Codificação de Seqüências de Aminoácidos e sua Aplicação na Classificação de Proteínas com Redes Neurais Artificiais

Thiago de Souza Rodrigues

Universidade Federal de Minas Gerais
Instituto de Ciências Biológicas
Programa de Pós-Graduação em Bioinformática

Codificação de Seqüências de Aminoácidos e sua Aplicação na Classificação de Proteínas com Redes Neurais Artificiais

Thiago de Souza Rodrigues

Orientador: *Prof. Dr. Antônio Pádua Braga*
Co-orientador: *Prof. Dr. Sérgio Costa Oliveira*
Co-orientadora: *Profª. Drª. Santuza Maria Ribeiro Teixeira*

Tese submetida à Banca Examinadora designada pelo Programa de Pós-Graduação em Bioinformática da Universidade Federal de Minas Gerais, como requisito parcial à obtenção do título de Doutor em Bioinformática.

Belo Horizonte
Abril/2007

À minha querida Dadá

e à Helenna

Agradecimentos

- À minha querida Dadá pelo apoio, paciência e constante incentivo.
- Ao meu orientador Prof. Antônio Pádua Braga, pelas horas de dedicação, pela confiança em meu trabalho, pelos conselhos e incentivos na minha vida profissional.
- Aos meus co-orientadores Profa. Santuza Maria Ribeiro Teixeira e Prof. Sérgio Costa Oliveira, pelas sugestões sempre muito relevantes para a execução do trabalho.
- Às amigas Lucilla Grossi e Fernanda Caldas pela ajuda essencial para a obtenção dos resultados.
- Aos amigos do LITC, pela ajuda nos momentos mais necessários.
- Ao Alberto Salazar pelo profissionalismo e inúmeros certificados que foram requisitados por mim.
- Aos colegas do DCC-UFLA pela liberação nos muitos dias em que foi necessário eu estar em Belo Horizonte.
- Aos integrantes da banca examinadora pelas contribuições ao trabalho.

Resumo

Este trabalho visa propor um sistema de codificação de proteínas de modo que seqüências contendo diferentes quantidades de aminoácidos possam ser convertidas em vetores de mesma dimensão para serem classificadas funcionalmente por Redes Neurais Artificiais.

O método proposto utiliza janelas deslizantes de tamanhos previamente definidos, que percorrem a seqüência a ser codificada de modo a resultar em um vetor contendo informações sobre a seqüência propriamente dita. O esquema de codificação deve resultar em vetores não ambíguos, deve considerar a similaridade entre os aminoácidos e deve considerar pequenas regiões de similaridade dando uma relevância proporcional ao tamanho da janela deslizante.

Uma comparação entre o método proposto e o método utilizado na literatura é realizada, onde sequências de aminoácidos correspondentes às proteínas de 10 bactérias foram codificadas e utilizadas para treinamento de Redes Neurais Artificiais a fim de classificar essas seqüências de acordo com as classes funcionais da base de dados do *Cluster of Orthologous Groups* (COG).

A comparação mostra a superioridade do esquema de codificação proposto visto que a informação armazenada nos vetores resultantes permitiu que as Redes Neurais Artificiais classificassem corretamente os dois conjuntos de sequências de aminoácidos de acordo com as classes funcionais do COG de várias sequências que não haviam sido anteriormente classificadas. As Redes Neurais Artificiais treinadas com os vetores gerados pelo esquema *E-SCSW* tiveram taxa de acerto que variou de $90,2\%$ à 100% para as proteínas da *Chromobacterium violaceum* e de $62,5\%$ à 100% para as proteínas da *Chlamydophila felis*.

Todas as proteínas, cujos vetores correspondentes foram classificados pelas Redes Neurais Artificiais de forma diferente com a classificação encontrada nos bancos de dados, tiveram sua classificação verificada através do alinhamento realizado

pelo *CD-Search* e a base de dados do *COG*. As Redes Neurais Artificiais treinadas com os vetores gerados pelo esquema *E-SCSW* foram capazes de reclassificar corretamente 184 proteínas da *Chromobacterium violaceum* e 94 proteínas da *Chlamydophila felis* as quais haviam sido classificadas de maneira inconsistente nos banco de dados públicos.

Este trabalho tem como principal contribuição um novo método de codificação de sequências de aminoácidos onde Redes Neurais Artificiais possam utilizar os vetores resultantes como conjunto de entrada. A verificação dos resultados mostrou que os bancos de dados públicos possuem algumas inconsistências e que as proteínas depositadas necessitam ser verificadas com uma certa freqüência. O método de codificação aqui proposto poderia portanto ser utilizado como um complemento aos métodos tradicionais de classificação de proteínas que utilizam como base o alinhamento par-a-par.

Abstract

This work aims to develop a protein coding system in which sequences with different numbers of amino acids can be converted in vectors with the same dimension to be functionally classified by Artificial Neural Networks.

The proposed scheme uses sliding windows with previous defined length. The sliding windows run over the sequence, and results in a vector containing information about the sequence. The coding method must result in unambiguous vectors, must consider the similarity between amino acids and must consider small regions with similarity in which the sliding windows must have a relevancy proporcional to their length.

In this word we presented a study of similarity and dissimilarity measure between amino acid sequences, where the pair-to-pair alignment is the metric more frequently used. Some problems using the pair-to-pair alignment to measure dissimilarity is shown , where other metrics became more effective. In other to use these metrics it is necessary a coding scheme called Sequence Coding by Sliding Window, which generates vectors with the same dimension. This coding scheme was used to classify amino acid sequences using Artificial Neural Networks.

We present a comparison between both coding schemes, in which amino acids sequences from proteins of 10 bacteria were coded and used to train Artificial Neural Networks to classify these sequences according to the *Cluster of Orthologous Groups* (COG). Two groups of sequences derived from proteins of *Chromobacterium violaceum* and *Chlamydophila feli*s were selected in other to test our method.

The comparison shows the superiority of the proposed coding scheme in which the information stored in the resulting vectors allows the Artificial Neural Networks to classify the two sets of proteins according the COG functional classes.

All sequences that were classified in a different way by the Artificial Neural Networks, had its classification verified by CD-Search alignment against the COG

data base. The results showed that some sequences are classified incoherently in the public data bases. The Artificial Neural Networks trained with the vectors generated by the *E-SCSW* scheme were able to classify correctly 184 sequences derived from *Chromobacterium violaceum* and 94 from *Chlamydophila felis*.

This work has the main contribution of developing a new protein coding method in which Artificial Neural Networks are used. The verification of the results showed that the public repositories contain some inconsistencies and that the amino acid sequences deposited should be verified in a frequent basis. The proposed codification method can thus be used as a complement to the traditional protein classification methods which are based in a par-to-par alignment.

Sumário

Lista de Figuras

Lista de Tabelas

CAPÍTULO

1

Introdução

Neste capítulo é apresentado o problemas de classificação funcional de proteínas para o qual esta tese se propõe a minimizar através do uso de Redes Neurais Artificiais. Uma visão geral de Redes Neurais Artificiais é apresentada assim como os requisitos básicos para sua aplicação na classificação funcional de proteínas. Os objetivos, geral e específicos, e a organização geral do texto são mostrados no final do capítulo.

1.1 Classificação funcional de proteínas

O crescimento do conjunto de dados referente à seqüências (nucleotídeos ou aminoácidos) teve início por volta dos anos 80, quando os métodos para seqüenciamento de DNA se tornaram largamente difundidos. Essas sequências estão acumuladas em diversos bancos de dados públicos tais como GenBank[1], EMBL (European Molecular Biology Laboratory)[2], DDBJ (DNA Data Bank of Japan)[3], PIR (Protein Information Research)[4], Swiss-Prot (Protein knowledgebase)[5], Smart (Simple Modular Architecture Research Tool)[6], CDD (Conserved Domain Database)[7],

[1] http://www.ncbi.nlm.nih.gov/Genbank/
[2] http://www.ebi.ac.uk/embl/
[3] http://www.ddbj.nig.ac.jp/
[4] http://pir.georgetown.edu/
[5] http://ca.expasy.org/sprot/
[6] http://smart.embl-heidelberg.de/
[7] http://www.ncbi.nlm.nih.gov/Structure/cdd/cdd.shtml

PFam (Protein Family)[8], COG (Clusters of Orthologous Groups)[9], dentre outros.

A Figura 1.1 mostra o número de sequências do *GenBank* desde 1983 até 2005.

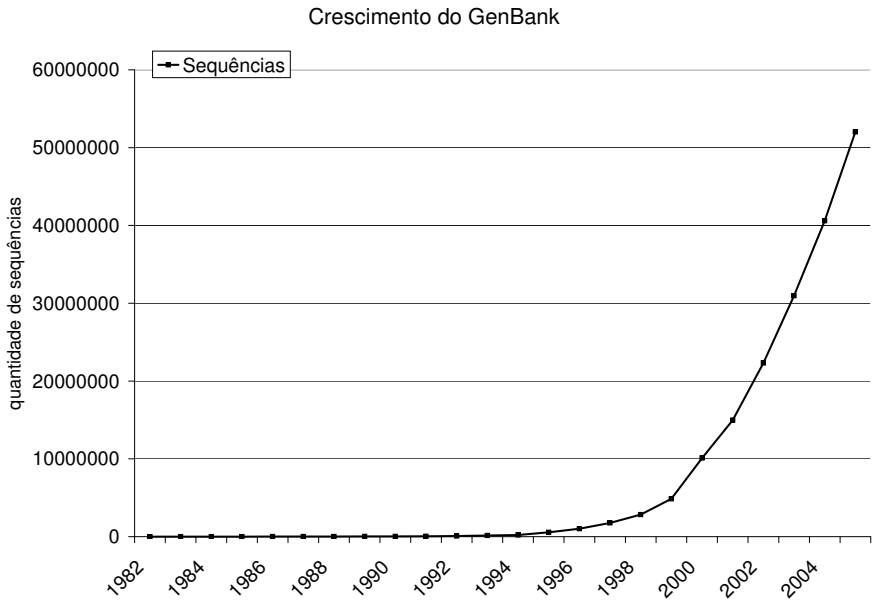


Figura 1.1: número de sequências depositadas no GenBank desde 1983 até 2005.
fonte: http://www.ncbi.nlm.nih.gov/Genbank/genbankstats.html

Adicionalmente aos bancos de dados de seqüências, métodos computacionais foram e ainda estão sendo desenvolvidos para recuperação e análise de dados como busca de similaridade, predição de estrutura, predição de função dentre outros objetivos (Kanehisa and Bork, 2003).

A geração de dados a partir do seqüenciamento do genoma tem como objetivos, dentre outros, a descoberta do conjunto de proteínas existentes no organismo em questão e a função que cada proteína desempenha. Com estas informações pode-se entender melhor o funcionamento do organismo. Após o seqüenciamento, o próximo passo é a predição do conjunto de proteínas e posterior inferência de funções. Duas estratégias podem ser utilizadas para atribuição de função a uma dada proteína: a realização de testes em laboratório ou utilização de métodos computacionais. A primeira alternativa é a mais adequada do ponto de vista de confiabilidade, entretanto demanda mais tempo e recursos. A segunda alternativa se apresenta como a mais adequada para tratamento de grandes quantidades de seqüencias, onde uma certa confiabilidade é esperada sendo a velocidade de obtenção dos resultados a principal vantagem.

[8]http://www.sanger.ac.uk/Software/Pfam/
[9]http://www.ncbi.nlm.nih.gov/COG/

Comparar seqüências é a mais fundamental operação na análise de proteínas quando se utilizam métodos computacionais. Embora uma proteína seja descrita sobre quatro aspectos relacionados à estrutura:

- estrutura primária: seqüência de aminoácidos que compõem a proteínas especificada pela ordem exata desta seqüência;
- estrutura secundária: diz respeito aos padrões regulares e repetitivos que ocorrem localmente no enovelamento do esqueleto da proteína. Os dois arranjos locais mais comuns nas proteínas são a α-hélice e a folha-β;
- estrutura terciária: diz respeito à forma tridimensional específica assumida pela proteína como resultado do enovelamento global de toda a cadeia;
- estrutura quaternária: descreve a forma com que as diferentes subunidades de uma proteína se agrupam e se ajustam para formar a estrutura total da proteína, quando esta é formada por mais de uma subunidade;

a comparação entre proteínas através de métodos computacionais normalmente é realizada através de suas *estruturas primárias*.

Quando a comparação indica a similaridade entre duas proteínas, pode-se sugerir relações envolvendo estrutura, função e evolução, sendo essas proteínas provavelmente originárias de um ancestral comum. Quando uma das proteínas é bem caracterizada, em termos de estrutura e função, essa similaridade permite que suas características sejam associadas às características da outra proteína. O grau de certeza na qual estas características podem ser associadas depende de quão similar as duas proteínas são. De qualquer forma, mesmo se a similaridade das seqüências for relativamente distante, é possível que assumam estruturas secundárias e terciárias semelhantes, sugerindo uma classificação funcional que pode servir como base para a realização de experimentos com a nova proteína (Eidhammer et al., 2004).

Sendo uma proteína composta por uma seqüência de aminoácidos, onde a comparação entre duas proteínas é realizada, em sua maioria, pelo alinhamento par-a-par (Altschul et al., 1990) (Kork et al., 2003) (Pearson, 1990) (Altschul et al., 1997) (Seção 2.1). Em um alinhamento, uma correspondência de $1:1$ é definida entre os caracteres correspondentes aos aminoácidos das duas proteínas. A cada par de aminoácidos alinhados é atribuído um *score* baseado em sua similaridade. A soma dos *scores* resulta em uma pontuação para o alinhamento, que é proporcional à

similaridade entre as duas proteínas em questão (neste trabalho o termo aminoácido é utilizado para referenciar os caracteres correspondentes a cada resíduo de aminoácido de uma proteína).

Atualmente, as proteínas são classificadas com base na ocorrência de padrões conservados de aminoácidos que definem os domínios. Bancos de dados públicos que permitem classificar proteínas de acordo com seus domínios estão disponíveis para serem consultados, onde podemos citar:

- Pfam: é um banco de dados de famílias de domínios de proteínas o qual é construído a partir de dois bancos de dados, *Pfam-A* e *Pfam-B*. *Pfam-A* é um banco de dados curado de 2700 padrões. *Pfam-B* é gerado automaticamente através das seqüências do *Pfam-A*. Para cada seqüência em Pfam-A é construído um padrão de *Hidden Markov Model* o qual é utilizado para busca em outros bancos de dados de proteínas[10];

- Blocks: um serviço do *Fred Hutchinson Cancer Research Center*, é um banco de dados gerado automaticamente de segmentos alinhados, sem *gaps*, que correspondem as mais conservadas regiões de proteínas (blocos)[11];

- Prosite: é um banco de dados de padrões conservados, o qual utiliza um padrão de consensus simples para caracterizar cada família. Os padrões não são criados automaticamente e sim selecionadas através de dados publicados[12];

- Prints: é uma coleção de domínios conservados de proteínas similar ao PROSITE, exceto pelo fato de utilizar "fingerprints"compostos por mais de um padrão que caracteriza uma proteína[13];

- COG: banco de dados de padrões de proteínas construído pela comparação de todas as proteínas de 66 genomas completos. Cada grupo consiste de proteínas originadas de, pelo menos, três genomas diferentes. O COG se baseia na premissa de que proteínas que são conservadas ao longo de, pelo menos, três genomas possuem funções conservadas ao longo da evolução[14].

A partir do seqüeciamento de um genoma, uma das principais tarefas é identificar todos os genes codificadores de proteínas para, posteriormente, identificar a

[10] http://www.sanger.ac.uk/Software/Pfam/
[11] http://blocks.fhcrc.org/
[12] http://expasy.org/prosite/
[13] http://www.bioinf.manchester.ac.uk/dbbrowser/PRINTS/
[14] http://www.ncbi.nlm.nih.gov/COG/

função de tantas proteínas quanto possível através da busca de similaridade nos bancos de dados públicos. Esta tarefa é chamada de anotação (Mount, 2004).

De acordo com (Kyrpides and Ouzounis, 1999), na avaliação da anotação é importante verificar a significância estatística dos resultados, os métodos que foram utilizados e o grau de confiança do alinhamento realizado. Sempre que necessárias as análises devem ser repetidas a fim de confirmar os resultados da anotação.

Normalmente, as análises realizadas na anotação não são repetidas com freqüência, pelo fato de que a quantidade de seqüências é elevada e esta repetição levaria muito tempo. Conseqüentemente, algumas seqüências depositadas como não-classificadas podem ter similaridade com alguma seqüência classificada recentemente, necessitando serem re-anotadas. Atualmente, existe um grande número de proteínas já depositadas que não possui nenhuma classificação, sendo importante a reavaliação destas. Adicionalmente, seqüências anotadas em uma classe podem ter sua classificação modificada pelo fato de um novo domínio, presente na proteína, ter sido identificado recentemente.

Neste trabalho propomos a aplicação de métodos de aprendizado de máquina, especificamente *Redes Neurais Artificiais* (RNAs), a fim de reavaliar seqüências já anotadas e tentar classificar aquelas ainda não classificadas, levando em conta a classificação funcional do COG.

1.2 Aprendizado de Máquina e Redes Neurais Artificiais

Um dos objetivos de um método de aprendizagem é estimar um mapeamento desconhecido a partir de um conjunto de dados de entrada e dados de saída disponíveis. Para realizar esta tarefa, basicamente duas operações são realizadas, a *Aprendizagem*, que realiza um mapeamento baseado em dados de treinamento e a *Predição*, que infere uma classificação a um conjunto de dados não apresentados no treinamento.

O aprendizado pode ocorrer de forma supervisionada ou não-supervisionada. O aprendizado supervisionado é utilizado para estimar um mapeamento desconhecido, baseado em dados de entrada/saída. Neste tipo de treinamento, os valores para a saída das amostras são conhecidos. Para o aprendizado não-supervisionado, somente os dados de entrada são fornecidos ao sistema de aprendizado. O objetivo principal do aprendizado não-supervisionado é estimar a distribuição dos dados de entrada (Braga et al., 2000).

Uma Rede Neural Artificial (RNA) é um modelo de aprendizado de máquina cujo funcionamento é baseado na estrutura do cérebro humano. São sistemas paralelos

compostos por unidades de processamento simples (neurônios), dispostas em uma ou mais camadas interligadas por um grande número de conexões (Braga et al., 2000). A Figura 1.2 mostra um exemplo de uma Rede Neural Artificial com duas camadas, uma camada intermediária com 4 neurônios e uma camada de saída com 2 neurônios, onde cada neurônio na camada intermediária tem como entrada um vetor de 3 dimensões.

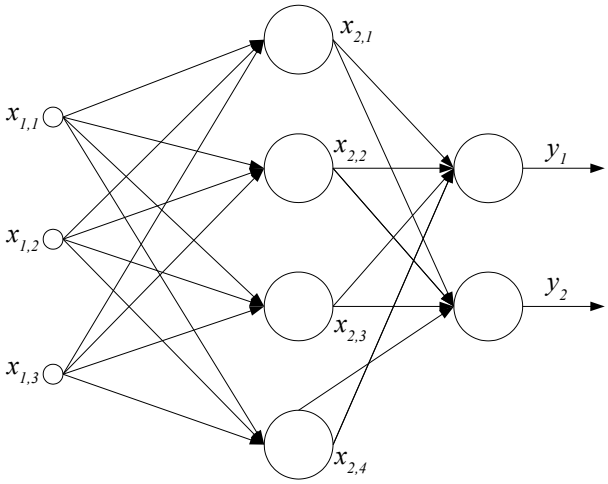


Figura 1.2: Exemplo de uma Rede Neural Artificial de duas camadas.

Neste trabalho foi utilizado o modelo *perceptron multicamadas* (Braga et al., 2000) onde cada neurônio pode ser visto como ilustrado na Figura 1.3. O vetor $X_i = [x_{i1}, x_{i2}, \cdots x_{in}]$ representa o vetor de entrada no neurônio i. Para cada elemento desse vetor existe um peso associado, representado pelo vetor $W_i = [w_{i1}, w_{i2}, \cdots w_{in}]$. A saída y_i do neurônio i é definida pela aplicação de uma função de ativação $f(.)$ ao somatório de cada elemento de entrada multiplicado pelo peso associado.

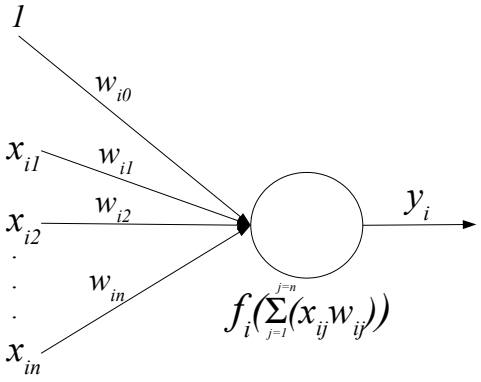


Figura 1.3: Exemplo de um neurônio do modelo perceptron.

Os pesos w_{ij} são os parâmetros da Rede Neural Artificial que devem ser ajustados de modo que a Rede Neural Artificial seja capaz de mapear os dados de entrada

de modo a fornecer uma saída o mais próxima possível da saída desejada, ou seja, o aprendizado ocorre de forma supervisionada (Braga et al., 2000) (Mount, 2004).

A aplicação de RNAs em problemas no campo de análise de sequências teve início na década de 80 onde uma RNA perceptron de camada única foi utilizada para predição de sítio de início de tradução onde a RNA foi superior aos métodos desenvolvidos anteriormente (Stormo et al., 1982), (Stormo et al.,). Posteriormente, uma RNA perceptron multicamadas foi utilizada na predição de estrutura secundária em proteínas (Bohr et al.,), (Holley and Karplus,). Outras aplicações que podem ser encontradas de RNAs é a predição de peptídeo sinal e seus sítios de cleavage (Nielsen et al.,) e em classificação de proteínas (Petrilli, 1993), (Blaisdell, 1986), (Wu et al., 1992), onde se enquadra o trabalho proposto. Existem também algumas aplicações de RNAs em seqüências de nucleotídeos como predição de genes (Snyder and Stormo, 1995), predição de introns e exons (Brunak et al., 1991) e predição de início de tradução (Pedersen and Nielsen, 1997) e (H Nielsen, 1997).

É importante perceber que, para se utilizar Redes Neurais Artificiais em uma determinada aplicação, os dados de entrada devem possuir sempre a mesma dimensão, como mostrado no Figura 1.2, onde o vetor de entrada possui dimensão 3.

Para os casos onde os dados a serem utilizados no treinamento da Rede Neural Artificial possuem valores nominais, como seqüências de nucleotídeos (alfabeto de 4 letras) e aminoácidos (alfabeto de 20 letras), cada elemento deve ser convertido em um valor numérico já que os dados de entrada da Rede Neural Artificial são, necessariamente, numéricos. Portanto algum tipo de codificação deve ser aplicada às seqüências de nucleotídeos e aminoácidos antes de serem utilizadas no treinamento da Rede Neural Artificial.

Um tipo muito simples de codificação é chamado de *codificação direta* (Baldi and Brunak, 2001), onde cada elemento do alfabeto é representado por um valor numérico, normalmente um *vetor binário* $(1,0,\ldots,0)(0,1,\ldots,0)\ldots(0,0,\ldots,1)$, de modo que cada elemento da seqüência é convertido para seu valor numérico correspondente. Entretanto, se um conjunto qualquer de seqüências for tomado para treinamento da Rede Neural Artificial a diferença de dimensão permanece, impossibilitando a aplicação deste conjunto de dados.

A Figura 1.4 mostra a quantidade de aminoácidos de um conjunto de proteínas armazenadas no banco de dados público de proteínas *COG*, onde pode ser observada a diferença de dimensionalidade entre os dados.

