizados, em sua grande maioria, fora do país. Não existe até o momento no Brasil, algum grupo, órgão ou centro de pesquisa voltado para o estudo desta ferramenta na predição de estruturas secundária de proteínas.

6.3 Propostas para Trabalhos Futuros

Tendo em vista todo o conteúdo deste projeto, muito ainda pode ser feito para melhorá-lo, como:

- Estudo mais aprofundado no uso de janelas diferentes de aminoácidos;
- Projetar novas bases de treinamento;
- Colocar informação evolutiva na entrada;
- Utilização de um simulador de redes neurais artificiais para plataforma linux.
- A disponibilidade de um Web para predição on-line;
- E o desenvolvimento de uma rede que possa identificar sítios de interação proteína-proteína.

Referências Bibliográficas

BATTITI, R. First and Second-Order Methods for Learning: Between Steepest Descent and Newton's Method. *Neural Comput,* 4:141, 1992.

BAUCOM, A., CLINE, M., HAUSSLER, D., GREGORET, L. M. Prediction of Beta-Sheet Structure Using Neural Networks. Poster presented at the *10th Annual Protein Society Meeting,* San Jose, California, August 3-7, 1996.

BRAGA, A. P., CARVALHO, A. C. P. & LUDERMIR, L. F. E T. B. *Fundamentos de redes neurais artificiais*. XI Escola Brasileira de Computação. Rio de Janeiro: Limited Edition, 1998.

BURKE, H., ROSEN, D., & GOODMAN, P. Comparing the Prediction Accuracy of Artificial Neural Networks and Other Statistical Models for Breast Cancer Survival. In: TESAURO, G., TOURETZKY, D., & LEEN, T. (org.), *Advances in Neural Information Processing Systems.* The MIT Press, 7:1063-67, 1995.

CAMPBELL, M. K. *Bioquímica.* 3ª ed. Porto Alegre: Artes Médicas Sul, 2000.

COSTA, T. B. S., & MELO, E. V. *aplicação de redes neurais mult layer perceptron na predição de estruturas secundárias de proteínas.* 2002, Monografia (Projeto Final de Graduação), Departamento de Física, UNESP, São José do Rio Preto.

CYBENKO, G. Approximation by Superpositions of a Sigmoidal Function. Mathematics of Control. *Signals and Systems*, 2:303-314, 1989.

CYBENKO, G. *Continous Valued Neural Networks with Two Hidden Layers Are Sufficient. Technical Report*. Departament of Computer Science. Madford: Tufts University, 1988.

CHANDONIA, J. M. & KARPLUS, M. The Importance of Larger Data Sets for Protein Secondary Structure Prediction with Neural Networks. *Protein Science*, 5:768-74, 1996.

____. New Methods for Accurate Prediction of Protein Secondary Structure. *PROTEINS: Structure, Function, & Genetics*, 35:293-306, 1999.

CHOU, K.C. Prediction of Tight Turns and Their Types in Protein. *Aal. Biochem,* 298: 1-16, 2000.

DENKER, J. *at all*. Neural Network Recognizer for Hand Written Zip Code Digits. *Neural Networks World,* 6(3):241-249, 1996.

DILL, K. A. *et al*. Principles of Protein Folding – A Perspective Form Simple Exact Models. *Protein Science*, 4:561-602, 1995.

____. *Dominate Forces in Protein Folding.* Biochemistry, v. 29(31):7133-55, August, 1990.

HAGAN, M. & MENHAJ, M. Training Feed Forward Networks with the Marquard Algorithm. *IEEE Transactions on Neural Networks*, 5(6): 989-93, 1994.

HARPREET, K., GAJENDRA, P. S. R. Prediction of ?-Turns in Proteins From Multiple Alignment Using Neural Network. Institute of Microbial Technology, Setor 39A, chandigarh. Protein Science, 12(3): 627-34, 2003.

HEAD–GORDON, T. & STILINGER H. F. Optimal Neural Networks for Protein Structure Prediction. *Physical Review E*, v.48, 2:1502-15, August, 1993.