Uma forma de se treinar uma Rede Neural Artificial com um conjunto de seqüências de nucleotídeos ou aminoácidos é selecionar somente uma faixa das seqüên-

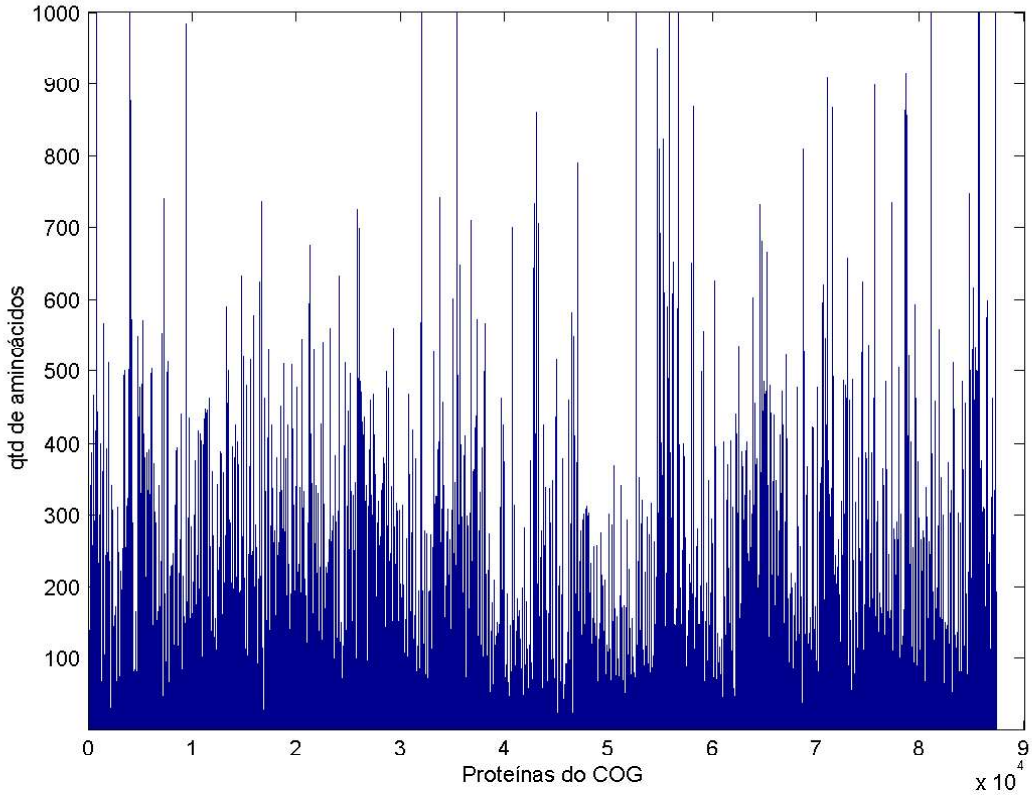


Figura 1.4: Diferença na quantidade de aminoácidos entre um conjunto de sequências pertencentes ao COG

cias, sempre de mesma dimensão, e aplicar a codificação direta. Esta metodologia é útil em algumas aplicações, como predição de início da transcrição, onde somente uma subseqüência é utilizada como dado de entrada para a Rede Neural Artificial. Entretanto, para uma classificação funcional de proteínas onde todos os resíduos de aminoácidos são relevantes, a seleção de uma faixa da seqüência original se torna inapropriada pois algum domínio importante para a função dessa proteína pode não ser selecionado, resultando em um conjunto de dados não representativo.

Um método de codificação de seqüências, chamado aqui de *Sequence Coding by Sliding Window* - (SCSW) (Blaisdell, 1986), pode ser utilizado para extrair a informação de uma seqüência completa e gerar vetores de mesma dimensão. Entretanto alguns problemas foram encontrados com o esquema *SCSW* (Seção 2.3) de modo que, neste trabalho foi proposto um novo esquema de codificação de seqüências, aqui chamado de *Extended-Sequence Coding by Sliding Window* - (E-SCSW). O objetivo do novo esquema de codificação é minimizar os problemas encontrados com o esquema *SCSW*. A comparação realizada com os dois esquemas de codificação (Seções 3.3.2 e 4.2 e) mostrou que o método proposto é mais eficiente em extrair a informação de uma sequência de aminoácidos de modo que o vetor resultante da

codificação proporciona melhores resultados no treinamento e teste de RNAs.

1.3 Objetivo geral

Como foi mostrado na Seção 2.3 o esquema de codificação *SCSW* apresenta alguns problemas como a ambigüidade, a não avaliação de pequenas regiões de similaridade e o crescimento do vetor resultante quando mais de um tamanho de janela deslizante são utilizados.

O objetivo geral deste trabalho é propor um esquema de codificação para proteínas que gere vetores de mesma dimensão, independente do tamanho das seqüências, de modo que estes vetores possam ser utilizados na classificação de proteínas com Redes Neurais Artificiais.

1.4 Objetivos específicos

O presente trabalho apresenta os seguintes objetivos específicos:

- analisar a metodologia de codificação de seqüências *SCSW* e identificar seus pontos fracos e limitações;
- propor uma nova metodologia de codificação de proteínas que solucione, ou pelo menos minimize, os problemas e pontos fracos encontrados no esquema *SCSW*;
- selecionar o conjunto de sequências aminoácidos de proteínas de 10 bactérias, já classificado de acordo com o COG, de modo a aplicar os esquemas de codificação *SCSW* e o esquema proposto. Utilizar os vetores gerados para treinamento das Redes Neurais Artificiais;
- selecionar o conjunto de sequências aminoácidos de proteínas de duas bactérias, *Chromobacterium violaceum* e *Chlamydophila felis*, a fim de testar e comparar os resultados das Redes Neurais Artificiais previamente treinadas com os vetores resultantes dos dois esquemas de codificação;
- reavaliar todas as seqüências classificadas de forma incongruente pelas Redes Neurais Artificiais, utilizando o *CD-Search* e o banco de dados do *COG*, a fim de confirmar se as proteínas reclassificadas em uma nova classe ou se uma proteína sem classificação e classificada em uma classe funcional estão corretas.

1.5 Organização do Texto

Este trabalho de tese está organizado da seguinte maneira:

- O Capítulo 2 apresenta o método de codificação de sequências *Sequence Coding by Sliding Window* (SCSW) e algumas aplicações para medir similaridade e dissimilaridade entre seqüências.
- O Capítulo 3 apresenta o esquema de codificação proposto neste trabalho, aqui chamado de *Extended-Sequence Coding by Sliding Window* a fim de minimizar os problemas encontrados com o esquema de codificação *Sequence Coding by Sliding Window*. A metodologia utilizada para comparar os dois esquemas de codificação é mostrada também neste capítulo.
- O Capítulo 4 apresenta os resultados deste trabalho de tese onde foi realizado um teste com o esquema de codificação *Sequence Coding by Sliding Window*, a fim de verificar sua eficácia em se medir a similaridade entre seqüências, e, posteriormente, a comparação entre os dois métodos de codificação de seqüências. A comparação foi realizada utilizando Redes Neurais Artificiais para classificar as seqüências codificadas de acordo com as classes funcionais do *COG*.
- Finalizando, o Capítulo 5 apresenta a discussão dos resultados encontrados assim como a conclusão deste trabalho de tese e propostas de continuidade.

CAPÍTULO

2

Esquema de Codificação *Sequence Coding by Sliding Window*

Neste capítulo é apresentado o método de alinhamento par-a-par e algumas limitações o que motivou o desenvolvimento do esquema de codificação aqui chamado de *Sequence Coding by Sliding Window* (SCSW). São apresentadas algumas aplicações do esquema *SCSW* para medir similaridade e dissimilaridade entre sequências e sua utilização na classificação funcional de seqüências utilizando Redes Neurais Artificiais. São apresentados três problemas com o esquema *SCSW* o que motivou o esquema de codificação proposto neste trabalho.

2.1 Medida de Similaridade entre Seqüências

Atualmente, uma das mais abrangentes áreas de atuação da bioinformática é a aplicação de algoritmos de alinhamento de seqüências. Baseado na questão onde seqüências que possuem uma homologia em sua composição (aminoácidos / nucleotídeos), possuem funções correlatas, alinhar corretamente duas seqüências pode revelar características, a princípio, desconhecidas. O principal método de alinhamento para busca de similaridade entre seqüências é referido como alinhamento par-a-par (Altschul et al., 1990) (Kork et al., 2003) (Pearson, 1990) (Altschul et al., 1997).

Alinhamento de seqüências par-a-par é o procedimento para comparar duas ou

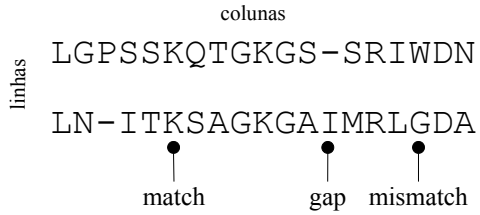


Figura 2.1: *Match*, *Mismatch* e *Gap* no alinhamento entre duas seqüências.

mais seqüências de nucleotídeos ou aminoácidos através da busca de uma série de caracteres individuais ou padrões de caracteres que estejam na mesma ordem nas seqüências comparadas. O alinhamento entre duas seqüências de caracteres pode ser visto como essas seqüências dispostas em uma matriz $2 \times n$, onde n indica o número de caracteres alinhados. Cada seqüência está disposta em uma linha da matriz e cada um de seus caracteres em uma coluna, sempre mantendo a mesma ordem. Em um alinhamento, três casos podem ocorrer em uma coluna da matriz, como mostrado na Figura 2.1:

- *Match*, onde dois caracteres idênticos aparecem na mesma coluna;
- *Mismatch*, onde dois caracteres diferentes aparecem na mesma coluna;
- *Gap*, onde um espaço aparece em uma posição da coluna correspondente;

Para o alinhamento entre seqüências de aminoácidos ou nucleotídeos o que se procura é o maior número possível de caracteres idênticos na mesma coluna. Esta operação é realizada através de inclusão de *mismatches* e *gaps*. A qualidade de um alinhamento é medida pelo *score de alinhamento* que é simplesmente a soma dos *scores* de cada caracter alinhado. O alinhamento com um *gap* também possui um *score* associado, normalmente baixo.

Deste modo, os algoritmos de alinhamento tentam encontrar o melhor alinhamento possível, considerando um padrão existente entre proteínas relacionadas.

É importante ressaltar que, freqüentemente, mais de um alinhamento é possível e algumas regiões podem alinhar muito melhor que outras regiões. Deste modo sempre as regiões com o melhor alinhamento possuem prioridade.

Existem dois tipos de alinhamentos par-a-par, alinhamento global e alinhamento local. No alinhamento global é feita uma tentativa de alinhar toda a seqüência, utilizando todos os caracteres, como mostrado na Figura 2.2(a). Seqüências que são muito similares e que possuem o mesmo tamanho são boas candidatas a este tipo de alinhamento. No alinhamento local, trechos das seqüências com a mais

alta densidade de similaridade são alinhadas gerando ilhas de sub-alinhamentos entre estas seqüências, como mostrado na Figura 2.2(b). Seqüências que são similares em certas regiões e dissimilares em outras, seqüências que diferem no tamanho ou que conservam uma certa região ou domínio são adequadas para este tipo de alinhamento.

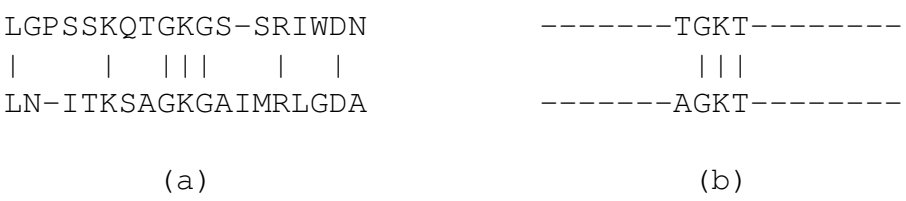


Figura 2.2: Em (a) é mostrado um alinhamento global e em (b) um alinhamento local

Existem três métodos principais de alinhamento de seqüências par-a-par:

1. Matriz Dot-plot (Gibbis and Cohen, 1970);
2. Programação dinâmica (Needleman and Wunsch, 1970) e (Smith and Waterman, 1981);
3. Método de k-tuplas (Pearson, 1990), (Altschul et al., 1990) e (Altschul et al., 1997);

Com exceção do método *Matriz Dot-Plot*, os dois outros métodos de alinhamento par-a-par medem a a qualidade do alinhamento pela soma dos *scores* de cada caracter alinhado (*match*, *mismatch* e *gap*). Para o alinhamento entre sequências de nucleotídeos, normalmente é utilizado um *score* positivo para *match* e um *score* negativo para *mismatch* e *gap*. Enquanto que, para fazer o alinhamento de proteínas, deve-se levar em consideração não só a identidade, mas também a similaridade entre os aminoácidos. Para cada par de aminoácidos existe um grau de similaridade definido por uma matriz de substituição, onde as mais utilizadas são a matriz *PAM* (Percent Accepted Mutation) e a matriz *BLOSUM* (Dayhoff, 1978) (Block Amino Acid Substitution Matrices) (Henikoff and Henikoff, 1992), como discutino na Seção 2.1.1.

Entretanto, os métodos de alinhamento par-a-par possuem duas limitações que devem ser consideradas. A primeira limitação diz respeito à medida da divergência entre sequências. Os métodos de alinhamento par-a-par buscam sempre otimizar o *score* de alinhamento entre seqüências e, além disto, este *score* é calculado com base em uma matriz de similaridade que por sua vez é definida a partir grupos de

sequências sabidamente similares. Portanto a determinação do grau de divergência entre sequências fica vinculada a uma metodologia que leva em consideração especificamente o grau de similaridade e não o grau de divergência (Vinga and Almeida, 2003). A segunda limitação diz respeito ao método de alinhamento propriamente dito. Nos métodos de alinhamento par-a-par caracteres seqüenciais e caracteres individuais possuem o mesmo valor quando é calculado o *score*. Entretanto o alinhamento de caracteres seqüenciais deveria ter um valor mais significativo, pois a subseqüência alinhada pode caracterizar um domínio relevante para a função das proteínas que estão sendo alinhadas (Vinga and Almeida, 2003).

As seqüências mostradas na figura 2.3(a) e 2.3(b) possuem os mesmos elementos alinhados resultando no mesmo score de alinhamento. Entretanto, o *score* resultante do alinhamento da figura 2.3(a) deveria ser maior, pois a seqüência de caracteres alinhados pode ser um domínio que caracteriza a função das duas seqüências.

```
A B C D E        A Z B Z C
| | |            |   |   |
A B C Z Z        A D B E C

   (a)              (b)
```

Figura 2.3: Caracteres isolados × Seqüência de caracteres

Portanto, outras métricas para medir a similaridade e a dissimilaridade entre duas seqüências foram utilizadas a fim de evitar as limitações inerentes aos métodos de alinhamento par-a-par.

2.1.1 Matriz de substituição de aminoácido

Existem duas matrizes de substituição de aminoácidos que são amplamente utilizadas para comparar duas proteínas:

- *PAM* - Percent Accepted Mutation (Dayhoff, 1978);
- *BLOSUM* - Block Amino Acid Substitution Matrices (Henikoff and Henikoff, 1992);

A matriz de substituição *PAM* mostra a probabilidade de mudança de um aminoácido para outro em proteínas homólogas durante a evolução (Arthur, 2002). A preparação da matriz de substituição *PAM* foi realizada sobre 1572 mudanças ocorridas

em 71 grupos de proteínas que possuem similaridade de, pelo menos, 85% entre si. O nome *Accepted Mutation* vem do fato de que a matriz foi construída levando-se em consideração as modificações realizadas sem interferir na função da proteína. Mais detalhes são encontrados em (Arthur, 2002),(Dayhoff, 1978).

A matriz de substituição *BLOSUM* mostra a probabilidade de mudança de um aminoácido para outro em seqüências mais divergentes em relação à *PAM*. A preparação da matriz de substituição *BLOSUM* foi realizada sobre 2000 padrões de seqüências de aminoácidos, chamados de blocos, representando em torno de 500 famílias definidas no repositório público Prosite[1]. Para cada família, os blocos foram alinhados, indicando todas as substituições que podem ocorrer para cada aminoácido. As substituições foram pontuadas e utilizadas para a preparação da matriz de substituição *BLOSUM*. Mais detalhes são encontrados em (Arthur, 2002), (Henikoff and Henikoff, 1992).

As diferenças básicas entre as duas matrizes de substituição são:

- a matriz *PAM* é baseada no modelo de mutações que ocorrem durante a evolução, levando em consideração proteínas homólogas;
- a matriz *BLOSUM* é baseada em todas as mudanças ocorridas em uma região característica de uma família de proteínas;
- a matriz *PAM* utiliza o alinhamento de todos os aminoácidos de seqüência;
- a matriz *BLOSUM* utiliza o alinhamento somente em regiões conservadas que caracteriza cada família;

Portanto, a matriz de substituição *PAM* é útil para verificar a relação evolucionária de um conjunto de proteínas, enquanto que a matriz de substituição *BLOSUM* é útil para a verificação de domínios conservados em um conjunto de proteínas.

2.2 Método alternativo para medida de similaridade

Funções de distância cujas entradas são vetores de mesma dimensão foram utilizadas em vários trabalhos para medir a similaridade entre duas seqüências (Blaisdell, 1986), (Blaisdell, 1989b), (Blaisdell, 1989a), (Wu et al., 1997) e (Petrilli, 1993). Para todas estas funções, além de os vetores de entrada possuírem a mesma

[1] http://au.expasy.org/prosite/

dimensão devem possuir também valores numéricos. Portanto seqüências de nucleotídeos e aminoácidos devem ser codificadas de modo a resultar em vetores com estas características.

Como discutido na Seção 1.2 a *codificação direta* não é adequada quando se pretende utilizar todos os caracteres da seqüência. Uma codificação alternativa, baseada na codificação proposta por (Blaisdell, 1986) e utilizada em diversos trabalhos como (Wu et al., 1997), (Petrilli, 1993), (Wu et al., 1991a), (Wu et al., 1991b), (Wu et al., 1992), (Wu, 1997), (Rodrigues et al., 2003a), (Rodrigues et al., 2003b), (Rodrigues et al., 2004) e (Rodrigues et al., 2005) resolve o problema da diferença de dimensionalidade, convertendo seqüências de dimensões diferentes em vetores de mesma dimensão. A codificação é definida da seguinte forma:

- Considerando uma seqüência qualquer S de tamanho N definida sobre um alfabeto de tamanho α;
- Uma janela deslizante w_n de tamanho $1 \leq n \leq N$ é posicionada na posição 1 da seqüência S e vai sendo deslocada até posição $N - n + 1$;
- Um vetor V_n de dimensão α^n é definido, onde cada posição corresponde a uma possível $n - tupla$ dos elementos de α;
- A cada deslocamento de w_n em S a posição de V_n correspondente à $n - tupla$ encontrada é incrementada de 1;
- Após w_n atingir a posição $N - n + 1$ em S, o vetor V_n conterá a quantidade de cada $n - tupla$ da seqüência percorrida e, independentemente do tamanho da seqüência, o vetor V_n terá dimensão α^n.

Para manter um padrão de nomenclatura, a codificação será denominada de *Sequence Coding by Sliding Window SCSW* (Rodrigues et al., 2003a), (Rodrigues et al., 2003b) e (Rodrigues et al., 2004).

A Figura 2.4 mostra um antígeno *Cs44* do *Clonorchis sinensis* (gi:4927222), proteína com 274 aminoácidos. O vetor correspondente da aplicação da codificação *SCSW* com janela deslizante de tamanho $n = 2$ à proteína da Figura 2.4 é mostrado na Tabela 2.1. Para uma melhor visualização, o vetor de tamanho 400 é apresentado em forma de uma matriz 20×20, onde cada posição corresponde a um par de aminoácidos relativos à linha e coluna daquela posição. Por exemplo, existe somente 1 subseqüência *MK* indicado pela linha *M* coluna *K*. Da mesma forma existem 24 subseqüências *AQ* indicado pela linha *A* coluna *Q*.

MKFLKLVIIGALFLNVLCLDGGAQPPKSGDGGAQPPKSGDGGAQPPKSGDGGAQPPKS
GDGGAQPPKSGDGGAQPPKSGDGGAQPPKSGDGGAQPPKSGDGGAQPPKSGDGGAQ
PPKSGDGGAQPPKSGDGGAQPPKSGDGGAQPPKSGDGGAQPPKSGDGGAQPPKSGD
GGAQPPKSGDGGAQPPKSGDGGAQPPKSGDGGAQPPKSGDGGAQPPKSGDGGAQPPK
SGDGGAQPPKSGDGGAQPPKSGAQRPFSHWIAGWFLVPLEVKASDHF

Figura 2.4: Antígeno Cs44 do *Clonorchis sinensis* - gi:4927222

Tabela 2.1: Matriz representando o vetor de 400 dimensões resultante da codificação *SCSW* aplicada à seqüência da Figura 2.4

	M	**A**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**K**	**L**	**N**	**P**	**Q**	**R**	**S**	**T**	**V**	**W**	**Y**
M	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
A	0	0	0	0	0	0	1	0	0	0	1	0	0	24	0	1	0	0	0	0
C	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
D	0	0	0	0	0	0	23	1	0	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
F	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	1	0	0	0	0
G	0	25	0	22	0	0	23	0	0	0	0	0	0	0	0	0	0	0	1	0
H	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
I	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
K	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	23	0	0	0	0
L	0	0	1	1	1	1	0	0	0	1	0	1	0	0	0	0	0	2	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
P	0	0	0	0	0	1	0	0	0	23	1	0	23	0	0	0	0	0	0	0
Q	0	0	0	0	0	0	0	0	0	0	0	0	23	0	1	0	0	0	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
S	0	0	0	1	0	0	23	1	0	0	0	0	0	0	0	0	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0
W	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Alguns trabalhos como (Hide et al., 1994) e (Blaisdell, 1989b) mostram que a busca de similaridade e dissimilaridade baseada na codificação *SCSW* é eficiente computacionalmente, e que pode encontrar características que não são levadas em consideração pelos algoritmos de alinhamento par-a-par (Pearson, 1990), (Altschul et al., 1997), (Altschul et al., 1990), (Needleman and Wunsch, 1970) (Smith and Waterman, 1981), onde seqüências de caracteres têm maior relevância que caracteres individuais quando os vetores resultantes da codificação de duas seqüências são comparados.

As Tabelas 2.2 e 2.3 mostram a codificação *SCSW* aplicada às seqüências da Figura 2.3(a) e (b) respectivamente, com janela deslizante de tamanho $n = 2$. As duplas de caracteres não representadas possuem valor 0. Se considerarmos a distância Euclidiana, Equação 1.1, os vetores da Tabela 2.2 possuem distância igual à 2 enquanto os vetores da Tabela 2.3 possuem distância igual à $2,82$. Pode-se

perceber que os vetores da Tabela 2.2 são mais "próximos"que os vetores da Tabela 2.3, dando uma maior relevância à seqüência de caracteres em comparação à caracteres isolados.

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \text{ para } x \text{ e } y \text{ com } i \text{ dimensões} \tag{1.1}$$

Tabela 2.2: *SCSW* aplicado à seqüência da Figura 2.3(a)

	AB	BC	CD	DE	CZ	ZZ
Seq 1	1	1	1	1	0	0
Seq 2	1	1	0	0	1	1

Tabela 2.3: *SCSW* aplicado à seqüência da Figura 2.3(b)

	AZ	ZB	BZ	ZC	AD	DB	BE	EC
Seq 1	1	1	1	1	0	0	0	0
Seq 2	0	0	0	0	1	1	1	1

O primeiro trabalho a utilizar a codificação $SCSW$ foi publicado por (Blaisdell, 1986). Neste trabalho foi utilizado uma janela deslizante $w_n = 1$ e $w_n = 2$ modelando as seqüências como Cadeias de Markov de ordem 1 e ordem 2, respectivamente. Seu objetivo era testar a homogeneidade de um conjunto de seqüências de nucleotídeos.

Para medir a similaridade, foi utilizada a *matriz de transição* da Cadeia de Markov que, na verdade, indica a freqüência de cada $n - tupla$ na seqüência corrente. Neste trabalho (Blaisdell, 1986) utilizou o teste χ^2 para medir a significância estatística de uma comparação específica. A métrica utilizada para medir a similaridade entre os pares de seqüências foi o *quadrado da Distância Euclidiana*. Ainda em (Blaisdell, 1986) é realizada uma comparação com o alinhamento ótimo global (Needleman and Wunsch, 1970), onde é visto que:

- o método proposto consegue medir similaridade entre duas seqüências tão dissimilares que não possam ser tratadas pelo alinhamento ótimo global;
- para seqüências dissimilares mas sendo o alinhamento possível, o mesmo resultado é encontrado pelos dois métodos. Esta conclusão também é obtida em (Blaisdell, 1989a);
- para seqüências muito correlatas, o método proposto em (Blaisdell, 1986) se mostra inferior ao alinhamento ótimo global de (Needleman and Wunsch, 1970);

Em (Wu et al., 1997) foi utilizada a codificação *SCSW* para medir a similaridade entre seqüências de nucleotídeos. Seu objetivo era avaliar a performance de três métricas, *Euclidiana*, *Standardized Euclidiana* e *Mahalanobis*.

A medida de similaridade foi realizada comparando-se uma seqüência de *mRNA de lipase lipoproteica humana* de 1612 nucleotídeos contra uma biblioteca de 30 seqüências originadas de mamíferos, invertebrados, vírus, plantas, etc. O tamanho das seqüências contidas na biblioteca variaram de 322 à 14121 nucleotídeos. Destas 30 seqüências, era sabido que 20 possuíam função relacionada com a seqüência utilizada e 10 não possuíam.

Para cada tamanho de janela deslizante, todas as comparações realizadas sobre as seqüências foram feitas sobre uma outra janela deslizante de tamanho

l_n = *min{tamanho de L, 1612}*

onde *L* denota a biblioteca de seqüências e 1612 é o tamanho do *mRNA de lipase lipoproteica humana* utilizada, ou seja, l_n varia de acordo com a biblioteca de seqüências utilizada. A janela l_n é deslocada sobre a maior seqüência da esquerda para a direita, iniciando na posição 1 e deslocando-se para a posição ml_n+1, $2ml_n+1$ até atingir a posição $N - l_n + 1$, onde N denota o tamanho da maior seqüência e m é um fator de deslocamento. A Figura 2.5 mostra como é realizada a comparação entre duas seqüências. A menor seqüência, no caso a seqüência 2, determina o tamanho da janela deslizante l_n que é posicionada na posição 1 da seqüência 1. O esquema de codificação *SCSW* é aplicado à seqüência 2 e à janela deslizante l_n. A distância entre os dois vetores resultantes é calculada e l_n é deslocada sobre a seqüência 1. Novamente o esquema de codificação é aplicado e a distância é calculada. O processo continua até que a janela deslizante atinja o final da seqüência 1. A menor distância encontrada é definida como a distância entre as duas seqüências.

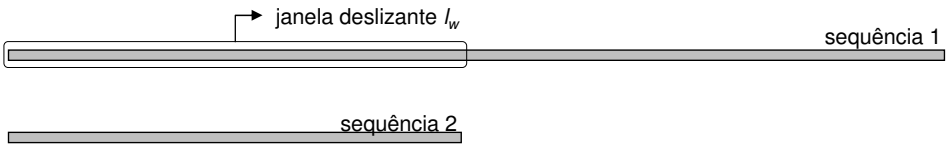


Figura 2.5: Cálculo da similaridade entre seqüências utilizada em (Wu et al., 1997).

Para o deslocamento da janela deslizante l_n, o valor de m foi escolhido de acordo com o tamanho de w_n. A tabela 2.4 mostra os valores de m correspondentes a cada w_n utilizado.

Para cada métrica e para cada valor de w_n utilizado, as 30 seqüências da biblioteca utilizada foram colocadas em ordem crescente, de acordo com as distâncias encontradas em relação ao *mRNA de lipase lipoproteica humana* utilizada. Uma medida de *sensitividade* e *seletividade* foi utilizada, sendo *sensitividade* definida como o número de seqüências relacionadas funcionalmente entre as 20 primeiras seqüências da lista e *selectividade* definida como, a partir da primeira seqüência da lista, o número total de seqüências relacionadas funcionalmente até a primeira seqüência não relacionada.

Como mostrado na Tabela 2.4, foram utilizadas janelas deslizantes de tamanho 1 até 9.

Tabela 2.4: Taxa de deslocamento de l_n

$n-word$	m utilizado
1 até 5	0,1
6	0,2
7	0,4
8	0,6
9	0,8

Em (Wu et al., 1997) é utilizado o modelo de independência dos nucleotídeos, onde a ocorrência de cada um é independente da ocorrência dos demais. Baseado neste modelo de independência, é descrito o cálculo da matriz de covariância de um conjunto de seqüências, onde a probabilidade de cada um dos quatro nucleotídeos é de $\frac{1}{4}$.

Com relação à *sensitividade* o melhor resultado obtido foi de 19 seqüências. Este valor foi obtido pelas três métricas:

- *Euclidiana* com janelas 2 e 3 (Wu et al., 1997).

  A distância Euclidiana é dada por

  $d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - yi)^2}$,

  onde x_i e y_i corresponde às posições dos vetores x e y, respectivamente;

- *Standardized Euclidiana*, com janelas 2 e 3 (Wu et al., 1997).

  A distância *Standardized Euclidiana* é definida por

  $d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - yi)^2/\sigma_{i,i}}$,

  onde x_i e y_i corresponde às posições dos vetores x e y, respectivamente e $\sigma_{i,i}$ é a variância da freqüência de cada subseqüência correspondente à x_i e y_i.

- *Mahalanobis*, com janelas 2, 3 e 5 (Wu et al., 1997).

  A distância *Mahalanobis* é definida por

  $d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - yi)\Sigma^{-1}(x_i - yi)'}$

  onde x_i e y_i corresponde às posições dos vetores x e y, respectivamente e Σ^{-1} é a pseudo-inversa da matriz de co-variância da freqüência de cada subseqüência correspondente à x_i e y_i.

Com relação à *seletividade*, o melhor resultado obtido foi 18 seqüências utilizando a métrica *Mahalanobis* com janela de tamanho 5.

Em (Wu et al., 1997) foi enfatizado a superioridade da métrica *Mahalanobis* e *Standardized Euclidean* para o problema de similaridade e dissimilaridade entre seqüências de DNA.

Entretanto, vale ressaltar que, para se comparar duas seqüências completas, o método proposto compara as seqüências com a menor quantidade de nucleotídeos com subseqüências da outra seqüência completa, como definido em l_n. Por exemplo, quando compara-se o *mRNA da apoliproteína B-100 humana* de 14121 nucleotídeos contra o *mRNA da lipase lipoproteica humana* de 1612 nucleotídeos, apenas $\frac{1}{8}$ da primeira seqüência influencia no resultado da similaridade e/ou dissimilaridade entre as duas seqüências fazendo com que alguns domínios importantes para a função da proteína resultante fique de fora da comparação.

Em (Petrilli, 1993) foi utilizada a codificação *SCSW* com uma janela deslizante $w_n = 2$, para identificação de proteínas homólogas.

Para a validação do método foi utilizado um banco de dados de 6000 proteínas. Estas foram agrupadas em superfamílias de acordo com a sua homologia e posteriormente cada superfamília foi agrupada em famílias, de acordo com a similaridade entre a composição dipeptídeo de cada proteína.

A similaridade entre as proteínas foi medida pelo *coeficiente de correlação linear* (LCC) (Petrilli, 1993).

Como enfatizado neste trabalho, a comparação entre duas proteínas diferindo muito em número de aminoácidos facilmente resulta em falso positivo, ou seja, indicação de similaridade em sequências não similares. Portanto, para a comparação entre duas proteínas, somente aquelas que diferem em 10% da quantidade de aminoácidos foram utilizadas. As seqüências com *LCC* acima de $0,3$ são consideradas homólogas.

Embora (Petrilli, 1993) tenha obtido excelentes resultados para seu conjunto de validação (100% de acerto), é destacado o problema de ambigüidade onde duas

proteínas não correlatas possuem a mesma codificação. Sendo que este problema não ocorreu em suas 6000 amostras. É sugerido a utilização de um método mais sensível para estes casos mas não se diz que método é este. É destacada também a medida de similaridade/dissimilaridade possível de se obter com o método, embora o método deva ser utilizado para uma classificação preliminar antes de se utilizar métodos mais sensitivos (Petrilli, 1993).

É importante destacar mais uma vez que nos trabalhos de (Petrilli, 1993) e (Wu et al., 1997) foram obtidos excelentes resultados embora as comparações realizadas foram sempre com seqüências do mesmo tamanho, ou tamanhos muito parecidos. Em contrapartida, os resultados obtidos por (Blaisdell, 1986) e (Blaisdell, 1989a) foram menos significativos tendo em vista que as comparações, neste trabalhos, foram feitas com seqüências de tamanhos diferentes.

Provavelmente, a utilização das métricas apresentadas sofrem alguma interferência com relação à discrepância de tamanho das seqüências, sendo que outras metodologias devem ser utilizadas para uma comparação mais genérica entre seqüências de nucleotídeos ou aminoácidos com a codificação *SCSW*.

O esquema de codificação *SCSW* também foi utilizado para classificar proteínas através de Redes Neurais Artificiais, como mostrado na Seção 2.2.1, sendo este o principal objetivo deste trabalho de tese.

2.2.1 Classificação de Proteínas com Redes Neurais Artificiais

Em (Wu et al., 1992) foi desenvolvido um sistema para a classificação de proteínas utilizando *Redes Neurais Artificiais*. O método foi chamado de *ProCANS* (*Protein Classification Artificial Neural System*) e é derivado do modelo de classificação de proteínas descrito em (Wu et al., 1991a) e em (Wu et al., 1991b).

Para o treinamento e validação, o sistema utilizou quatro bancos de dados de sequências de aminoácidos completas e classificadas, totalizando sete grupos funcionais de proteínas consistindo de 620 superfamílias e 2148 entradas, como mostrado na Tabela 2.5.

Das 2148 proteínas, 1656 foram utilizadas para treinamento e o restante das 492 proteínas para a validação, como mostrado na Tabela 2.6.

Foi construída uma *Rede Neural Artificial* para cada banco de dados, onde cada uma foi treinada com seu próprio conjunto de treinamento (557, 383, 455 e 261, respectivamente para os bancos de dados EO, TR, HY e LI). Cada proteína foi codificada a partir do método *SCSW* descrito anteriormente com o tamanho da janela w_n variável.

Tabela 2.5: Proteínas Utilizadas pelo ProCANS
fonte:(Wu et al., 1992)

Banco de Dados	Grupo Funcional	Qtd de Superfamílias	Qtd de Proteínas
EO	Transferência de Elétrons	28	385
	Oxiredutase	120	368
TR	Transferase	157	499
HY	Hidrolase	178	584
LI	Liases	66	196
	Isomerase	23	47
	Ligase	48	69

Tabela 2.6: Dados para treinamento e validação
fonte:(Wu et al., 1992)

Superfamílias	Total de Proteínas	Qtd Treinamento	Qtd Validação
Transferência de Elétrons	385	266	119
Oxiredutase	368	291	77
Transferase	499	383	116
Hidrolase	584	455	129
Liases	196	156	40
Isomerase	47	41	6
Ligase	69	64	5

Com relação à arquitetura das *Redes Neurais Artificiais* utilizadas, todas possuíam uma *camada intermediária* e uma *camada de saída*. O número de entradas depende do tamanho de W_n utilizado.

Pode-se perceber que na codificação *SCSW* o tamanho do vetor resultante cresce exponencialmente com o tamanho da janela, fazendo com que o treinamento da Rede Neural Artificial fique menos eficiente e a convergência mais demorada.

Embora o esquema de codificação resolva o problema de diferença de dimensionalidade entre seqüências de aminoácidos e nucleotídeos, a ordem das subseqüências extraídas pela janela deslizante não é preservada, como já foi levantado anteriormente. Para resolver este problema, *Wu et al., 1992* utiliza um segundo vetor, também de tamanho α^n, entretanto, cada posição é composta pela média das posições de todas as subseqüências correspondentes, normalizado entre 0 e 1.

Em (Wu et al., 1992), os vetores são utilizados de três formas: o vetor que conta o número de subseqüências somente, o vetor da posição média de cada subseqüência somente e os dois anteriores concatenados. Além disto, três alfabetos diferentes foram utilizados: tamanho 20 para os aminoácidos possíveis, tamanho 6 chamado de *exchange group*, que foi construído a partir da matriz de similaridade PAM (Dayhoff, 1978) e tamanho 2 relacionado à hidrofobicidade.

De acordo com (Wu et al., 1992), o número ótimo de nodos na camada inter-

mediária está entre 100 e 300, sendo que nos experimentos realizados em (Wu et al., 1992) as redes possuíam 200 nodos na camada intermediária.

A camada de saída depende do número de superfamílias em cada um dos quatro módulos.

O algoritmo de treinamento foi o backpropagation com momentum (Braga et al., 2000), (Haykin, 1999) onde a taxa de aprendizado foi de $0,8$ e o termo momentum de $0,3$, o treinamento foi realizado em 800 iterações.

Foram utilizados 3 valores de limiar, $0,01$, $0,3$ e $0,9$ acima dos quais as superfamílias eram identificadas. O método utilizada para a escolha dos valores de limiar não foi especificado. Para o limiar mais baixo, a taxa de acerto variou de $79,76\%$ a $90,04\%$, enquanto que a taxa de erro variou de $7,52\%$ a $15,45\%$ e padrões indefinidos variou de $0,81\%$ a $6,10\%$. Para o limiar $0,3$, a taxa de acerto variou de $73,17\%$ a $80,69\%$, enquanto a taxa de erro variou de $0,41\%$ a $2,44\%$ e padrões indefinidos variou de $18,29\%$ a $26,42\%$. E para o limiar 0.9, a taxa de acerto variou de $61,99\%$ a $69,31\%$ enquanto que a taxa de erro variou de $0,0\%$ a $0,61\%$ e padrões indefinidos variou de $30,69\%$ a $38,62\%$.

Percebe-se que, embora a taxa de acerto seja menor, a taxa de erro é praticamente nula, quando se utiliza um alto valor de limiar, no caso 0.9. Para um baixo valor de limiar, no caso $0,01$, tanto a taxa de acerto quanto a taxa de erro são altas, pelo fato de o valor de limiar estar muito próximo da fronteira entre pertencer a uma classe ou não.

De acordo com (Wu et al., 1992), os melhores resultados foram obtidos com os alfabetos de tamanho 20 (todos os aminoácidos possíveis) e de tamanho 6 (exchange group (Dayhoff, 1978)) concatenados. As janelas de tamanho 1 e 2 concatenadas e 1, 2 e 3 concatenadas obtiveram os melhores resultados respectivamente. Com relação ao vetor de posição média, sua inclusão não melhorou a performance do classificador.

De acordo com (Wu et al., 1992), seu método é ligeiramente inferior ao *FASTA* (Pearson, 1990), (Pearson et al., 1997), e que a comparação com *BLAST* (Altschul et al., 1990) está sendo realizada, mas nada foi publicado até a data atual. De acordo com (Wu et al., 1992), a acurácia do método tende a aumentar com o aumento dos bancos de dados de proteínas classificadas, adicionalmente, o método é perfeitamente adaptado à classificação de seqüências de nucleotídeos (Wu et al., 1992), (Wu, 1997).

De qualquer forma, o esquema de codificação *SCSW* para medir a similaridade entre seqüências é útil para a conversão de seqüências de diferentes dimensões em vetores de mesma dimensão, servindo como entrada para as Redes Neurais

Artificiais. Entretanto, a codificação *SCSW* apresenta alguns problemas que podem resultar em uma baixa performance da Rede Neural Artificial, como mostrado na Seção 2.3.

2.3 Problemas com o esquema de codificação SCSW

Percebe-se que o esquema de codificação *SCSW* não preserva a ordem original dos caracteres na seqüência codificada, portanto o problema de *ambigüidade* pode ocorrer, onde diferentes seqüências podem resultar em vetores idênticos. As seqüências hipotéticas da Figura 2.6 possuem a mesma codificação quando uma janela deslizante de tamanho $n = 2$ é utilizada. A Tabela 2.7 mostra os segmentos contidos em cada seqüência da Figura 2.6, representando os valores não nulos no vetor resultante da codificação *SCSW*.

A B A A A C A
A A B A A C A
A A A B A C A
A A B A C A A
A B A A C A A
A B A C A A A
A A B A C A A

Figura 2.6: Seqüências que geram vetor idênticos quando utilizada janela deslizante $n = 2$

Tabela 2.7: Número de segmentos de tamanho $n = 2$ em cada seqüência da Figura 2.6

AA	AB	BA	AC	CA
2	1	1	1	1

O problema de ambigüidade pode ser facilmente solucionado aumentando-se o tamanho da janela deslizante. Para as seqüências da Figura 2.6, a utilização de uma janela deslizante de tamanho $n = 3$ resultará em vetores diferentes para cada seqüência. A Tabela 2.8 mostra os valores não nulos no vetores resultantes da codificação *SCSW*. Portanto, para uma janela suficientemente grande, o problema de ambigüidade não existe.

Tabela 2.8: Número de segmentos de tamanho $n = 3$ para cada seqüência da Figura 2.6

	ABA	BAA	AAA	AAC	ACA	AAB	BAC	CAA
seq1	1	1	1	1	1	0	0	0
seq2	1	1	0	1	1	1	0	0
seq3	1	0	1	0	1	1	1	0
seq4	1	0	0	0	1	1	1	1
seq5	1	1	0	1	1	0	0	1
seq6	1	0	1	0	1	0	1	1
seq7	1	0	0	0	1	1	1	1

Percebe-se que a dimensão dos vetores resultantes do esquema de codificação *SCSW* aumenta exponencialmente com o tamanho da janela deslizante, aumentando, conseqüentemente, o custo computacional para a manipulação destes vetores. Portanto, é importante determinar a menor janela deslizante de modo que não haja ambigüidade.

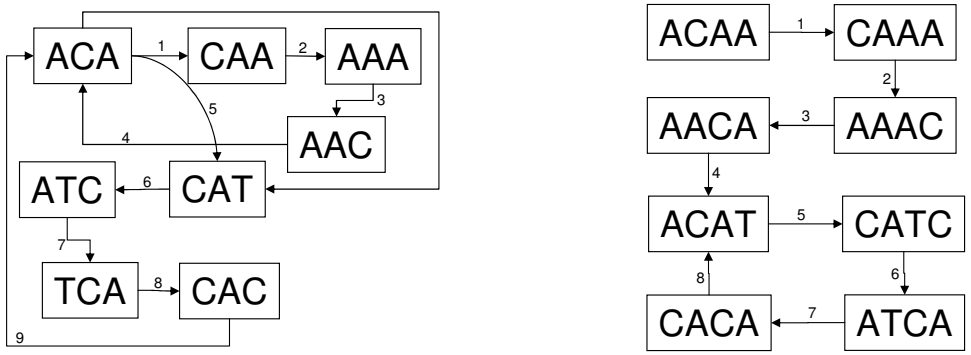
Em (Reinert et al., 2000) é apresentado o problema de seqüenciamento por hibridização, onde o objetivo é determinar a seqüência de DNA a partir de uma lista desordenada de *n-tuplas*. A principal dificuldade do seqüenciamento por hibridização é que mais de uma seqüência pode produzir o mesmo conjunto de *n-tuplas*, caracterizando a ambigüidade. Em (Reinert et al., 2000) é proposto um método para verificar se, a partir de conjunto desordenado de *n-tuplas*, uma seqüência é *unicamente* reconstruída. É utilizado um grafo chamado *Bruijn-graph* que é construído da seguinte maneira:

- Considere todas as *n-tuplas* geradas a partir de uma seqüência de caracteres;
- As *(n-1)-tuplas* são vértices do grafo, sem repetição;
- Para todas as *n-tuplas*, o vértice v, correspondente aos primeiros *n-1* caracteres, é conectado ao vértice w, correspondente aos últimos *n-1* caracteres, por uma aresta direcionada de v para w;

Uma seqüência é unicamente reconstruída a partir do seu conjunto de *n-tuplas* se e somente se existir um único caminho *Euleriano* conectando todos os vértices do grafo.

A Figura 2.3(a) mostra o *Bruijn-graph* construído a partir do conjunto das *4-tuplas* da seqüencia *ACAAACATCACAT*, onde as arestas direcionadas estão rotuladas por números arábicos. Existem dois caminhos Eulerianos conectando todos

os vértices, os caminhos *1-2-3-4-5-6-7-8-9* e *5-6-7-8-9-1-2-3-4*. Conseqüentemente duas seqüências podem ser reconstruídas a partir do conjunto de *4-tuplas*, *ACAAA-CATCACAT* e *ACATCACAAACAT*. A Figura 2.3(b) mostra o *Bruijn-graph* construído a partir do conjunto das *5-tuplas* da mesma seqüência *ACAAACATCACAT*, onde as arestas também estão rotuladas por números arábicos. Neste caso existe somente um único caminho Euleriano conectando todos os vértices do grafo, *1-2-3-4-5-6-7-8*. Portanto o tamanho ideal da janela deslizante é 5 pois somente uma seqüência é reconstruída, ou seja, não existe ambigüidade.



(a) (b)

Figura 2.7: Em (a)-Bruijn-graph construído com *4-tuplas* e em (b)-Bruijn-graph construído com *5-tuplas*

Em (Pevzner, 1995) foi mostrado que existem exatamente três casos em que a ambigüidade entre seqüências pode aparecer, de modo que não é necessário construir o *Bruijn Graph*. Considerando que queremos verificar se há ambigüidade utilizando janela deslizante de tamanho n, os três casos são listados a seguir.

O primeiro caso ocorre quando existe repetição de dois pares de *(n-1)-tuplas*. Como na seqüência $S_1 = Y_1Z_1Y_2Z_2Y_3Z_1Y_4Z_2Y_5$, onde Z_1 e Z_2 são *(n-1)-tuplas* que se repetem e $Y_1, \ldots, Y_5$ são strings. Tanto a string Y_2 quanto a string Y_4 são precedidas de Z_1 e seguidas de Z_2, logo a troca de posições entre elas não vai afetar a composição de *n-tupas*, entretanto, a seqüência resultante será diferente da seqüência original acarretando em ambigüidade. As strings Y_1, Y_3 e Y_5 podem ser $\emptyset$ e as strings Y_2 e Y_4 devem ser diferentes. Considere a seqüência da Figura 2.8(a) onde as subseqüências *CGA* e *CTA*, em negrito, se repetem. As subseqüências *AT* e *GA*, em cinza, podem ser trocadas de lugar resultando na seqüência da Figura 2.8(b). A composição de subseqüências de tamanho 3 permanece a mesma, resultando em ambigüidade para janela deslizante de tamanho $n = 3$.

A**CGA**AT**CTA**T**CGA**GA**CTA**A
(a)

A**CGA**GA**CTA**T**CGA**AT**CTA**A
(b)

Figura 2.8: Caso 1 para verificação de ambigüidade.

O segundo caso ocorre quando existem três repetições de uma *(n-1)-tupla*. Como na seqüência $S_2 = Y_1ZY_2ZY_3ZY_4$, onde Z é uma *(n-1)-tupla* e $Y_1, \ldots, Y_4$ são strings. Tanto a string Y_2 quanto a string Y_3 são precedidas e seguidas de Z, logo Y_2 e Y_3 podem ser trocadas de lugar na seqüência que a composição de *n-tuplas* não será afetada de modo a resultar em ambigüidade. As strings Y_1 e Y_4 podem ser $\emptyset$ e as strings Y_2 e Y_3 devem ser diferentes. Considere a seqüência da Figura 2.9(a) onde a subseqüência *CGA*, em negrito, repete 3 vezes. As subseqüências *ATC* e *AT* podem ser trocadas de lugar resultando na seqüência da Figura 2.9(b). A composição de subseqüências de tamanho 3 continua a mesma resultando em ambigüidade para janela deslizante de tamanho $n = 3$.

AG**CGA**ATC**CGA**AT**CGA**GAA
(a)

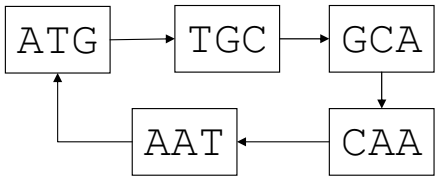
AG**CGA**AT**CGA**ATC**CGA**GAA
(b)

Figura 2.9: Caso 2 para verificação de ambigüidade.

O terceiro caso ocorre quando uma seqüência é iniciada e terminada com a mesma *(n-1)-tupla*. Como na seqüência $S_3 = Z_1Y_1Z_2Y_2Z_1$, onde Z_1 e Z_2 são *(n-1)-tuplas* e Y_1 e Y_2 são strings. Se considerarmos a construção do *Bruijn Graph*, existirá um ciclo, logo qualquer vértice pode ser escolhido como início do caminho Euleriano. Considere a seqüência da Figura 2.10(a) iniciando e terminando com a subseqüência *ATG*. O *Bruijn Graph* correspondente construído para janela deslizante de tamanho $n = 4$ é mostrado na Figura 2.10(b). Pode-se perceber que o grafo é um ciclo e qualquer vértice pode ser tomado como início para o caminho Euleriano, logo mais de um caminho é possível resultando em ambigüidade. A Figura 2.10(c) mostra as seqüências que geram ambigüidade com a seqüência original obtidas a partir do percurso do *Bruijn Graph*.

ATGCA**ATG**

(a)



(b)

TGCAATGC
GCAATGCA
CAATGCAA
AATGCAAT

(c)

Figura 2.10: Caso 3 para verificação de ambigüidade (a), o *Bruijn Graph* correspondente (b) e as seqüências ambíguas obtidas pelo *Bruijn Graph* (c).