HOLLEY, L. H. & KARPLUS M. *Neural networks for protein Structure Prediction*. Methods in Enzymology, v. 2002, 204-24, 1991.

JONES, D. T. Protein Secondary Structure Prediction Based on Position-Specific Scoring Matrices. *J. Mol. Biol.,* 292:195-202, 1999.

MATHEWS, C. K., Van HOLDE, K. E. *Biochemistry.* Redwood City: Benjamin/Cummings Publishing Company Inc, 1990

MIGHELL, D. A., WIKINSON, T. S., & GOODMAN J. W. Backpropagations and Its Application To Handwritten Signature Verification. In: LIPPMANN, R. P., MODDY, J. E., TOURETZKY, D. S. (org.) Advances In Neural Information Processing Systems 2. Morgan Kaufmann, 1988.

PAULING, L.; COREY, R.B.; BRANSON, H.R. The Structure of Proteins: Two Hydrogen-Bonded Helical Configurations of the Polypeptide Chain. Proc. *Natl. Acab. Sci. USA,* 37:205-211, 1951.

PITTS, W. & MCCULLOCH, W.S. A Logical Calculus of the Ideas Immanent in Nervous Activity*. Bulletin of Matematical Biophysics*, 5:115-33, 1943.

QIAN, N. & SEJNOWSKI, T. J. Predicting the Secundary Structure of Globular Proteins Using Neural Network Models*. J. Mol. Biol.*, 202:865-884, 1988.

REATEGUI, E., CAMPBELL, J. A. A Classification System for Credit Card Transactions. In: Keane, M., Haton, J. & Manago, M. (org.) *Preceedings of the Second European Workshop on Case-Based Reasoning*. 167-74, November 1994.

ROSENBLATT, F. *Principles of Neurodynamics: Perceptrons and the Teory of Brain Mechanisms.* New York: Spartan Books, 1962.

ROST B. Better 1D Predictions by Experts with Machines. *Proteins,* 1:192-197, 1997.

____. Evolution Teaches Neural Networks. In: CLARK, J. W., LINDENAU, T. & RISTIG, M. L (org.) *Scientific Applications of Neural Nets.* 207-223, 1999.

____. PHD: Predicting One-Dimensional Protein Structure by Profile Based Neural Network. *Methods in Enzymology*, 266:525-539, 1996.

____. Protein Structure Prediction in 1D, 2D e 3D 2000. *The Encyclopedia of Computacional Chemistry*, 3:2242-2255, 1998.

____. & SANDER, C. Improved Prediction of Secondary Structure by Use Sequence Profiles & Neural Networks. *Biophysics,* 90:7558-62, 1993.

RUMELHART, D. E., HINTON, G. E. & WILLIAMS, R.J. Learning Representations by Back-Propagation Errors. *Nature*, 323:533-6, 1986.

SCOTT, L. B. P. Investigação da utilização de algoritmos genéticos e redes neurais artificiais na predição de estruturas protéicas. 2003, Departamento de Física, UNESP, São José do Rio Preto.

SNOW M. E. Powerful Simulated-Annealing Algorithm Locates Global Minimum of Protein-Folding Potentials from Multiple Starting Conformation. *J. Comp. Chem.* ,13:584-97, 1992.

STRYER L. *Bioquímica.* 4ª ed. Rio de Janeiro: Guanabara Koogan, 1995.

STULTZ, C. M., WHITE, J. V. & SMITH, T. F. Structural Analysis Based on State-Space Modeling. *Protein Science,* 2:305-14, 1993.

VASCONCELOS, A.T. Bioinformática: análise de banco de dados genéticos. *II Escola de Verão: Métodos Computacionais em Biologia*, 47-55, 2001.

YODA, M. Predicting the Tokyo Stock Market. In: Deboeck, G. (org.) *Trading on the Edge: Neural, Genetic & Fuzzy Systems for Chaotic Financial Markets.* John Wiley and Sons, 66-79, 1994.

ZEMLA, A., VENCLOVAS, C. & FIDELIS, K. Protein Structure Prediction Center. *http://PredictionCenter.llnl.gov/*: California, USA: Lawrence Livermore National Laboratory, last modified August 4, 2003.