Logo podemos encontrar o menor tamanho de janela deslizante de modo que o problema de ambigüidade não ocorra. De acordo com os resultados mostrados na Seção 3.1, com o aumento do tamanho da janela deslizante w_n, a similaridade entre subseqüências menores que n é ignorada, conseqüentemente, pequenas regiões de similaridade não são avaliadas. Esta não avaliação de subseqüências pode ser mostrada considerando as três seqüências hipotéticas da Figura 2.11.

ACE
ACH
YQP

Figura 2.11: Similaridade desconsiderada entre subseqüências

Sendo a janela deslizante utilizada de tamanho $n = 3$, a distância entre os vetores resultantes da codificação será a mesma, embora as seqüências ACE e ACH tenham claramente um maior grau de similaridade devido à subseqüência AC. Portanto vários tamanhos de janelas devem ser considerados, tanto para evitar a ambigüidade quanto para considerar pequenas regiões de similaridade.

Em (Wu et al., 1992) mais de um tamanho de janela deslizante é utilizado, sendo que o vetor resultante é a concatenação dos vetores gerados por cada janela deslizante. Isto faz com que pequenas regiões de similaridade sejam consideradas; entretanto, a dimensionalidade dos vetores aumenta a medida que uma maior quantidade de janelas deslizantes sejam utilizadas. Outra observação que deve

ser feita com relação ao esquema de codificação *SCSW* é que, independente dos tamanhos das janelas deslizantes utilizadas, o peso associado a cada uma é sempre o mesmo. Entretanto janelas maiores deveriam possuir um peso maior, pois indicam uma maior identidade entre duas seqüências quando estas possuem a mesma subseqüência associada.

Neste trabalho de tese é proposto um esquema de codificação de seqüências chamado *Extended-Sequence Coding by Sliding Window* (*E-SCSW*) a fim de minimizar os problemas levantados com o esquema *SCSW*.

CAPÍTULO 3

Metodologia

Neste capítulo será apresentada a metodologia para testar o esquema de codificação *SCSW* aplicado a um conjunto de sequências de aminoácidos correspondentes à 112 proteínas, de modo que os vetores resultantes serão agrupados pelo método *K-Médias*. É apresentada também nossa proposta de codificação de seqüências para solucionar ou minimizar os problemas com o esquema *SCSW* apresentados na Seção 2.3, assim como a metodologia utilizada para comparar os dois esquemas de codificação de sequências utilizando Redes Neurais Artificiais como ferramenta de comparação.

3.1 Teste do esquema de codificação SCSW

Com a finalidade de comprovar a eficiência do esquema de codificação *SCSW* (Wu et al., 1992; Blaisdell, 1986) foram selecionados 112 antígenos de 19 diferentes helmintos disponíveis no banco de dados público do *National Center for Biotechnology Information* (NCBI)[1]. A Tabela 3.1 apresenta os 19 helmintos e o correspondente número de proteínas cujas sequências de aminoácidos foram selecionadas, resultando em 112 sequências.

[1] http://www.ncbi.nlm.nih.gov/

Tabela 3.1: Helmintos e correspondente número (n) de proteínas cujas sequências de aminoácidos foram utilizadas para testar o esquema de codificação *SCSW*.

Helminto	n	Helminto	n
Taenia solium	18	*Trichinella spiralis*	02
Taenia ovis	07	*Taenia crassiceps*	01
Schistosoma japonicum	13	*Fasciola hepatica*	04
Schistosoma haematobium	01	*Nippostrongylus brasiliensis*	04
Echinococcus multilocularis	13	*Clonorchis sinensis*	03
Echinococcus granulosus	22	*Ascaris suum*	02
Trichostrongylus colubriformis	02	*Toxocara canis*	01
Paragonimus westermani	01	*Onchocerca volvulus*	11
Trichuris trichiura	01	*Taenia asiatica*	05
Wuchereria bancrofti	01		

O número de aminoácidos de cada uma das seqüências pode ser observado na Figura 3.1, onde algumas seqüências possuem menos de 60 aminoácidos e outras com tamanhos que variam de 400 a 800 aminoácidos. A Figura 3.1 mostra a impossibilidade em se aplicar uma codificação direta a fim de utilizar todos os aminoácidos das seqüências como dados de entrada de algum método que utilize vetores de mesma dimensão (Wu, 1997).

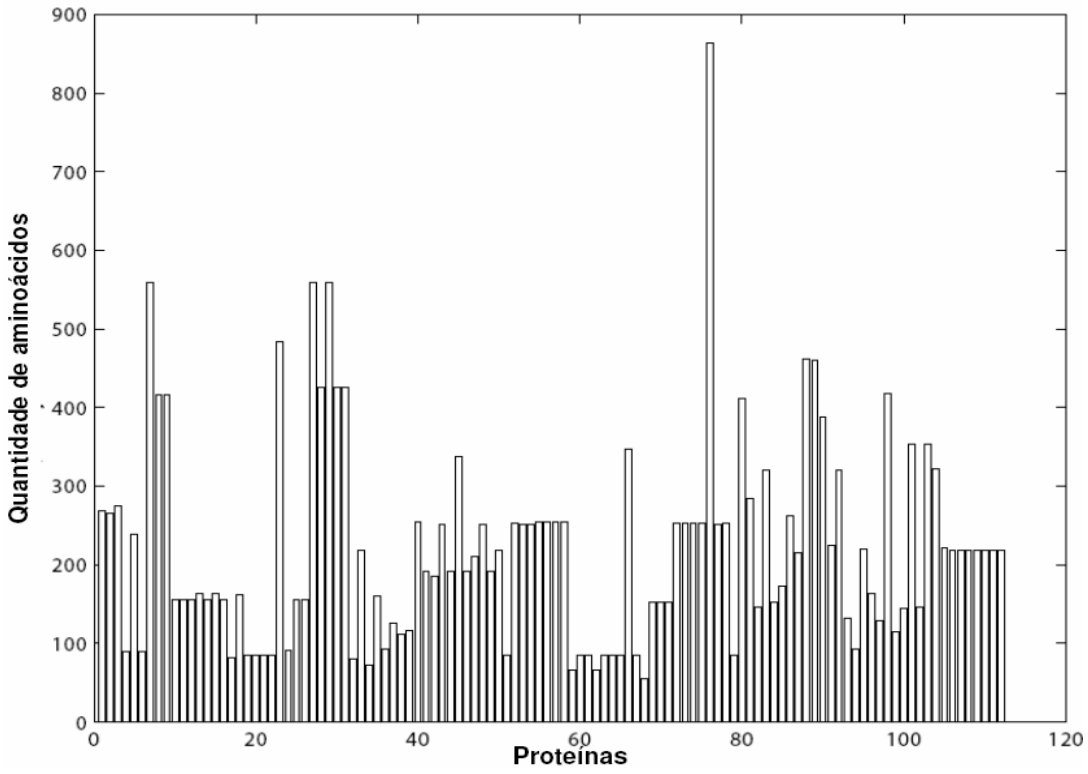


Figura 3.1: Número de aminoácidos correspondente à cada uma das 112 seqüências analisadas.

A fim de encontrar alguma regularidade entre os dados de entrada, os gráficos das Figuras 3.2 e 3.3 foram gerados. Na Figura 3.2, o número de ocorrências de cada resíduo de aminoácido em todas as seqüências é apresentado e na Figura 3.3, a concentração dos aminoácidos ao longo das seqüências é mostrada.

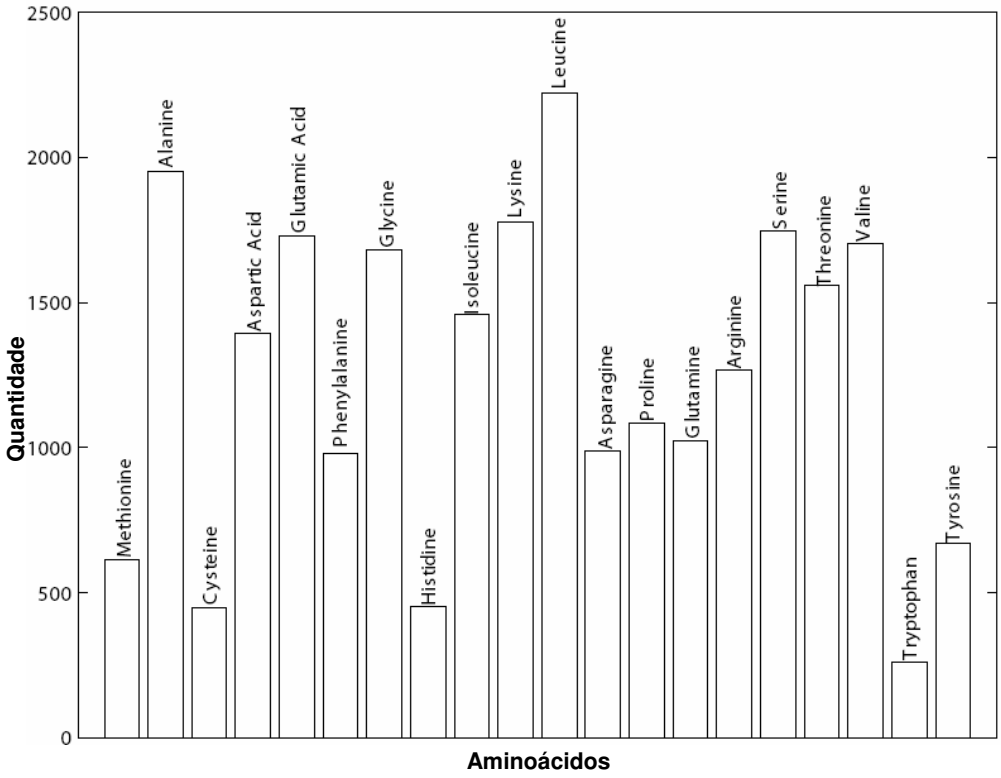


Figura 3.2: Quantidade de cada aminoácido que compõe as 112 seqüências analisadas.

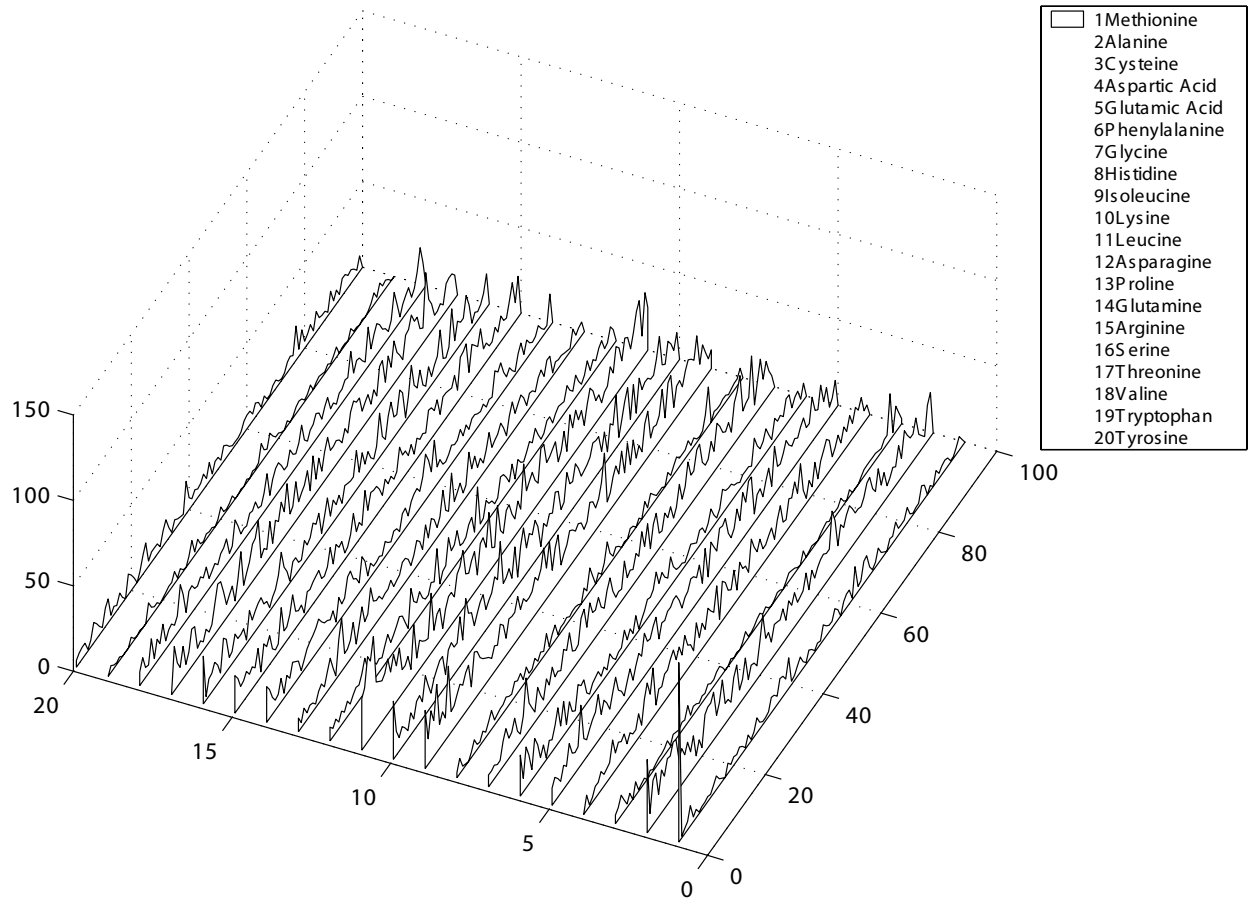


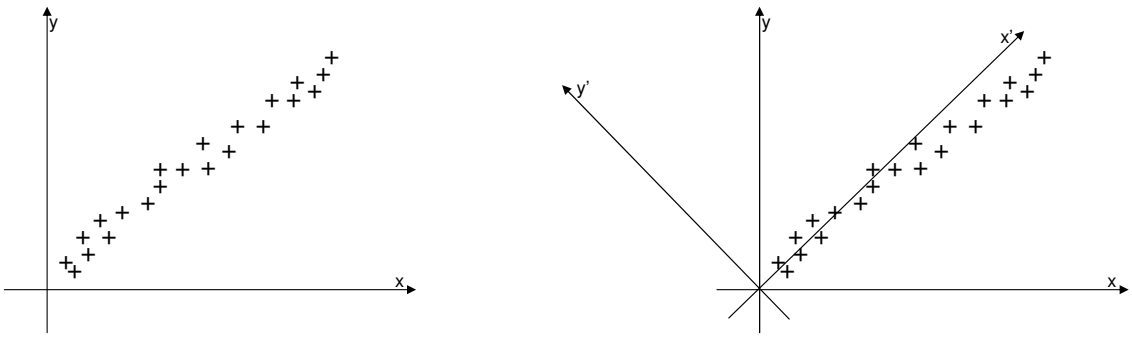
Figura 3.3: Distribuição de cada aminoácido ao longo das 112 sequências analisadas.

Observa-se nas Figuras 3.2 e 3.3 que não existe nenhum resíduo de aminoácido proeminente ao longo das sequências de aminoácidos analisadas e que a *alanina*, *lisina* e *leucina* aparecem am altas concentrações mas bem distribuídas ao longo das seqüências, como já era esperado (Stryer et al., 2002). Nenhuma observação relevante relacionada às seqüências foi realizada, deste modo, a codificação *SCSW* foi aplicada.

Foi utilizada uma janela deslizante de tamanho $n = 2$ para todas as seqüências, resultando em uma matriz com 112 linhas e 400 colunas, onde cada linha representa uma seqüência de aminoácidos codificada em um vetor de 400 dimensões, de acordo com o tamanho da janela deslizante utilizada.

A fim de reduzir a dimensão dos vetores resultantes do esquema de codificação *SCSW* de modo a melhorar a performance no processamento destes vetores, foi utilizado o método estatístico *Principal Component Analysis* (PCA) (Cherkassky and Mulier, 1998), (Haykin, 1999). O *PCA* transforma os dados para um novo sistema de coordenadas tal que a maior variância de qualquer projeção desses dados se torne a primeira coordenada, a segunda maior variância a segunda coordenada, e assim sucessivamente. A Figura 3.4 ilustra o funcionamento do *PCA*. A Figura 3.4(a) mostra um conjunto de pontos bidimensionais projetados sobre os eixos x e y. Após a aplicação do *PCA* o eixo x é projetado de tal forma que os dados tenham uma maior variância sobre ele onde o eixo y acompanha a projeção. A Figura 3.4(b) mostra o novo sistema de coordenadas *(x',y')*. Se somente o valor de cada ponto referente ao eixo *x'* for tomado haverá uma pequena perda de informação relativa ao eixo *y'* entretanto a informação com maior variância será preservada. Para o cálculo do *PCA* foi utilizada a função *SVD* disponibilizada pelo Matlab[2], onde a matriz de covariância e correlação são calculadas de forma adaptativa.

[2]http://www.mathworks.com/



(a) (b)

Figura 3.4: Exemplificação do funcionamento do *PCA*. Em (a) é mostrado o sistema de coordenadas original e em (b) o novo sistema de coordenadas após a aplicação do *PCA*.

A Figura 3.5 mostra os valores da variância do resultado da aplicação do *PCA*. Quanto maior o valor da variância, mais informação está armazenada na coordenada correspondente. Coordenadas com variância nula não contém informação relevante sobre a distribuição dos dados de entrada. Como mostrado na Figura 3.5, existem variâncias não nulas até a dimensão 73, indicando que os vetores de 400 dimensões podem ser transformados em vetores de 73 dimensões. Para obter um redução ainda maior dos vetores de entrada haverá alguma perda de informação, ficando o ponto de corte um parâmetro definido pelo pesquisador. Nos testes realizados com o conjunto de antígenos previamente selecionados, a manipulação de dimensões maiores ou iguais à 5 não resultou em mudança no resultado do agrupamento realizado à posteriori. Portanto, com a aplicação do *PCA*, foi possível reduzir a dimensão dos dados de entrada de 400 para 5 dimensões, $1,25\%$ da dimensão original.

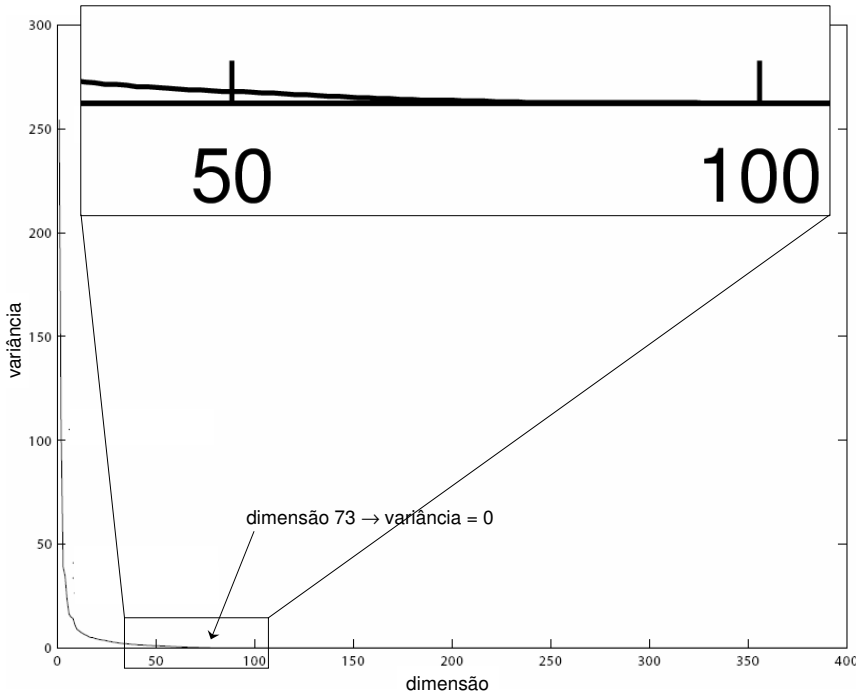


Figura 3.5: Variância correspondente a cada dimensão após a aplicação do *PCA*. A variância possui valor 0 a partir da dimensão 73, ou seja, não existe perda de informação a partir desta dimensão.

Depois de obtidos os 112 vetores de 5 dimensões pela aplicação do *PCA*, o conjunto de dados foi agrupado em 40 diferentes grupos. O método utilizado para o agrupamento foi o algoritmo *K-Médias* (Likas et al., 2003) (Braga et al., 2000) cujo objetivo consiste em encontrar K conjuntos de dados com variância mínima . O algoritmo pode ser dividido nos seguintes passos:

1. Define-se o número de grupos *K*;
2. Define-se os *K* centróides arbitrariamente;
3. Iteração para cada amostra:
   - Procura-se o centróide mais próximo de acordo com uma métrica previamente definida, no nosso caso foi utilizada a distância Euclidiana;
   - Atribui-se a amostra ao grupo correspondente;
4. Recalcula o centróide;
5. Volta-se ao passo 3 até um critério de convergência ser cumprido, no nosso caso, até que nenhum ponto mude de classe.

A Figura **??** mostra os passos da execução do algoritmo *K-Médias*, onde é selecionado o valor 2 para o número de clusters a serem encontrados.

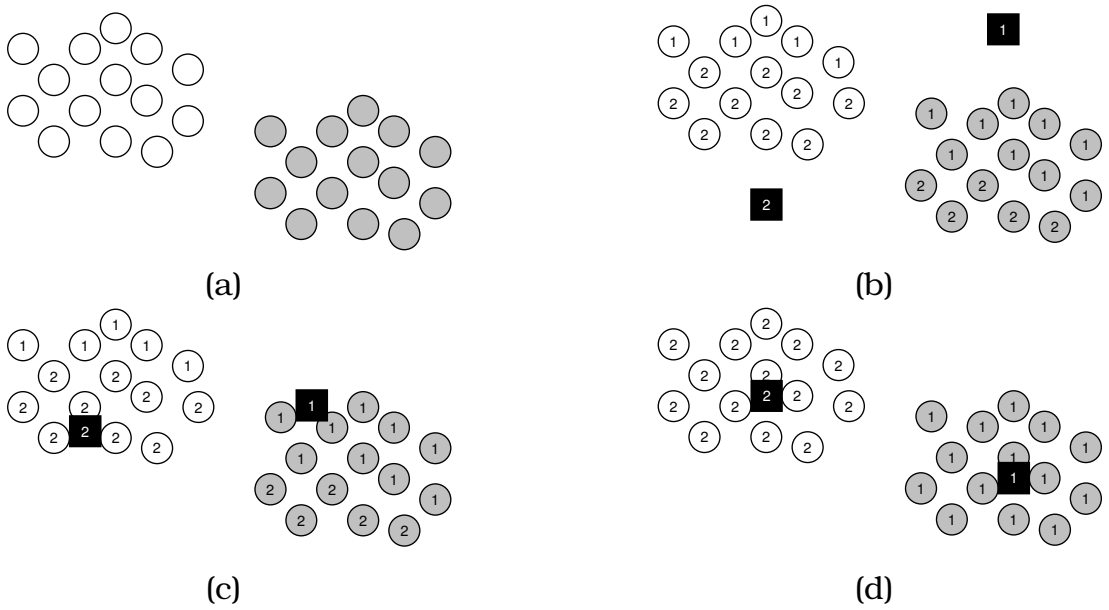


Figura 3.6: Execução do algoritmo *K-Médias*. Em (a) é dado o conjunto de pontos a serem agrupados. Em (b) são definidos 2 centróides arbitrariamente, cada ponto é associado ao centróide mais próximo. Em (c) os centróides são recalculados e o algoritmo é continuado até que algum critério de convergência seja alcançado. Em (d) é mostrado o resultado final do algoritmo, com os 2 grupos definidos.

O algoritmo *K-means* foi definido para encontrar 40 grupos, o melhor valor encontrado de modo que o resultado do agrupamento seja compatível com os domínios definidos no *PFAM*[3].

Os agrupamentos obtidos pelo *K-means* foram comparados com o resultado do alinhamento múltiplo das 112 sequências de aminoácidos realizado pelo *ClustalW*[4]. Alinhamento múltiplo é comumente aplicado ao alinhamento de seqüências de aminoácidos, sendo um registro de similaridade evolucionária e estrutural entre as sequências presentes no alinhamento (Gibas and Jambeck, 2001). O alinhamento par-a-par é utilizado no alinhamento múltiplo de sequências, onde a *estratégia progressiva* é utilizada pelo *ClustalW*. Nessa estratégia um par de sequências é selecionada para ser alinhada pelo alinhamento par-a-par. Cada seqüência subseqüente é alinhada com todas as seqüências alinhadas anteriormente. No final do alinhamento as sequências são dispostas em um dendograma de acordo com o score de alinhamento obtido (Gibas and Jambeck, 2001)

[3]http://www.sanger.ac.uk/Software/Pfam/
[4]http://www.ebi.ac.uk/clustalw/

Como é mostrado no capítulo 4, o esquema de codificação *SCSW* se mostrou útil para a determinação de similaridade entre sequências mas a metodologia apresentada possui uma acurácia inferior aos métodos tradicionais de alinhamento par-a-par (BLAST). Por este motivo propomos um novo esquema de codificação, aqui chamado de *Extended-Sequence Coding by Sliding Window*, detalhado na seção a seguir.

3.2 Extended-Sequence Coding by Sliding Window

Para a resolução do problema de ambigüidade sem desconsiderar a similaridade entre subseqüências e evitando o crescimento do vetor resultante quando se utiliza mais de um tamanho de janela deslizante, nossa proposta é uma adaptação do esquema de codificação *SCSW*, onde é utilizado mais de um tamanho de janela deslizante, sendo associado a cada uma um peso proporcional ao seu tamanho. Esta nova codificação é chamada de *Extended-Sequence Coding by Sliding Window* (E-SCSW), e consiste de:

- para um dado conjunto de seqüências, definir o tamanho mínimo da janela deslizante de modo a não existir ambigüidade. Esse é o maior tamanho de janela deslizante a ser utilizada sendo definido como max;
- as janelas deslizantes a serem utilizadas possuem tamanhos possuem tamanhos max, $max - 1$, ..., mim sendo mim definido pelo usuário;
- para uma seqüência qualquer S de tamanho N definida sobre um alfabeto de tamanho α;
- um vetor V_{max} de dimensão α^{max} é definido, onde cada posição corresponde a uma possível $tupla$ de tamanho max dos elementos do alfabeto;
- para cada janela deslizante w_i, $i = max, max - 1, \ldots, mim$:
  - a janela deslizante w_i é posicionada na posição 1 da seqüência S e vai sendo deslocada até posição $N - i + 1$;
  - para cada subseqüência de tamanho i encontrada, todos os elementos em V_{max}, onde os $i's$ primeiros elementos são encontrados, são incrementados com um peso E_i;
  - onde, $E_{max} > E_{max-1} > \ldots > E_{min}$.

Para a definição do valor do peso E_i para cada tamanho de janela deslizante, o seguinte método é aplicado:

- um *score* é estabelecido para cada *identidade* entre os caracteres do alfabeto α.
- para cada subseqüência encontrada por uma janela deslizante w_n, o peso associado é a soma dos *scores* de identidade de cada caracter na subseqüência.

As Figuras 3.7 e 3.9 mostram o vetor resultante da aplicação do esquema de codificação *E-SCSW* à seqüência *S=ABAAB* gerada a partir do alfabeto $\alpha = \{A, B\}$, com janelas deslizantes de tamanhos $k_{max} = 3$ e $k_{min} = 2$. O valor do peso para cada janela w_k foi determinado usando-se o score 1 para identidade.

De acordo com o esquema de codificação *E-SCSW*, o vetor resultante possui dimensão $2^3 = 8$. A Figura 3.7 mostra a janela deslizante de tamanho $k = 3$ aplicada à seqüência *S*. Para cada subseqüência de tamanho $n = 3$ encontrada, a posição correspondente no vetor resultante é incrementada pela soma dos scores de identidade de cada caracter da subseqüência. A Figura 3.8 mostra as subseqüências encontradas e os *scores* correspondentes.

AAA	AAB	ABA	ABB	BAA	BAB	BBA	BBB
0	3	3	0	3	0	0	0

Figura 3.7: Janela deslizante $k = 3$ aplicada à *S=ABAAB*.

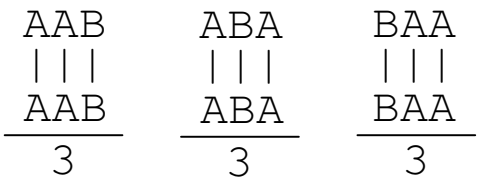


Figura 3.8: *Scores* referentes às subseqüências de tamanho $n = 3$ encontradas na seqüência original

A Figura 3.9 mostra a aplicação subseqüênte da janela deslizante de tamanho $k = 2$ à seqüência S. Para cada subseqüência de tamanho $n = 2$ encontrada, as posições no vetor resultante correspondentes às subseqüências que possuem os $n = 2$ primeiros caracteres são incrementadas pela soma dos scores de identidade da subseqüência encontrada. A Figura 3.10 mostra a subseqüência *AB*, encontrada na seqüência original, e as subseqüências cujas posições no vetor resultante serão incrementadas pelo score de similaridade.

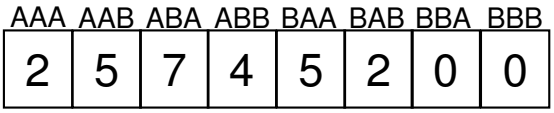


Figura 3.9: Janela deslizante $k = 2$ aplicada à *S=ABAAB* após a aplicação da janela deslizante $k = 3$.

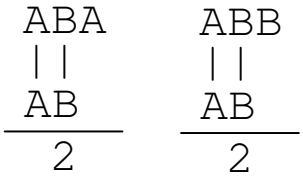


Figura 3.10: *Score* referente à subseqüência *AB* encontrada na seqüência original

Aplicando-se o esquema de codificação *E-SCSW* com o tamanho da janela deslizante apropriado o problema de ambigüidade pode ser evitado sem ignorar a identidade entre subseqüências menores que a janela deslizante. O peso associado à cada tamanho de janela tem o objetivo de dar uma maior importância às subseqüências mais extensas, como discutido na Seção 2.3.

Normalmente, quando se aplica técnicas de bioinformática a uma proteína ou a um conjunto de proteínas, o alfabeto utilizado é o dos aminoácidos (20 caracteres). A principal razão disto é que os bancos de dados públicos de seqüências disponibilizam as proteínas em sua forma primária[5] [6].

[5]http://www.ncbi.nlm.nih.gov/NCBI,
[6]http://www.ebi.ac.uk/

Entretanto, quando o alfabeto de aminoácidos é utilizado dois problemas surgem mediante a aplicação do esquema de codificação *E-SCSW*.

- O esquema de codificação *SCSW* e *E-SCSW* gera vetores cuja dimensão aumenta exponencialmente com o aumento do tamanho da janela deslizante. Para evitar o problema de ambigüidade, é necessário uma janela deslizante grande o suficiente, resultando em vetores de alta dimensão (20^n onde n é o tamanho da janela deslizante). Esta alta dimensionalidade faz com que o tempo computacional seja muito alto para a manipulação destes vetores, sendo conveniente a redução desta dimensão.

- Outro problema ocorre quando dois vetores gerados pela codificação *E-SCSW* são comparados. Somente os aminoácidos idênticos são considerados, entretanto existem similaridades entre eles que devem ser consideradas (Dayhoff, 1978) e (Henikoff and Henikoff, 1992).

Para solucionar estes problemas o tamanho do alfabeto pode ser reduzido, agrupando os aminoácidos similares em um sub-alfabeto, mesmo este agrupamento resultando em perda de informação dos aminoácidos que compõe uma dada seqüência. Os aminoácidos possuem uma grande variedade de propriedades tais como massa, polaridade e hidrofobicidade (Zvelebil et al., 1987), portanto muitos agrupamentos são possíveis. Neste trabalho foi utilizado o agrupamento chamado *Exchange group* (Wu et al., 1992) baseado na matriz de similaridade PAM (Dayhoff, 1978), onde os aminoácidos são agrupados em 6 grupos (Tabela 3.2). A escolha do *Exchange group* se deve ao fato de que os melhores resultados em (Wu et al., 1992) foram obtidos utilizando este alfabeto e o alfabeto de 20 caracteres.

Tabela 3.2: Agrupamento dos 20 aminoácidos de acordo com o *Exchange-group*

H, R, K
D, E, N, Q
C
S, T, P, A, G
M, I, L, V
F, Y, W

Desta forma, a dimensão dos vetores resultantes do esquema de codificação *E-SCSW* é reduzido sendo considerada a similaridade entre seqüências e não somente a identidade.

Os vetores gerados pelo esquema de codificação *E-SCSW*, sendo de mesma dimensão, podem ser utilizados como entrada em RNAs a fim de classificar seqüências de aminoácidos.

3.3 *E-SCSW* × *SCSW*

A fim de comparar o esquema de codificação *E-SCSW* com o esquema de codificação *SCSW* foram selecionadas proteínas de 12 bactérias, sendo a comparação realizada através da classificação dos vetores resultantes de cada esquema de codificação por *Redes Neurais Artificiais* (RNAs). A classificação foi baseada nas classes funcionais do *COG*, as quais são mostradas na Tabela 3.3.

Tabela 3.3: As 18 classes funcionais do COG sobre as quais foi realizada a classificação pelas *Redes Neurais Artificiais*

Classes funcionais do COG
J - Translation, ribosomal structure and biogenesis
K - Transcription
L - DNA replication, recombination and repair
D - Cell division and chromosome partitioning
O - Posttranslational modification, protein turnover
M - Cell envelope biogenesis, outer membrane
N - Cell motility and secretion
P - Inorganic ion transport and metabolism
T - Signal transduction mechanisms
C - Energy production and conversion
G - Carbohydrate transport and metabolism
E - Amino acid transport and metabolism
F - Nucleotide transport and metabolism
H - Coenzyme metabolism
I - Lipid metabolism
Q - Secondary metabolites biosynthesis, transport
R - General function prediction only
S - Function unknown

3.3.1 Seleção dos dados de entrada e treinamento das RNAs

As sequências de aminoácidos correspondentes às proteínas das bactérias *Burkholderia thailandensis* (Kim et al., 2005), *Carboxydothermus hydrogenoformans* (Wu et al., 2005), *Colwellia psychrerythraea* (Methé et al., 2005), *Hahella chejuensis* (Jeong et al., 2005), *Magnetospirillum magneticum* (Matsunaga et al., 2005),

Pseudomonas syringae (Joardar et al., 2005), *Salinibacter ruber* (Mongodin et al., 2005), *Shigella dysenteriae* (Yang et al., 2005), *Streptococcus agalactiae* (Tettelin et al.,) and *Xanthomonas campestris* (Qian et al., 2005) foram selecionadas para treinar as RNAs. Enquanto as proteínas das bactérias *Chromobacterium violaceum* (Vasconcelos et al., 2003) e *Chlamydophila felis* (Azuma et al., 2006) foram usadas para testar as RNAs.

Em todas as proteínas, o alfabeto de 20 caracteres foi substituído pelo alfabeto de 6 caracteres (*Exchange group*) a fim de solucionar os problemas de alta dimensionalidade dos vetores resultantes dos esquemas de codificação e de similaridade entre os aminoácidos.

O próximo passo foi verificar qual é o tamanho da janela deslizante ideal a ser aplicada às proteínas das 12 bactérias. As proteínas utilizadas para treinamento e teste das RNAs somam 31.525. Para todas as seqüências cada um dos três casos descritos em (Pevzner, 1995) e exemplificados na Seção 2.3 foram verificados a fim de se determinar quais seqüências eram ambíguas para um dado tamanho de janela deslizante. A verificação foi realizada para janelas deslizantes de tamanhos $n = 2, n = 3, n = 4, n = 5$ e $n = 6$. A Tabela 3.4 mostra o número de seqüências ambíguas para cada tamanho de janela deslizante.

Tabela 3.4: Número de sequências ambíguas obtido através da verificação de cada um dos três casos descritos na Seção 2.3. A verificação foi realizada em todas as sequências selecionadas para janelas deslizantes de tamanhos $n = 2, n = 3, n = 4, n = 5$ e $n = 6$

Tamanho da janela deslizante	Quantidade de seqüências ambíguas
2	20.462
3	9.289
4	3.356
5	70
6	20

Pode-se perceber que uma boa escolha para o tamanho da janela deslizante é $n = 6$, pois somente 20 sequências são ambíguas podendo ser eliminadas do conjunto de treinamento e teste. Entretanto, a fim de diminuir o custo computacional para a treinamento das RNAs foi utilizada janela deslizante de tamanho $n = 5$, sendo que as 70 sequências que apresentaram ambigüidade foram desconsideradas do conjunto de treinamento. Neste caso os vetores resultantes dos esquemas de codificação possuem dimensão $6^5 = 7.776$ para janela deslizante de tamanho

$n = 5$ ao invés de dimensão $6^6 = 46.656$ para janela deslizante de tamanho $n = 6$.

A classificação realizada pelas RNAs foi feita baseando-se nas 18 classes funcionais do *Clusters of orthologous groups* (COG)[7]. As classes *R - General function prediction only* e *S - Function unknown* foram consideradas como *não-classificadas*, por este motivo não foram utilizadas no treinamento das RNAs, sua utilização foi feita somente na fase de teste. A Tabela 3.5 mostra as 16 classes funcionais do COG que foram utilizadas na fase de treinamento das RNAs, assim como a quantidade de seqüências pertencentes à cada uma das classes, totalizando 26.089 seqüências.

Tabela 3.5: As 16 classes funcionais do COG utilizadas no treinamento das RNAs e as correspondentes quantidades de seqüências de aminoácidos selecionadas.

Classes funcionais do COG	Quantidade de Seqüências
J - Translation, ribosomal structure and biogenesis	1371
K - Transcription	2335
L - DNA replication, recombination and repair	2604
D - Cell division and chromosome partitioning	292
O - Posttranslational modification, protein turnover	1247
M - Cell envelope biogenesis, outer membrane	1984
N - Cell motility and secretion	999
P - Inorganic ion transport and metabolism	1671
T - Signal transduction mechanisms	2254
C - Energy production and conversion	1968
G - Carbohydrate transport and metabolism	1363
E - Amino acid transport and metabolism	2735
F - Nucleotide transport and metabolism	647
H - Coenzyme metabolism	1175
I - Lipid metabolism	1213
Q - Secondary metabolites biosynthesis, transport	884

A fim de melhorar a representatividade das classes *D*, *F* e *Q*, proteínas das bactérias *Geobacter metallireducens* (Childers et al., 2002), *Burkholderia pseudomallei* (Holdena et al., 2004), *Anabaena variabilis*[8], *Ralstonia eutropha*[9] e *Pseudomonas fluorescens* (Paulsen et al., 2005) referentes às três classes foram selecionadas. A Tabela 3.6 mostra as novas quantidades de proteínas das classes *D*, *F* e *Q* após a seleção das novas seqüências.

[7] http://www.ncbi.nlm.nih.gov/COG/old/palox.cgi?fun=all
[8] http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=genomeprj&cmd=Retrieve&dopt=Overview
[9] http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=genomeprj&cmd=Retrieve&dopt=Overview

Tabela 3.6: Quantidade de seqüências de aminoácidos após a nova seleção com o objetivo de melhorar a representatividade das classes *D*, *F* e *Q*.

Classes funcionais do COG	Quantidade de Seqüências
D - Cell division and chromosome partitioning	506
F - Nucleotide transport and metabolism	1075
Q - Secondary metabolites biosynthesis, transport	1711

Os esquemas de codificação *SCSW* e *E-SCSW* foram aplicados à todas as seqüências de aminoácidos. Foram utilizadas janelas deslizantes de tamanhos $n = 5$ e $n = 4$. No esquema de codificação *SCSW*, para cada seqüência, os vetores gerados pelas janelas deslizantes $n = 5$ e $n = 4$ foram concatenados resultando em um vetor de dimensão 9072 (Wu et al., 1992). Para evitar que a dimensão dos vetores gerados pelo esquema de codificação *SCSW* cresça, não foram utilizadas janelas deslizantes de tamanho $n = 3$, $n = 2$ e $n = 1$.

No esquema de codificação *E-SCSW* o peso para cada janela deslizante foi calculado utilizando score 1 para identidade, como mostrado na Seção 3.2.

Uma RNA foi criada para mapear cada uma das 16 classes funcionais do COG, onde a metodologia *um-contra-todos* (Hsu and Lin, 2002) foi utilizada de modo que a saída de cada RNA mapeia as seqüências de aminoácidos de uma classe contra as sequências de aminoácidos de todas as outras classes.

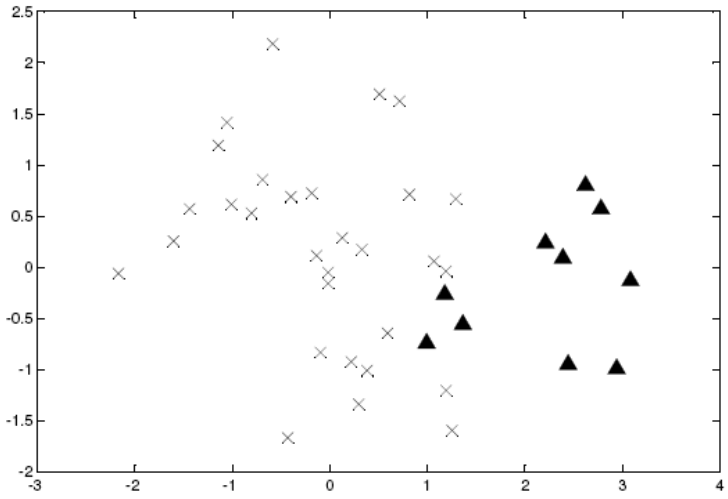
Pode-se perceber que as classes estão desbalanceadas e com a metodologia *um-contra-todos* o desbalanceamento fica ainda mais evidente. O treinamento de RNAs com classes desbalanceadas pode torná-las tendenciosas comprometendo a generalização e, conseqüentemente, o resultado dos testes quando estes forem realizados. Para minimizar o problema pode-se selecionar somente os pontos da margem de separação entre as classes mapeadas pela RNA.

O algoritmo *Condensed Nearest Neighbor* (CNN) (Hart, 1968) foi utilizado para realizar esta seleção. Para uma RNA que mapeia uma dada classe *A* contendo *x* seqüências, o *CNN* seleciona os pontos da seguinte forma:

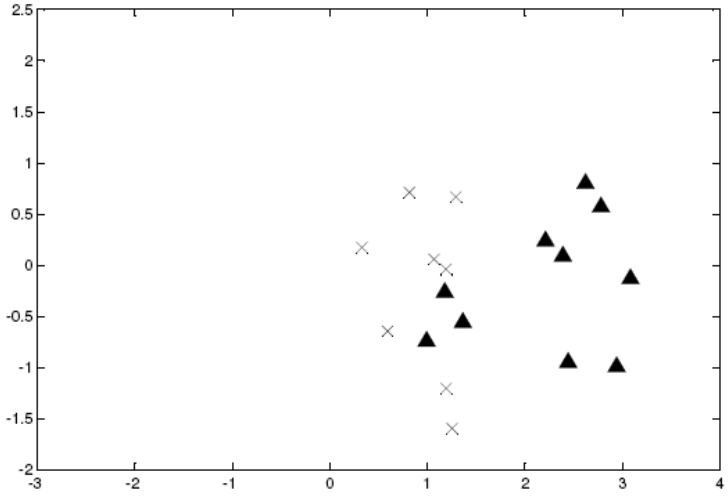
- é calculada a distância entre todos os vetores resultantes da aplicação do esquema de codificação a cada uma das *x* sequências de *A* contra todos os vetores correspondentes às seqüências das outras classes. Neste trabalho foi utilizada a distância Euclidiana;
- os pares de vetores são colocados em ordem crescente, de acordo com a distância entre eles;

- os x primeiros pares são selecionados para fazer parte do conjunto de treinamento e validação;

Como exemplo da aplicação do *CNN*, considere as duas classes da Figura 3.11(a) onde a *Classe 1* possui 30 elementos que são representados por $\times$ e a *Classe 2* possui 10 elementos que são representados por ▲. Aplicando-se o *CNN* para selecionar os pontos da margem de separação entre estas duas classes a distância entre $(\times, ▲)$ é calculada, $\forall \times \in$ *Classe 1* e $\forall ▲ \in$ *Classe 2*. Os pares de pontos são ordenados em ordem crescente de acordo com a distância entre eles. Os 10 primeiros pares de pontos são tomados como pertencentes à margem de separação. O valor 10 corresponde à quantidade de pontos da menor classe, no caso a *Classe 2*. A Figura 3.11(b) mostra o conjunto de pontos na margem de separação após a aplicação do *CNN*. Pode-se perceber que a *Classe 2* possui 10 pontos enquanto a *Classe 1* possui apenas 9, isto ocorre pelo fato de que um mesmo ponto na *Classe 1* é o mais próximo de dois pontos da *Classe 2*, ou seja, no cálculo da distância o mesmo ponto é tomado em dois pares diferentes. Isto faz com que o desbalanceamento entre as classes não seja totalmente resolvido e sim minimizado.



(a)



(b)

Figura 3.11: Exemplo da aplicação do *CNN*. Em (a) são mostradas duas classes contendo 30 e 10 elementos, respectivamente, ilustrando o desbalanceamento. Em (b) são mostrados os elementos de cada classe obtidos pela aplicação do *CNN*.

Após a aplicação do *CNN*, 82% dos vetores de cada classe foram tomados para treinamento e 18% para teste das RNAs.

Assim como o algoritmo utilizado no treinamento das RNAs, o número de iterações no treinamento e o número de neurônios na camada escondida foram os mesmo para todas as RNAs a fim de comparar com mais acurácia os dois esquemas de codificação.

O algoritmo de treinamento utilizado foi a *Regularização Bayesiana* (Mackay, 1992) em RNAs com 8 neurônios na camada escondida e 1 neurônio na camada de saída, embora em (Wu, 1997) tenha sido utilizado o algoritmo *backpropagation*.

O número de neurônios na camada escondida foi escolhido com base em testes realizados com sequências escolhidas aleatoriamente de duas classes funcionais do COG, classes *G* e *J*. Foram realizados testes com RNAs de 6, 7, 8 e 9 neurônios na camada escondida. O resultado das RNAs com 8 e 9 neurônios na camada escondida foram semelhantes e, adicionalmente, superiores aos resultados das RNAs com 6 e 7 neurônios na camada escondida. Todas as RNAs foram treinadas com 800 iterações.

A *Regularização Bayesiana* foi escolhida por ser capaz de tratar o problema de polarização e variância em RNAs. O algoritmo tenta minimizar um função composta pelo *erro quadrático médio*, pode levar a uma RNA super-ajustada (alta variância e baixa polarização), e pela *norma dos pesos*, que pode levar a uma RNA sub-ajustada (baixa variância e alta polarização). Deste modo a *Regularização Bayesiana* pode encontrar uma RNA com um bom ajuste em relação aos dados de treinamento (Mackay, 1992).

Todos os vetores de entrada das RNA's (treinamento e teste) foram normalizados com valores entre 0 e 1. Na fase de treinamento cada vetor de entrada possuía uma saída correspondente com valor 1, indicando a pertinência à classe em questão ou valor 0 indicando a não pertinência. Na fase de teste foi utilizado um valor de limiar para se determinar a pertinência ou não de uma seqüência a uma dada classe. Para um vetor de entrada, se a saída for maior que $0,75$ indica que a seqüência correspondente pertence à classe mapeada pela RNA, caso contrário a seqüência correspondente não pertence à classe em questão.

Após o treinamento das RNAs, foram construídos dois classificadores, um baseado no esquema de codificação *SCSW* e outro baseado no esquema de codificação *E-SCSW* a fim de comparar os dois esquemas de codificação. A Figura 3.12 mostra um esquema geral para cada classificador de sequências de aminoácidos construído. O classificador é composto por três partes:

- Um módulo para a codificação das sequências a serem classificadas;
- Um módulo composto pelas RNAs previamente treinadas, onde cada uma mapeia uma classe funcional do *COG* contra todas as outras;
- Um módulo para verificar a qual classe a seqüência original pertence;

O módulo de *codificação de seqüências* tem por objetivo codificar a seqüência de aminoácidos a ser classificada (*SCSW* ou *E-SCSW*). O módulo composto pelas *RNAs* tem por objetivo classificar os vetores gerados pelo módulo anterior. Cada *RNA* resulta em uma resposta de pertinência ou não à classe funcional do *COG* correspondente. O último módulo, tem como objetivo agrupar as respostas de todas as 16 *RNAs* resultando em na classe ou nas classes em que a seqüência original pertence. Caso mais de uma RNA classifique uma mesma seqüência esta é considerada pertencente às classes em questão. Caso nenhuma *RNA* classifique uma seqüência dada como entrada esta é considerada *não classificada* pelo **COG**.

3.3.2 Teste das RNAs treinadas com os vetores gerados pelos esquemas SCSW × E-SCSW

O próximo passo foi testar as RNAs com o conjunto de sequências de aminoácidos representando as proteínas das bactérias *Chromobacterium violaceum* (Vasconcelos et al., 2003) e *Chlamydophila felis* (Azuma et al., 2006). A *Chromobacterium violaceum* foi escolhida pelo fato de seu genoma ter sido inteiramente executado no Brasil pelo *Brazilian National Genome Sequencing Consortium* (Vasconcelos et al., 2003). Já a *Chlamydophila felis* foi escolhida pelo fato de seu genoma ter sido determinado e seu conjunto de proteínas depositado nos bancos de dados públicos em 2006 tendo como objetivo verificar como as RNAs irão se comportar classificando dados atualizados, já que foram treinadas com dados depositados nos bancos de dados públicos em 2005.

O número de proteínas em cada classe funcional do COG referente às bactérias *Chromobacterium violaceum* e *Chlamydophila felis* é mostrado na Tabela 3.7. As classes *R*, *S* e as proteínas não pertencentes a nenhuma classe do COG foram agrupadas em uma única classe indicando proteínas não classificadas. Para estas proteínas foi criada a classe *Not in COG*.

Os vetores gerados pelo esquema *SCSW*, a partir das proteínas das duas bactérias, foram aplicadas às respectivas RNAs previamente treinadas com os vetores gerados pelo esquema *SCSW*. Da mesma forma, os vetores gerados pelo esquema

Tabela 3.7: Quantidade de sequências de aminoácidos de cada classe funcional do COG utilizada para teste das RNAs previamente treinadas. A segunda coluna mostra a quantidade de sequências da *Chromobacterium violaceum* e a terceira coluna da *Chlamydophila felis*

COG	Chromobacterium violaceum	Chlamydophila felis
J	168	90
K	270	28
L	143	60
D	41	11
O	134	33
M	222	40
N	255	15
P	159	29
T	304	20
C	204	41
G	205	26
E	334	58
F	79	21
H	152	36
I	118	29
Q	130	8
Not in COG	1716	494

E-SCSW foram aplicados às respectivas RNAs treinadas com os vetores gerados pelo esquema *E-SCSW*.

Os testes para os dois esquemas de codificação foram conduzidos da seguinte forma:

- considerando a seqüência de aminoácidos correspondentes à uma proteína P pertencente à classe funcional Cl;
- P foi codificada gerando o vetor P_{cod};
- P_{cod} foi aplicado a cada uma das 16 RNAs;
- se somente a RNA que mapeia a classe Cl classificar P_{cod} como pertencente à classe Cl, então a classificação é tida como correta;
- se uma RNA que mapeia a classe $Cl' \neq CL$ classificar P_{cod} como pertencente à classe Cl', então P é tida como pertencente à classe Cl' e o resultado é dado como incorreto;
- se nunhuma RNA classificar P_{cod}, P é tida como sem classificação;

O treinamento das Redes Neurais Artificiais e os testes realizados com as seqüências de aminoácidos da *Chromobacterium violaceum* e *Chlamydophila felis* foram realizados utilizando o software *Matlab 6.0*[10] executando sobre o sistema operacional *Suse 9.0* [11] em um *intel pentium 4*[12] com $1GB$ de memória RAM e $2GB$ de partição swap. O treinamento de cada RNA levou em torno de 30 horas.

Para comprovar o resultado da classificação realizada pelas RNAs, cada seqüência de aminoácidos correspondente a um vetor classificado incorretamente pelas RNAs foram analisadas individualmente. Para este fim, a ferramenta *Conserved Domain Search* (CD-Search) para a detecção de domínios funcionais e estruturais em proteínas foi utilizada (Marchler-Bauer and Bryant, 2004). O *CD-Search* é baseado na heurística de alinhamento par-a-par *BLAST* que pode ser utilizado para a busca em várias bases de dados como *SMART*, *PFAM*, *COG*, *KOG* e *CDD*[13]. Como estamos interessados em verificar a classificação funcional em relação ao *COG* todas as buscas foram realizadas no banco de dados do *COG*. O valor de cada parâmetro utilizado foi o sugerido pela ferramenta. O parâmetro *Maximal hits* limita o tamanho da lista de *hits* produzida pelo *CDD*, sendo o valor padrão sugerido 100[14]. O parâmetro *Expect Value* indica o número de alinhamentos aleatórios esperados. Este valor depende do tamanho da seqüência, da matriz de similaridade e da penalidade dos *gaps*. Quanto menor o valor de *Expect Value* menos provável a similaridade encontrada ser aleatória (Kork et al., 2003). O valor utilizado foi 0.01 onde falsos positivos são raros de ocorrer[15]. O último parâmetro, chamado *Low complexity filter*, permite que seja aplicado um filtro de modo que somente os alinhamentos mais relevantes sejam mostrados[16]. O resultado do *CD-Search* com estes parâmetros foram considerados corretos neste trabalho.

A comparação estatística dos dois grupos de Redes Neurais Artificiais, treinadas com vetores gerados pelo esquema *SCSW* e *E-SCSW*, para a *Chromobacterium violaceum* e para a *Chlamydophila felis* foi realizada utilizando o *teste-t* (Ewens and Grant, 2001) com nível de significância de $p < 0,05$. A análise foi realizada através do software *GraphPad Prism* versão 4.0[17].

Um análise estatística dos dois grupos de RNAs

No Capítulo 4 será mostrado o resultado do teste realizado com o esquema de

[10] http://www.mathworks.com/
[11] http://www.opensuse.org/
[12] www.intel.com
[13] http://www.ncbi.nlm.nih.gov/Structure/cdd/wrpsb.cgi
[14] http://www.ncbi.nlm.nih.gov/Structure/cdd/cdd_help.shtml
[15] http://www.ncbi.nlm.nih.gov/Structure/cdd/cdd_help.shtml
[16] http://www.ncbi.nlm.nih.gov/Structure/cdd/cdd_help.shtml
[17] http://www.graphpad.com/prism/Prism.htm

codificação *SCSW* para agrupar seqüências de aminoácidos através do algoritmo k-means, onde o resultado do agrupamento foi comparado com o alinhamento múltiplo das mesmas seqüências realizado pelo *ClustalW*.

No Capítulo 4 será mostrado também o resultado dos testes realizados com as RNAs treinadas com os vetores gerados pelos esquemas de codificação *SCSW* e *E-SCSW*, assim como o resultado das análises realizadas, utilizando o *CD-Search*, com todas as seqüências de aminoácidos cujo resultado das RNAs foram incongruentes com a classificação nos bancos de dados públicos.

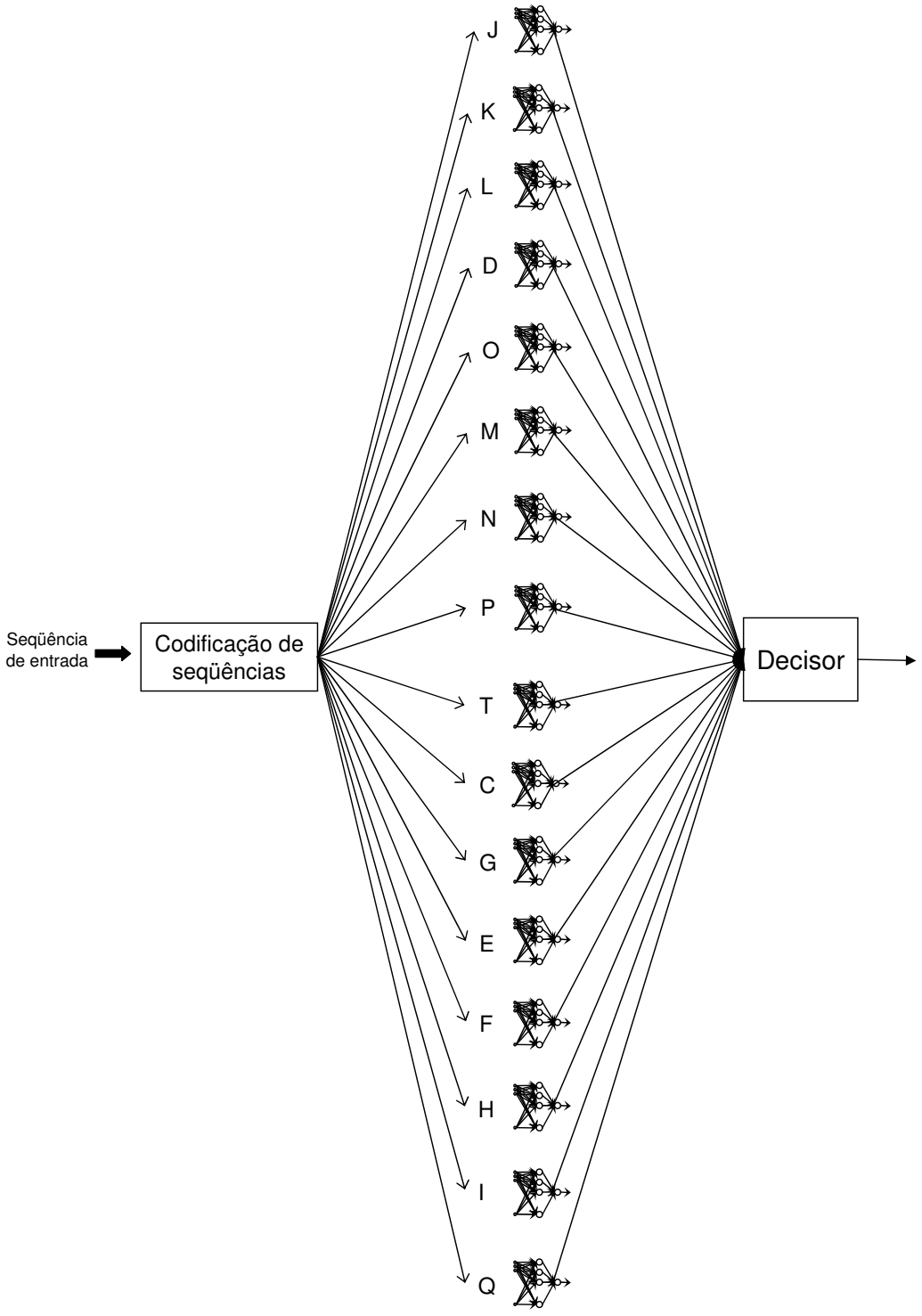


Figura 3.12: Modelo esquemático do classificador de sequências de aminoácidos construído.

CAPÍTULO 4

Resultados

Neste capítulo é mostrado um teste realizado com o esquema *SCSW* a fim de verificar sua aplicabilidade na busca de similaridade entre proteínas. Posteriormente é mostrado uma comparação entre os esquemas *SCSW* e *E-SCSW* utilizando, como ferramenta de comparação, RNAs. A Seção 4.1 mostra um teste realizado com o esquema de codificação *SCSW* onde foi aplicado o método *Principal Component Analysis* para redução de dimensão aos vetores resultantes do esquema de codificação. O método de agrupamento *k-means* foi aplicado aos vetores resultantes da aplicação do *Principal Component Analysis* onde o resultado foi comparado com o alinhamento múltiplo das seqüências utilizadas realizado pelo *ClustalW*. A Seção 4.2 mostra a comparação realizada entre os esquemas *SCSW* e *E-SCSW*. A comparação foi realizada através da classificação funcional de proteínas por RNAs. O conjunto de proteínas de 10 bactérias foi utilizado no treinamento das RNAs, sendo descartadas as proteínas ambíguas. Após treinadas as RNAs foram testadas com o conjunto de proteínas de 2 outras bactérias. Adicionalmente, as seqüências ambíguas que foram descartadas no treinamento foram utilizadas para testar as RNAs.

4.1 Teste do esquema de codificação SCSW

Após a aplicação da metodologia apresentada na Seção 3.1 para verificar a eficiência do esquema de codificação *SCSW*, dos 40 grupos obtidos pela aplicação do *K-means*, 15 grupos, com um total de 72 sequências de aminoácidos, foram com-

patíveis com os domínios do *PFAM*. A Tabela 4.1 mostra os 15 grupos encontrados pelo *K-meas* compatíveis com o *PFAM*. A primeira coluna mostra os domínios do *PFAM* correspondentes a cada um dos 15 grupos encontrados. A segunda coluna mostra a quantidade de sequências de aminoácidos em cada grupo, totalizando 72 sequências. A penúltima linha, correspondente à *No Match*, indica as sequências que foram agrupadas no mesmo grupo e que não possuem nenhum alinhamento com os domínios do *PFAM*. Os outros grupos não foram mostrados na Tabela 4.1 pois foram caracterizados pelo *PFAM* como *prováveis domínios*.

Tabela 4.1: Agrupamentos obtidos pela aplicação do *K-means* às 112 sequências selecionadas compatíveis com os domínios do *PFAM*. A primeira coluna mostra os domínios do *PFAM* correspondentes a cada um dos 15 grupos encontrados. A segunda coluna mostra a quantidade de sequências de aminoácidos em cada grupo.

Domínios	Número de sequências de aminoácidos
RRM	2
FERM	3
SCP	3
EF Hand	8
SH3	5
Four TRANSMEMBRANE	9
Fibronectin Type III	9
Extensin	1
Annexin	2
Myosin	1
ShTk	3
Calreticulin	1
TIM	2
Teaniidae	18
No Match	5
Total	72

Os agrupamentos obtidos pela *K-means* foram, em grande parte, confirmados pelo *ClustaW*[1] (Thompson et al., 1994), ferramenta para alinhamento múltiplo de sequências. A Figura 4.1 mostra parte dos agrupamentos obtidos pelo *ClustalW*, onde cada seqüência é representada pelo seu número de identificação (*GI*)[2], (Rodrigues et al., 2003b), (Rodrigues et al., 2004) e sobre cada agrupamento está o nome do domínio existente em cada seqüência no agrupamento.

[1] http://www.ebi.ac.uk/clustalw/
[2] http://www/cnbi.nlm.nih.gov/Sitemap/sequenceIDs.html

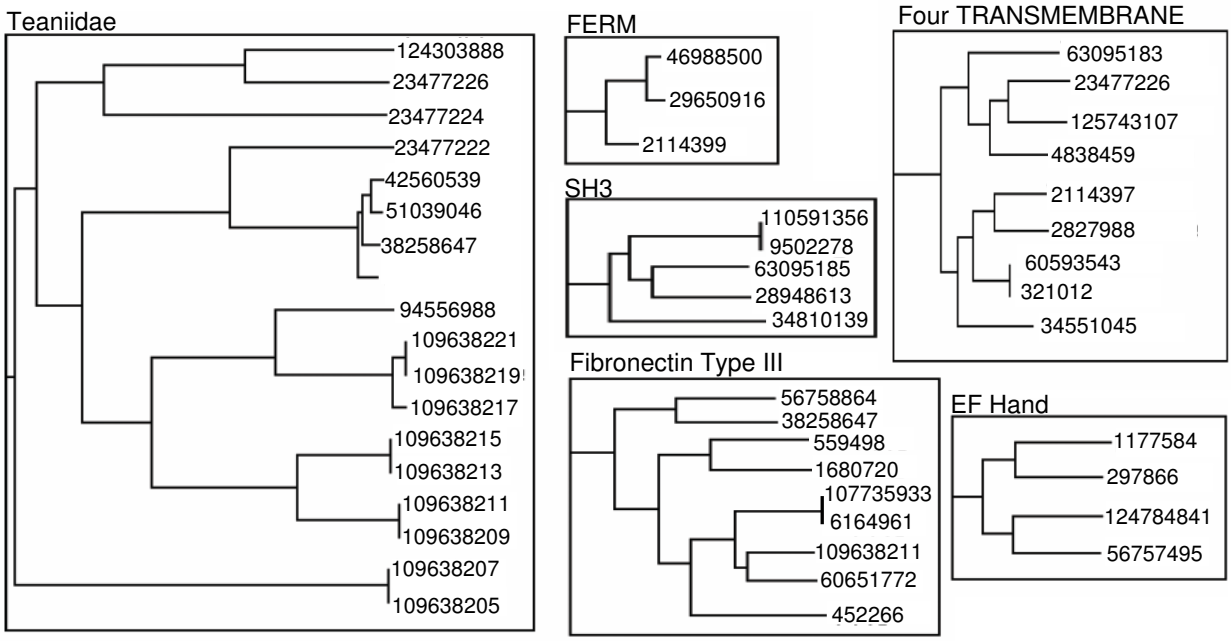


Figura 4.1: Alguns agrupamentos obtidos pelo alinhamento múltiplo das 112 seqüências selecionadas através do *ClustalW* que são compatíveis com os agrupamentos obtidos pela *K-means*. Cada seqüência é identificada pelo seu *GI* e sobre cada agrupamento está o nome do domínio existente em cada seqüência no agrupamento correspondente.

Das 112 sequências de aminoácidos agrupadas pelo *K-means*, um total de 67 sequências tiveram os domínios confirmados pelo *PFAM*, 5 sequências foram agrupadas mas não possuem nenhum domínio, de acordo com o *PFAM*, 3 sequências que não possuem nenhum domínio foram agrupadas juntamente com o grupo *Taeniidae* e 37 foram agrupadas em grupos distintos, onde todas elas possuem *prováveis domínios* de acordo com o *PFAM*.

A fim de testar a codificação *SCSW* para janelas deslizantes de tamanhos maiores que $n = 2$, foi utilizado o mesmo conjunto de 112 sequências de aminoácidos. Foi aplicada a mesma metodologia apresentada em (Rodrigues et al., 2004) e (Rodrigues et al., 2003b) para janelas deslizantes de tamanho variando de $n = 3$ a $n = 10$, resultando em vetores de dimensão variando de 20^3 a 20^{10}. Analisando os resultados verificamos que os agrupamentos se mantinham para janelas deslizantes de tamanho variando de 2 à 6. Entretanto, para as janelas deslizantes de tamanho variando de 7 à 10 a acurácia dos agrupamentos encontrados começou a diminuir (Rodrigues et al., 2003a).

Com este resultado podemos observar que, com o aumento do tamanho da janela deslizante, a similaridade entre subseqüências menores que n é ignorada,

conseqüentemente, pequenas regiões de similaridade não são avaliadas, problema já levantado na Seção 2.3. O esquema de codificação *Extended-Sequence Coding by Sliding Window* (E-SCSW), descrito na Seção 3.2, é capaz de minimizar este problema, assim como o problema de ambigüidade quando utilizada uma janela deslizante de tamanho apropriado (Seção 2.3).

A seção 4.1 mostra o resultado da comparação realizada entre os dois esquemas de codificação para verificar a superioridade do esquema proposto, sendo utilizadas Redes Neurais Artificiais como ferramenta de comparação.

4.2 Comparação entre os esquemas de codificação E-SCSW $\times$ SCSW

Como especificado na Seção 3.3, os dois esquemas de codificação foram comparados através da classificação de seus vetores resultantes por Redes Neurais Artificiais de acordo com as classes funcionais do *COG*.

Após a aplicação do CNN ao conjunto de dados, 82% dos vetores de cada classe foram tomados para treinamento e 18% para teste das RNAs.

Para os vetores gerados pelo esquema de codificação *SCSW*, a taxa de acerto do conjunto de teste variou de 79% à 87% entre as 16 RNAs. Enquanto que, para os vetores gerados a partir do esquema de codificação *E-SCSW*, a taxa de acerto variou de 89% à 95%.

A Figura 4.2 mostra a taxa de acerto no teste para cada uma das 16 RNAs (mapeando cada classe do COG), referentes aos vetores gerados pela esquema de codificação *SCSW* e *E-SCSW*. As barras em branco mostram a taxa de acerta para cada RNA treinada com os vetores gerados pelo esquema *SCSW*, as barras em cinza mostram a taxa de acerta para cada RNA treinada com os vetores gerados pelo esquema *E-SCSW*.

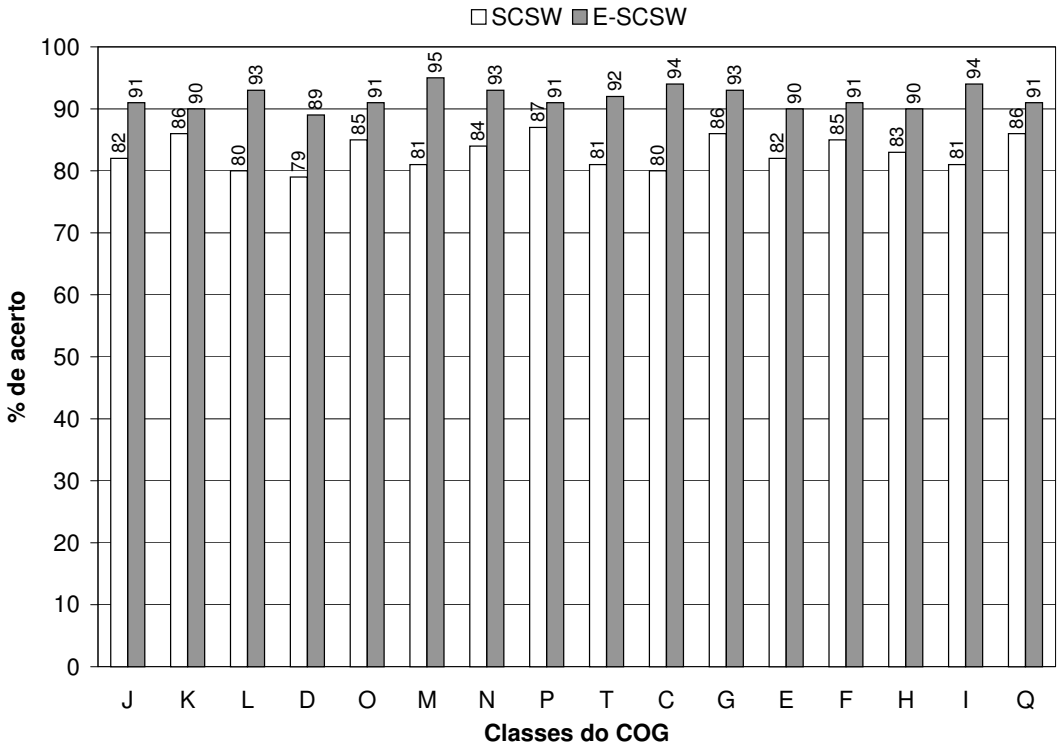


Figura 4.2: Taxa de acerto para cada RNA correspondente a uma classe funcional do COG treinada com os vetores resultantes do esquema *SCSW* (barras em branco) e *E-SCSW* (barras em cinza). Os dados utilizados para teste correspondem aos 18% dos vetores que foram selecionadas após a aplicação do *CNN*.

A próxima subseção mostra os testes realizados com as sequências de aminoácidos da *Chromobacterium violaceum*.

4.2.1 Teste das RNAs com as sequências de aminoácidos da Chromobacterium violaceum

Primeiramente os testes foram realizados com os vetores gerados a partir das sequências de aminoácidos da *Chromobacterium violaceum*.

A taxa de acerto para cada RNA foi calculada com base na classificação das proteínas depositadas nos bancos de dados públicos. A Figura 4.3 mostra a comparação das taxas de acerto para cada RNA (correspondente a uma classe funcional do COG) referentes à bactéria *Chromobacterium violaceum*. As barras em branco mostram as taxas de acerto para as RNAs treinadas com os vetores gerados pelo esquema de codificação *SCSW*, as barras em cinza mostram as taxas de acerto para as RNAs treinadas com os vetores gerados pelo esquema de codificação *E-SCSW*. Sobre cada barra é mostrada a porcentagem de acerto para cada RNA. As barras especificadas como *Not in COG* indicam as sequências que não foram classificadas

por nenhuma das 16 RNAs.

A taxa de acerto das RNAs para os vetores gerados pelo esquema de codificação *SCSW* variou entre $60,1\%$ e $78,9\%$ enquanto que a taxa de acerto das RNAs para os vetores gerados pelo esquema de codificação *E-SCSW* variou de $73,1\%$ à $98,3\%$.

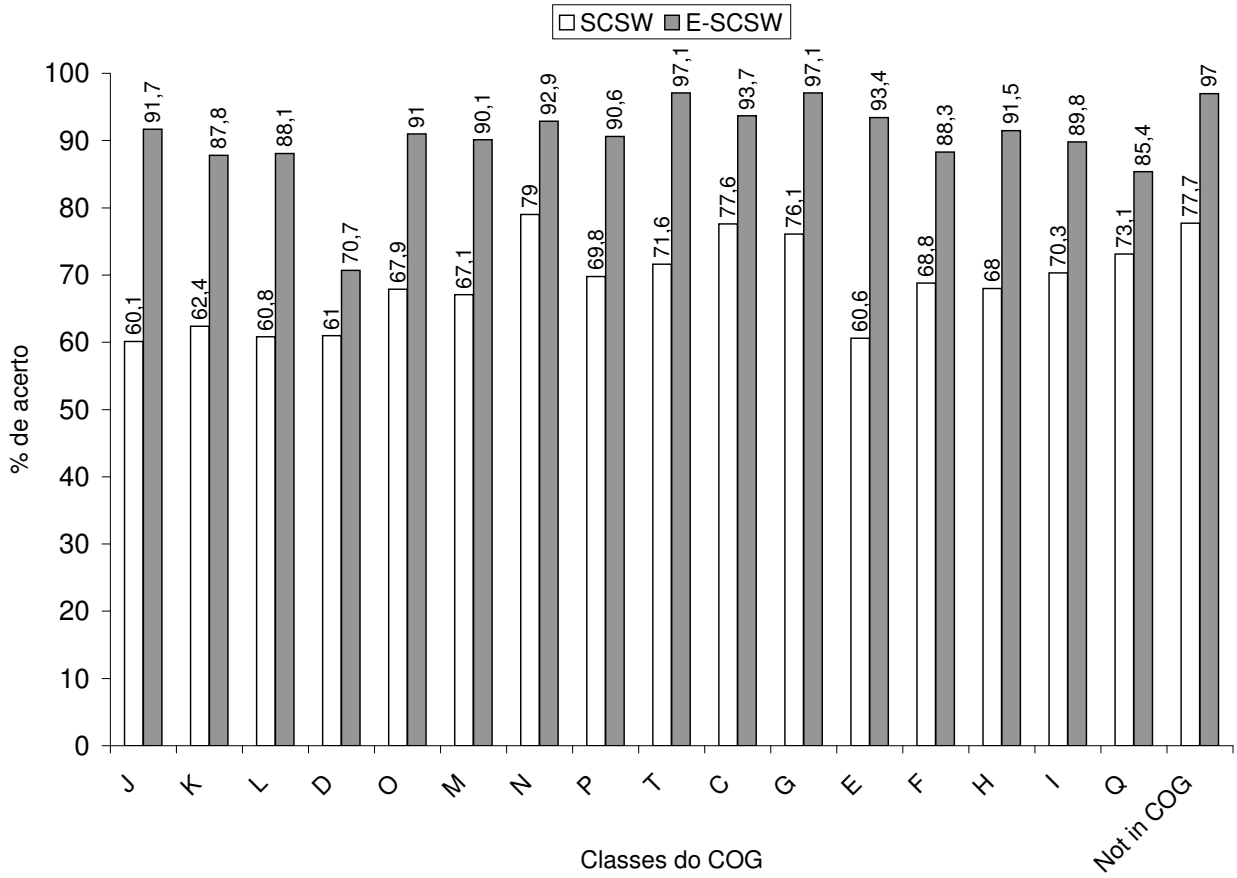


Figura 4.3: Resultado dos testes realizados com as sequências de aminoácidos da *Chromobacterium violaceum* aplicadas às RNAs que mapeiam cada classe funcional do COG treinadas com os vetores gerados pelos esquemas de codificação *SCSW* e *E-SCSW*. As barras em banco indicam a porcentagem de acerto das RNAs treinadas com os vetores gerados pelo esquema *SCSW*. As barras em cinza indicam a taxa de acerto das RNAs treinadas com os vetores gerados pelo esquema *E-SCSW*. Sobre cada barra é mostrada a porcentagem de acerto da RNA correspondente.

Como especificado na Seção 3.3.2, os vetores classificados de forma diferente pelas RNAs, levando em consideração a classificação nos bancos de dados públicos, tiveram as sequências de aminoácidos correspondentes analisadas individualmente. O *CD-Search* e a base de dados do *COG* foram utilizados como ferramenta de análise.

A Tabela 4.2 mostra os resultados das análises realizadas em cada seqüência de aminoácidos da *Chromobacterium violaceum* cujo vetor correspondente foi classificado de forma diferente pelas RNAs.

Para a Tabela 4.2:

- a primeira coluna indica as 16 classes funcionais do COG sendo que na última linha as classes *R*, *S* e *Not in COG* foram agrupadas em uma só classe indicando sequências de aminoácidos não classificadas;
- a segunda coluna mostra a quantidade de sequências de aminoácidos analisadas utilizando o *CD-Search*;
- a terceira coluna mostra a quantidade de sequências de aminoácidos que, depois da análise, se mostraram incoerentes nos bancos de dados públicos e que foram classificadas corretamente pelas RNAs;
- a quarta coluna mostra a quantidade de sequências de aminoácidos cuja classificação foi complementada pelas RNAs, ou seja, sequências de aminoácidos com domínios referentes a mais de uma classe funcional e classificadas em somente uma das classes nos bancos de dados públicos;
- a última coluna mostra quantas sequências de aminoácidos as RNAs realmente não conseguiram classificar.

Tabela 4.2: Análise das sequências de aminoácidos da *Chromobacterium violaceum* classificadas de maneira diferente em relação aos bancos de dados públicos pelas RNAs. A primeira coluna indica as 16 classes funcionais do COG sendo que na última linha as classes *R*, *S* e *Not in COG* foram agrupadas em uma só classe indicando sequências de aminoácidos não classificadas. A segunda coluna mostra a quantidade de sequências de aminoácidos analisadas utilizando o *CD-Search*. A terceira coluna mostra a quantidade de sequências de aminoácidos que, depois da análise, se mostraram diferentes com os bancos de dados públicos e que foram classificadas corretamente pelas RNAs; A quarta coluna mostra a quantidade de sequências de aminoácidos cuja classificação foi complementada pelas RNAs, ou seja, sequências de aminoácidos com domínios referentes a mais de uma classe funcional e classificadas em somente uma das classes nos bancos de dados públicos. A última coluna mostra quantas sequências de aminoácidos as RNAs realmente não conseguiram classificar.

Classes Funcionais do COG	Proteínas Analisadas		Classificação correta-RNAs		Complemento à classificação		Classificação incorreta-RNAs	
	SCSW	E-SCSW	SCSW	E-SCSW	SCSW	E-SCSW	SCSW	E-SCSW
J	67	14	0	5	2	1	65	8
K	102	33	0	22	0	0	102	11
L	56	27	0	9	1	4	55	4
D	16	12	0	8	0	0	16	4
O	43	12	0	5	0	0	43	7
M	73	22	0	14	3	1	70	7
N	53	18	0	13	1	1	52	4
P	48	15	0	6	0	0	48	9
T	87	9	0	1	4	2	83	6
C	46	13	0	8	0	0	46	5
G	49	6	0	2	0	0	49	4
E	132	22	0	12	1	2	131	8
F	24	9	0	4	0	1	24	4
H	49	13	0	6	0	0	49	7
I	35	12	0	4	0	0	35	8
Q	35	19	0	12	0	2	35	5
R, S and Not in COG	259	35	6	35	0	0	253	0
Total	1174	291	6	166	12	14	1156	101

Pode-se perceber que houve uma melhora nas taxas de acerto para algumas RNAs treinadas com vetores gerados pelo esquema de codificação *SCSW* após a análise individual das sequências que variou de $0,3\%$ à $1,4\%$. Adicionalmente, houve uma melhora na taxa de acerto para todas as RNAs treinadas com os vetores gerados pelo esquema de codificação *E-SCSW* que variou de $1,0\%$ à $19,5\%$. A Figura 4.4 mostra o percentual de melhora na taxa de acerto das RNAs treinadas com os dois esquemas de codificação. As barras em branco indicam o percentual de melhora na taxa de acerto para as RNAs treinadas com os vetores gerados pelo esquema *SCSW*, similarmente, as barras em cinza indicam o percentual de melhora na taxa de acerto para as RNAs treinadas com os vetores gerados pelo esquema *E-SCSW*. Sobre cada barra está o percentual de melhora na taxa de acerto da RNA correspondente.

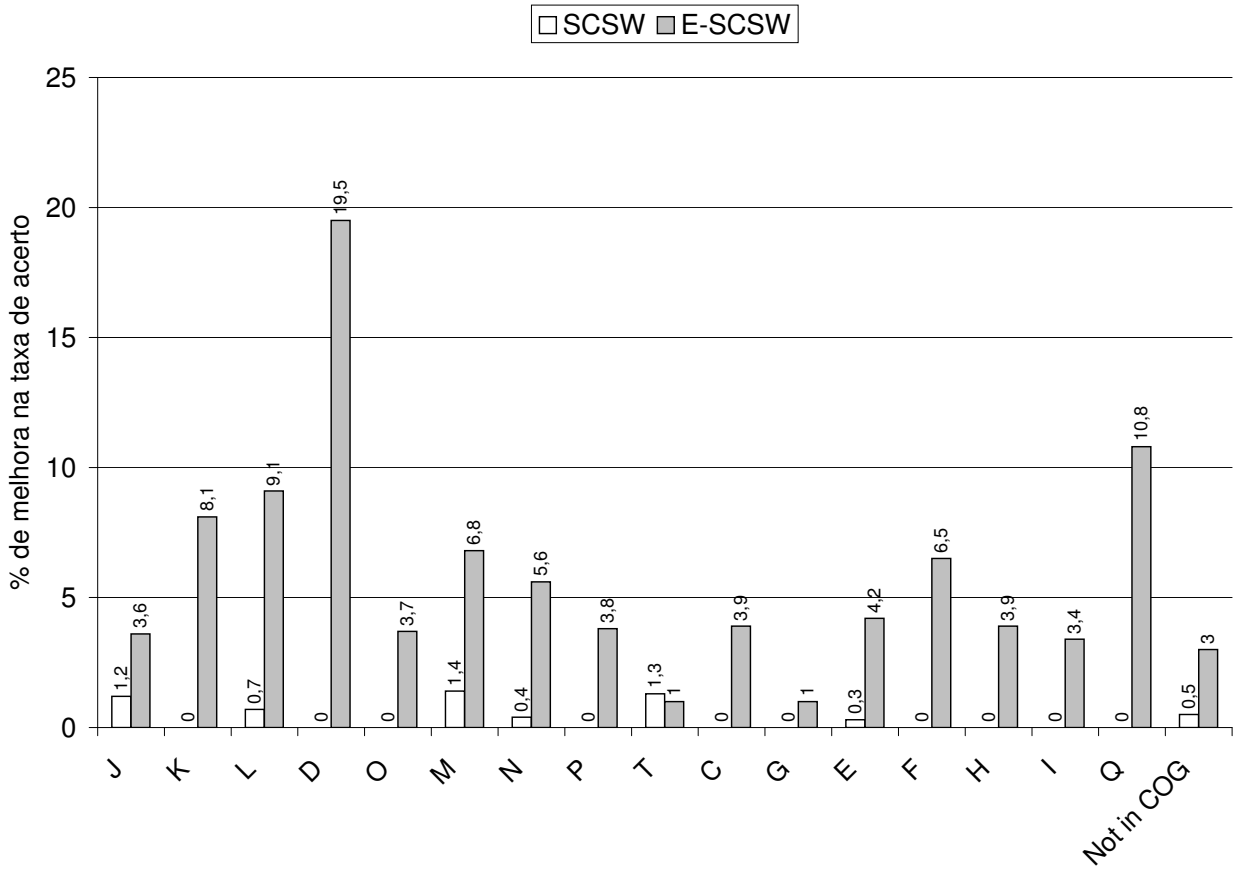


Figura 4.4: Porcentagem de aumento na taxa de acerto das RNAs após a análise, com o *CD-Search* contra o banco de dados do *COG*, das sequências de aminoácidos da *Chromobacterium violaceum* que foram classificadas de modo diferente pelas RNAs. As barras em branco indicam a melhora na taxa de acerto de cada RNA treinada com os vetores gerados pelo esquema de codificação *SCSW*. As barras em cinza indicam a melhora na taxa de acerto de cada RNA treinada com os vetores gerados pelo esquema de codificação *E-SCSW*. Sobre cada barra é mostrada a porcentagem de melhora após a análise das sequências.

Após as verificações realizadas com o *CD-Search* e atualizando a taxa de acerto de cada RNA, a Figura 4.5 mostra a comparação das taxas de acerto para cada RNA, correspondente às classes funcionais do *COG*, referentes às sequências de aminoácidos da *Chromobacterium violaceum*.

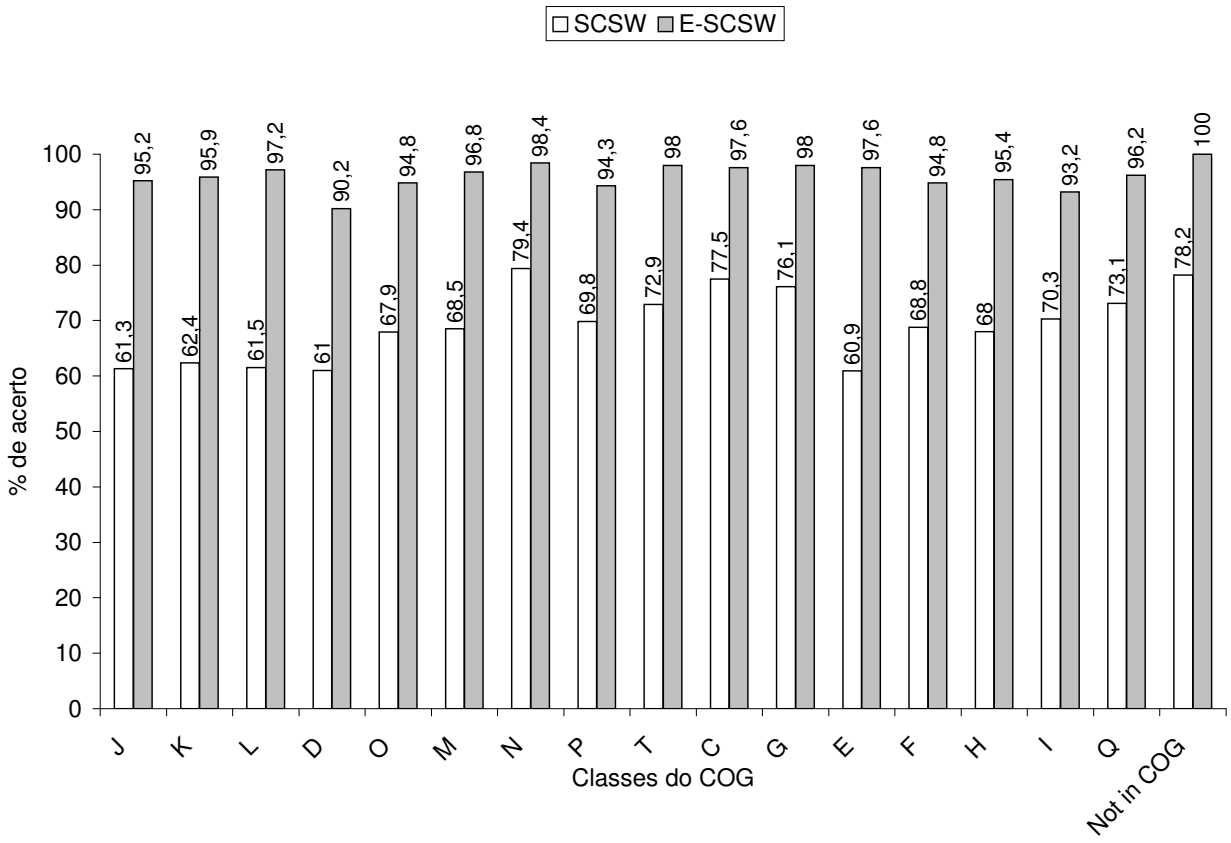


Figura 4.5: Resultado dos testes realizados com as sequências de aminoácidos da *Chromobacterium violaceum* aplicadas às RNAs que mapeiam cada classe funcional do COG treinadas com os vetores gerados pelos esquemas de codificação *SCSW* e *E-SCSW* após as análises realizadas com o *CD-Search*. As barras em banco indicam a porcentagem de acerto das RNAs treinadas com os vetores gerados pelo esquema *SCSW*. As barras em cinza indicam a taxa de acerto das RNAs treinadas com os vetores gerados pelo esquema *E-SCSW*. Sobre cada barra é mostrada a porcentagem de acerto da RNA correspondente.

A variação na taxa de acerto que era de $60,1\%$ à $78,9\%$ para as RNAs treinadas com o esquema *SCSW* passou a ser de $60,9\%$ à $79,4\%$, enquanto que a taxa de acerto das RNAs treinadas com os vetores gerados pelo esquema de codificação *E-SCSW* que variava de $73,1\%$ à $98,3\%$ passou a variar de $90,2\%$ à 100%.

A Figura 4.6 mostra a análise estatística das taxas de acerto das Redes Neurais Artificiais tendo como estrada as sequências de aminoácidos da *Chromobacterium violaceum*. Pode ser observada uma diferença significativa entre os dois grupos de RNAs, onde a diferença entre as médias é de $26,82 \pm 1,632$.

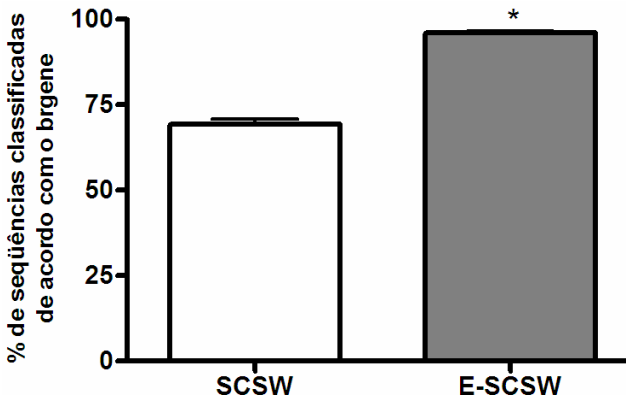


Figura 4.6: Análise estatística entre as taxas de acerto das Redes Neurais Artificiais tendo como estrada as sequências de aminoácidos da *Chromobacterium violaceum*. As barras representam a média $\pm$ erro-padrão com $n = 17$. A barra em branco corresponde ao resultado das RNAs treinadas com os vetores gerados pelo esquema *SCSW* e a barra em cinza corresponde ao resultado das RNAs treinadas com os vetores gerados pelo esquema *SCSW*; $^{*}p < 0,05\ vs\ SCSW$

4.2.2 Teste das RNAs com as sequências de aminoácidos da Chlamydophila felis

O próximo passo foi testar as RNAs com os vetores gerados pelos esquemas de codificação *SCSW* e *E-SCSW* a partir das sequências de aminoácidos da *Chlamydophila felis*. Todos os testes realizados foram similares aos testes com as sequências de aminoácidos da *Chromobacterium violaceum*.

A Figura 4.7 mostra a comparação das taxas de acerto para cada RNA (correspondente a uma classe funcional do COG) referentes à *Chlamydophila felis*. As barras em branco indicam a taxa de acerto das RNAs que foram treinadas com os vetores gerados pelo esquema *SCSW* e as barras em cinza a taxa de acerto das RNAs treinadas com o esquema de codificação *E-SCSW*. Sobre cada barra é mostrada a porcentagem de acerto para cada RNA. As barras especificadas como *Not in COG* indicam as sequências que não foram classificadas por nenhuma das 16 RNAs.

A variação na taxa de acerto foi de $61,9\%$ à $76,7\%$ para as RNAs treinadas com os vetores gerados pelo esquema de codificação *SCSW* e de $60,0\%$ à $93,3\%$ para as RNAs treinadas com o esquema de codificação *E-SCSW*.

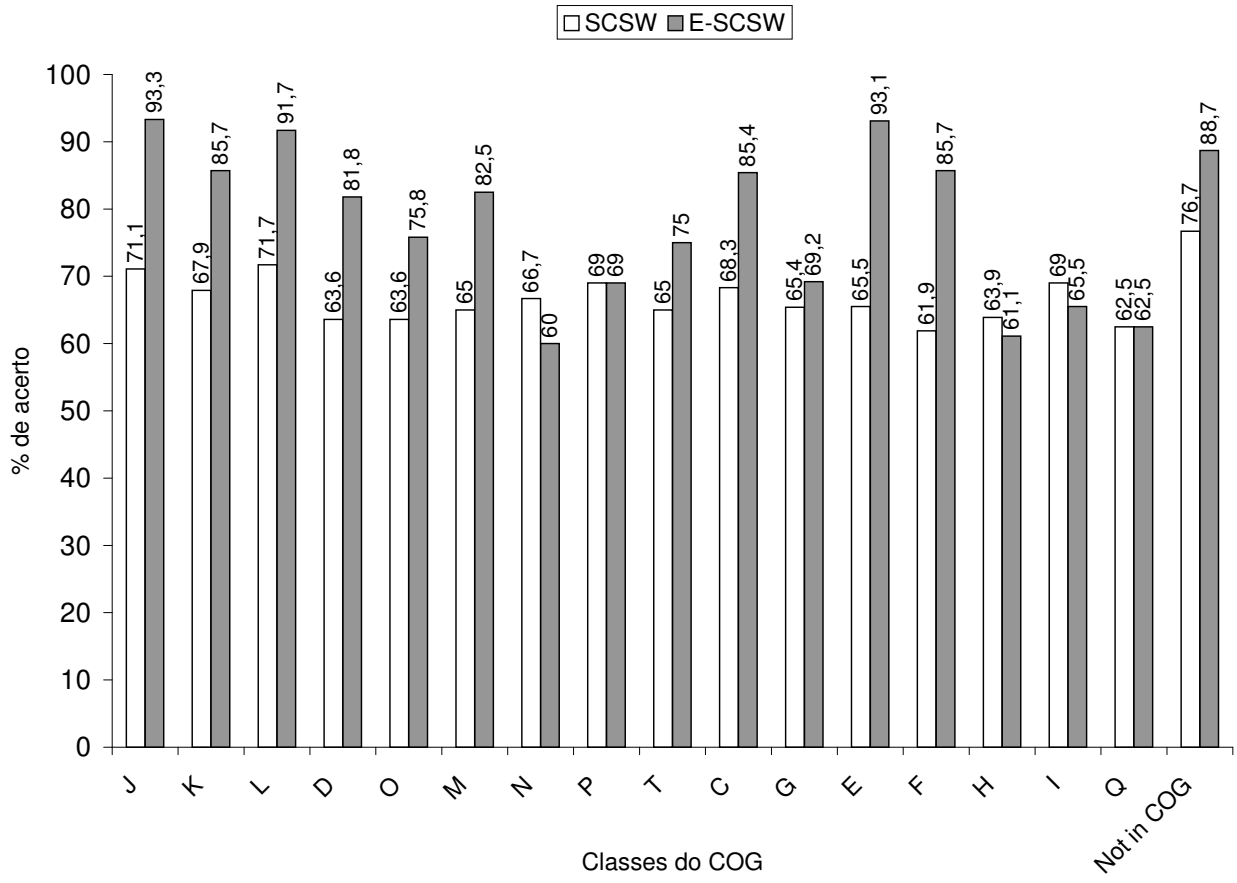


Figura 4.7: Resultado dos testes realizados com as sequências de aminoácidos da *Chlamydophila felis* aplicadas às RNAs que mapeiam cada classe funcional do COG treinadas com os vetores gerados pelos esquemas de codificação *SCSW* e *E-SCSW*. As barras em banco indicam a porcentagem de acerto das RNAs treinadas com os vetores gerados pelo esquema *SCSW*. As barras em cinza indicam a taxa de acerto das RNAs treinadas com os vetores gerados pelo esquema *E-SCSW*. Sobre cada barra é mostrada a porcentagem de acerto da RNA correspondente.

Para cada vetor classificado de forma diferente em relação aos bancos de dados públicos pelas RNAs, a seqüência de aminoácidos correspondente teve sua classificação verificada através do *CD-Search*, similarmente às classificações diferentes dos vetores correspondentes às proteínas da *Chromobacterium violaceum*. A Tabela 4.3 mostra os resultados das análises realizadas em cada seqüência da *Chlamydophila felis* cujo vetor foi classificado de modo diferente pelas RNAs.

A disposição das colunas da Tabela 4.3 é idêntica à disposição das colunas da Tabela 4.2.

Tabela 4.3: Análise das proteínas da *Chamydophila felis* classificadas de maneira diferente pelas RNAs em comparação aos bancos de dados públicos. A primeira coluna indica as 16 classes funcionais do COG sendo que na última linha as classes *R*, *S* e *Not in COG* foram agrupadas em uma só classe indicando sequências de aminoácidos não classificadas. A segunda coluna mostra a quantidade de sequências de aminoácidos analisadas utilizando o *CD-Search*; A terceira coluna mostra a quantidade de sequências de aminoácidos que, depois da análise, se mostraram diferentes em relação aos bancos de dados públicos e que foram classificadas corretamente pelas RNAs; A quarta coluna mostra a quantidade de sequências de aminoácidos cuja classificação foi complementada pelas RNAs, ou seja, sequências de aminoácidos com domínios referentes a mais de uma classe funcional e classificadas em somente uma das classes nos bancos de dados públicos; A última coluna mostra quantas sequências de aminoácidos as RNAs realmente não conseguiram classificar.

Classes Funcionais do COG	Proteínas Analisadas		Classificação correta-RNAs		Complemento à classificação		Classificação incorreta-RNAs	
	SCSW	E-SCSW	SCSW	E-SCSW	SCSW	E-SCSW	SCSW	E-SCSW
J	26	6	0	0	1	3	25	4
K	9	4	0	0	0	2	9	3
L	17	5	0	1	1	1	16	3
D	4	2	0	0	0	0	4	2
O	12	8	0	0	1	4	11	5
M	14	7	0	0	0	4	14	3
N	5	6	0	0	0	2	5	5
P	9	9	0	0	1	4	8	6
T	7	5	0	0	0	1	7	4
C	13	6	0	0	1	3	12	3
G	9	8	0	0	0	5	9	3
E	20	4	0	0	1	2	19	2
F	8	3	0	0	0	2	8	1
H	13	14	0	0	1	8	12	6
I	9	10	0	1	0	5	9	4
Q	3	3	0	0	0	0	3	3
R, S and Not in COG	95	46	0	46	0	0	92	0
Total	273	146	0	48	7	46	263	57

As análises resultaram numa melhora na taxa de acerto de algumas RNAs treinadas com vetores gerados pelo esquema de codificação *SCSW* que variou de $0,7\%$ à $3,4\%$ e para todas as RNAs treinadas com os vetores gerados pelo esquema *E-SCSW* que variou de $2,2\%$ à $22,2\%$, com exceção das RNAs correspondentes às classes *D* e *Q*. A Figura 4.8 mostra o percentual de melhora na taxa de acerto para as RNAs treinadas com os dois esquemas de codificação. As barras em branco indicam o percentual de melhora na taxa de acerto para as RNAs treinadas com os vetores gerados pelo esquema *SCSW*, similarmente, as barras em cinza indicam o percentual de melhora na taxa de acerto para as RNAs treinadas com os vetores gerados pelo esquema *E-SCSW*. Sobre cada barra está o percentual de melhora na taxa de acerto da RNA correspondente.

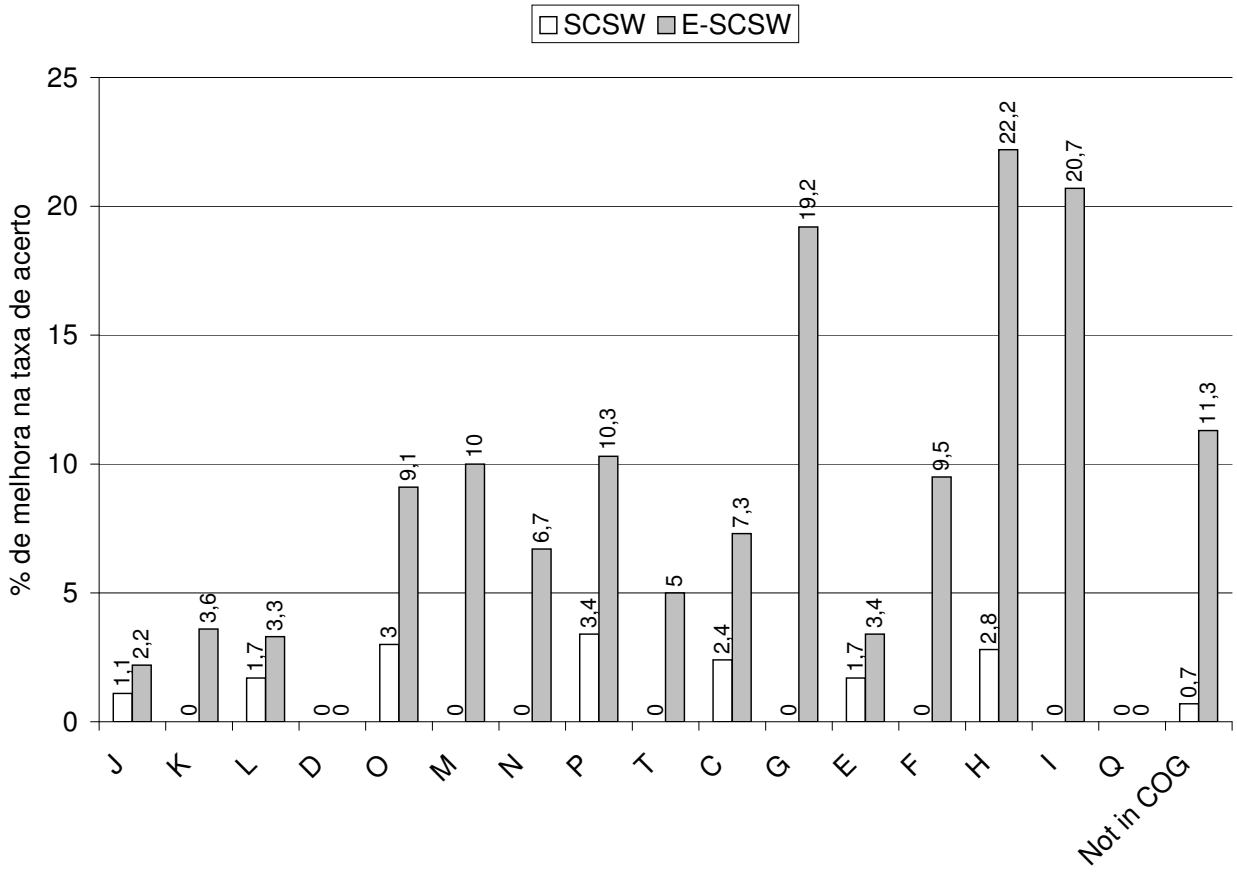


Figura 4.8: Porcentagem de aumento na taxa de acerto das RNAs após a análise, com o *CD-Search* contra o banco de dados do *COG*, das sequências de aminoácidos da *Chlamydophila felis* que foram classificadas de modo diferente pelas RNAs. As barras em branco indicam a melhora na taxa de acerto de cada RNA treinada com os vetores gerados pelo esquema de codificação *SCSW*. As barras em cinza indicam a melhora na taxa de acerto de cada RNA treinada com os vetores gerados pelo esquema de codificação *E-SCSW*. Sobre cada barra é mostrada a porcentagem de melhora após a análise das sequências.

Após as verificações realizadas com o *CD-Search* e atualizando a taxa de acerto de cada RNA, a Figura 4.9 mostra a comparação das taxas de acerto para cada RNA, correspondente às classes funcionais do *COG*, referentes às sequências de aminoácidos da *Chlamydophila felis*.

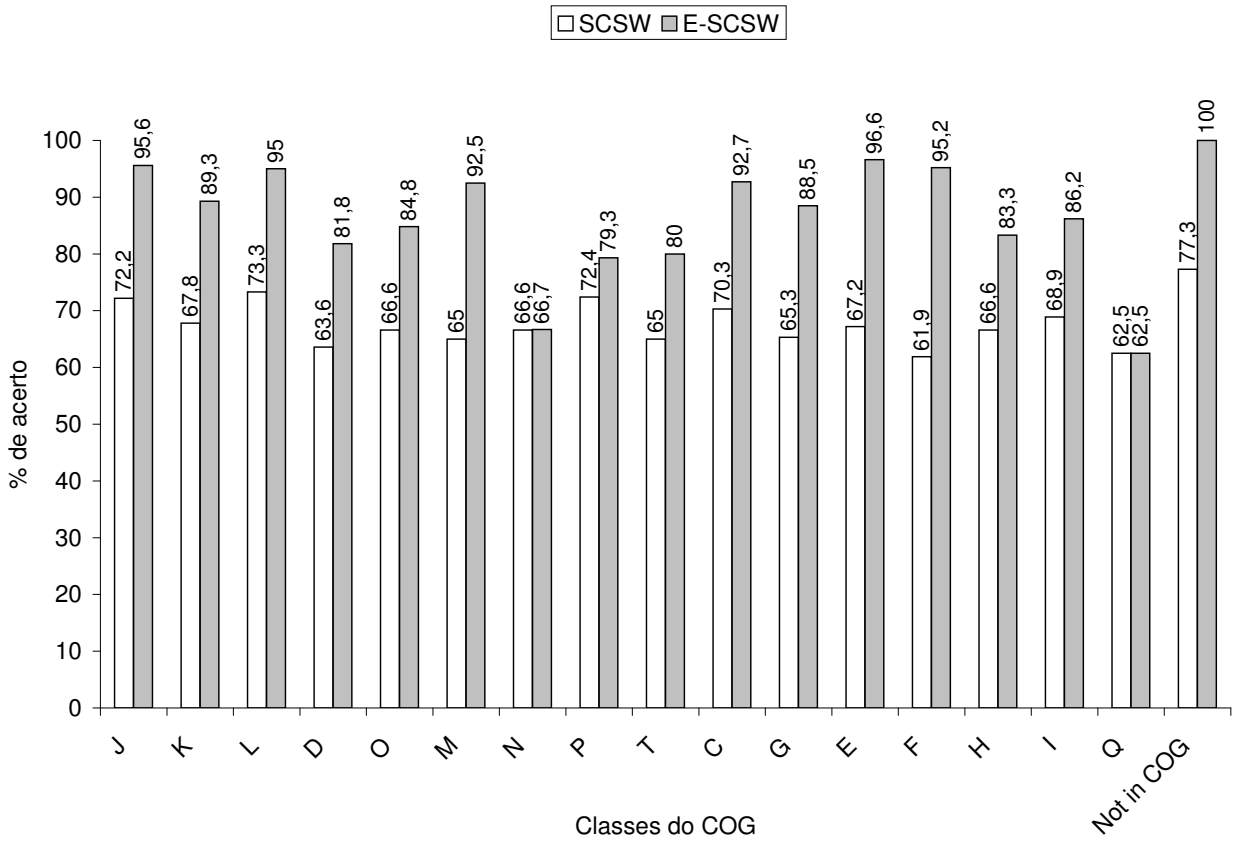


Figura 4.9: Resultado dos testes realizados com as sequências de aminoácidos da *Chlamydophila felis* aplicadas às RNAs que mapeiam cada classe funcional do COG treinadas com os vetores gerados pelos esquemas de codificação *SCSW* e *E-SCSW* após as análises realizadas com o *CD-Search*. As barras em banco indicam a porcentagem de acerto das RNAs treinadas com os vetores gerados pelo esquema *SCSW*. As barras em cinza indicam a taxa de acerto das RNAs treinadas com os vetores gerados pelo esquema *E-SCSW*. Sobre cada barra é mostrada a porcentagem de acerto da RNA correspondente.

A variação na taxa de acerto que era de $61,9\%$ à $76,7\%$ para as RNAs treinadas com o esquema *SCSW* passou a ser de $61,9\%$ à $77,3\%$, enquanto que a taxa de acerto das RNAs treinadas com os vetores gerados pelo esquema de codificação *E-SCSW* que variava de $60,0\%$ à $93,3\%$ passou a variar de $62,5\%$ à 100%.

A Figura 4.10 mostra a análise estatística das taxas de acerto das Redes Neurais Artificiais tendo como estrada as sequências de aminoácidos da *Chlamydophila felis*. Pode ser observada uma diferença significativa entre os dois grupos de RNAs, onde a diferença entre as médias é de $18,68 \pm 2,694$.

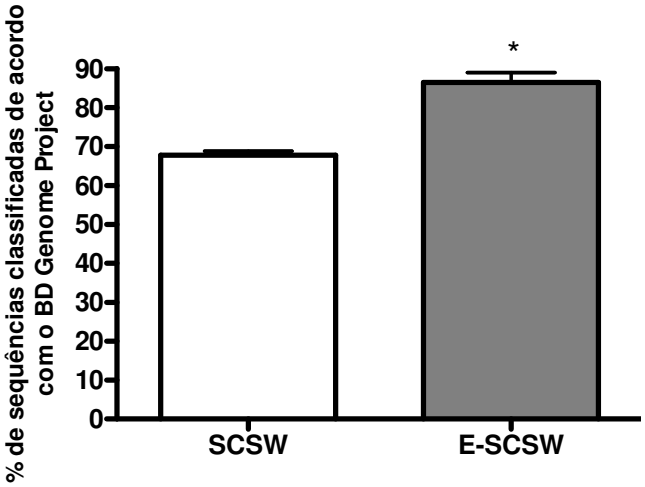


Figura 4.10: Análise estatística entre as taxas de acerto das Redes Neurais Artificiais tendo como estrada as sequências de aminoácidos da *Chlamydophila felis*. As barras representam a média $\pm$ erro-padrão com $n = 17$. A barra em branco corresponde ao resultado das RNAs treinadas com os vetores gerados pelo esquema *SCSW* e a barra em cinza corresponde ao resultado das RNAs treinadas com os vetores gerados pelo esquema *SCSW*; $^{*}p < 0,05\ vs\ SCSW$

Como exemplo de complementação à classificação já existente podemos citar as sequências de aminoácidos *CV3529* (*Chromobacterium violaceum*) e *CF0108* (*Chlamydophila felis*) que são classificadas, nos bancos de dados públicos, como pertencentes à classe *J* (*Translation, ribosomal structure and biogenesis*) e à classe *O* (*Posttranslational modification, protein turnover*), respectivamente.

As RNAs classificaram a seqüência de aminoácidos *CV3529* como pertencentes às classes *J* (Translation, ribosomal structure and biogenesis) e *E* (Amino acid transport and metabolism) sendo este resultado comprovado pelo *CD-Search*, como mostrado na Figura 4.11, onde existe um domínio caracterizando a classe *J* (Translation, ribosomal structure and biogenesis) e um domínio caracterizando a classe *E* (Amino acid transport and metabolism).

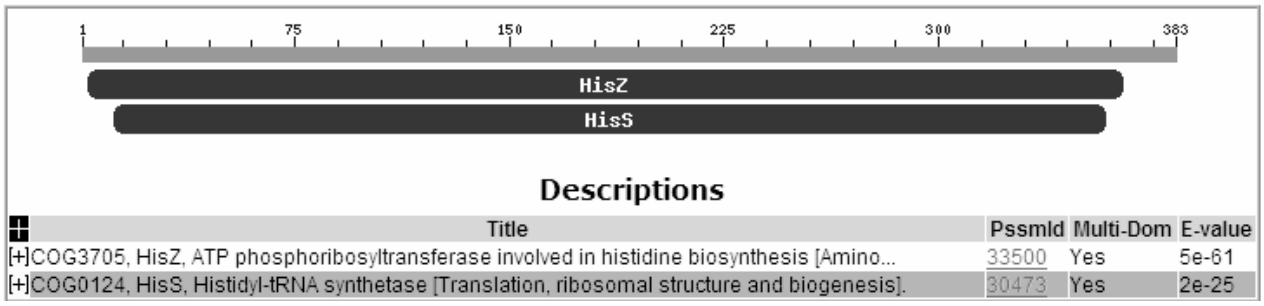


Figura 4.11: Complemento da classificação da proteína *CV3529 - Chromobacerium violaceum*

Similarmente, as RNAs classificaram a seqüência de aminoácidos *CF0108* como pertencentes às classes *C* (*Energy production and conversion*) e *O* (*Posttranslational modification, protein turnover*) sendo o resultado também comprovado pelo *CD-Search*, como mostrado na Figura 4.12, onde existe um domínio caracterizando a classe *C* e um domínio caracterizando a classe *O*.

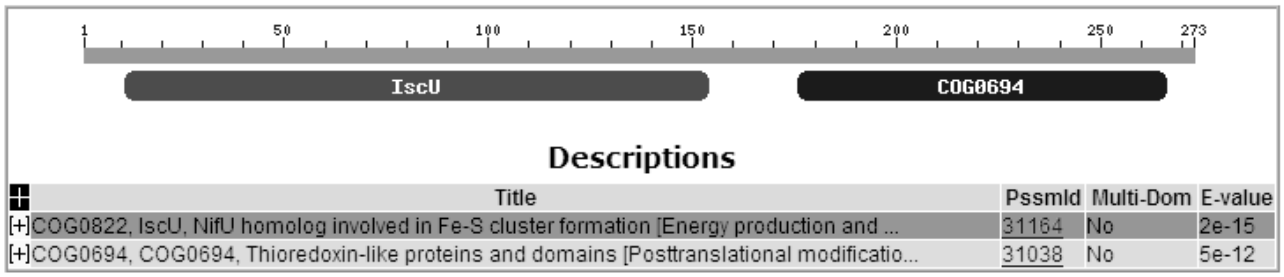


Figura 4.12: Complemento da classificação da proteína *CF0108 - Chlamydophyla felis*

Como exemplo de nova classificação realizada pelas RNAs podemos citar as seqüências de aminoácidos *CV0099* (*Chromobacterium violaceum*) e *CF0019* (*Chlamydophila felis*) que não são classificadas em nenhuma classe funcional do COG nos bancos de dados públicos. Estas sequências foram classificadas nas classes *C* (*Energy production and conversion*) e *H* (*Coenzyme metabolism*), respectivamente, sendo esta classificação comprovada pelo *CD-Search* (Figuras 4.13 e 4.14).

A Figura 4.13 mostra o alinhamento da seqüência de aminoácidos *CV0099* da *Chromobacterium violaceum* que apresenta um domínio com *e-value* $5e^{-143}$ que claramente a identifica como tendo uma função relacionada a *Energy production and conversion*, ou seja, como pertencente a classe *C* do COG.

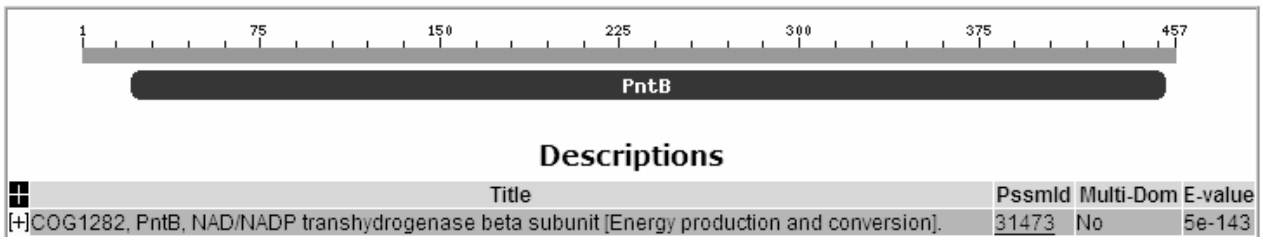


Figura 4.13: Nova classificação da proteína *CV0099 - Chromobacerium violaceum*

Da mesma forma a Figura 4.14 mostra o alinhamento da seqüência de aminoácidos *CF0019* da *Chlamydophila felis* que apresenta um domínio com *e-value* $2e^{-11}$ que a identifica como pertencente à classe *H* do COG.

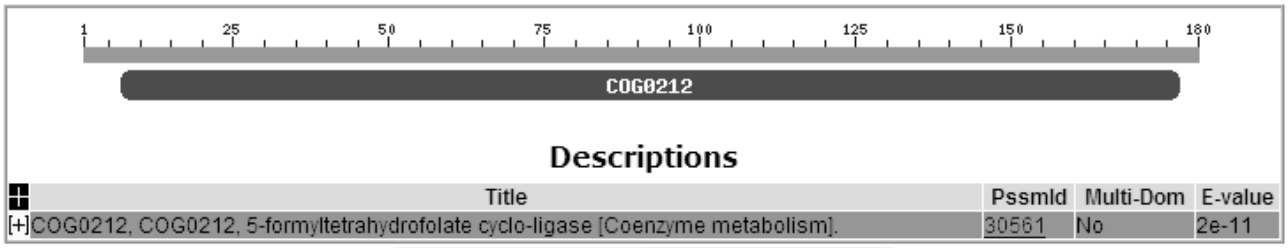


Figura 4.14: Nova classificação da proteína *CF0019 - Chlamydophyla felis*

Por último, como exemplo de correção da classificação existente nos bancos de dados públicos (reclassificação) realizada pelas RNAs podemos citar as sequências de aminoácidos *CV0779* (*Chromobacterium violaceum*) e *CF0217* (*Chlamydophila felis*). Estas sequências são classificadas nos banco de dados públicos como pertencentes às classes *M* (*Cell motility and secretion*) e *L* (*DNA replication, recombination and repair*) respectivamente.

As RNAs classificaram a proteína *CV0779* como pertencente à classe *M* (*Cell envelope biogenesis, outer membrane*) e a proteína *CF0217* como pertencente à classe *D* (*Cell division and chromosome partitioning*) . Estes resultados foram comprovados pelo *CD-Search*, como mostrado nas Figuras **??** e 4.16.

A Figura **??** mostra o alinhamento da seqüência de aminoácidos *CV0779* da *Chromobacterium violaceum* que apresenta um e *e-value* $6e^{-27}$ que a identifica como pertencente à classe *M* do COG.

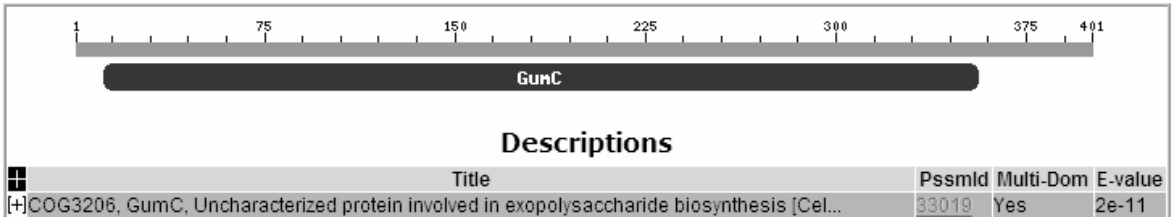


Figura 4.15: Correção da classificação da proteína *CV0779 - Chromobacerium violaceum*

Da mesma forma a Figura 4.16 mostra o alinhamento da seqüência de aminoácidos *CF0217* da *Chlamydophila felis* que apresenta um domínio com *e-value* $1e^{-7}$ que a identifica como pertencente à classe *D* do COG.

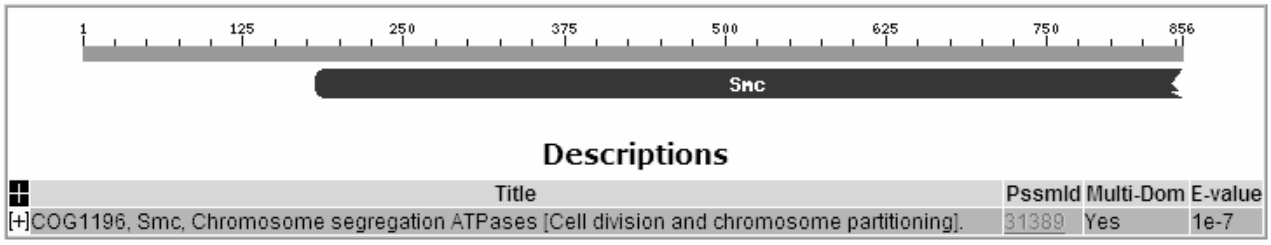


Figura 4.16: Correção da classificação da proteína *CF0217 - Chlamydophyla felis*

As Tabelas 4.4 e 4.5 mostram as sequências de aminoácidos que não estão classificadas nos bancos de dados públicos (*Not in COG*) e que foram classificadas corretamente pelas RNAs treinadas com os vetores gerados pelo esquema de codificação *E-SCSW*. As Tabelas 4.4 e 4.5 correspondem às sequências de aminoácidos da *Chromobacterium violaceum* e da *Chlamydophyla felis*, respectivamente. A primeira coluna contém os identificadores de cada seqüência de aminoácidos classificada e a segunda coluna contém a classificação atribuída a cada seqüência de aminoácidos.

Tabela 4.4: Sequências de aminoácidos da *Chromobacterium violaceum* que não possuem classificação nos banco de dados públicos (*Not in COG*) e que foram classificadas corretamente pelas RNAs treinadas com os vetores gerados pelo esquema de codificação *E-SCSW*. A primeira coluna mostra o código de cada seqüência de aminoácidos correspondente que não está classifica nos bancos de dados públicos. A segunda coluna mostra a classificação de cada seqüência de aminoácidos obtida pelas RNAs e confirmada pelo *CD-Search*.

Identificador da Seqüência	Classe Funcional do COG
CV0003	L
CV0099	C
CV0107	N
CV0164	L
CV0832	N
CV1709	N
CV0172	Q
CV0729	Q
CV0193	J
CV0217	T
CV0491	H
CV0702	K
CV1262	F
CV0911	L
CV1206	J
CV1697	J
CV1697	J
CV1878	M
CV1972	N
CV1984	N
CV2266	I
CV2527	K
CV2607	N
CV2713	O
CV2762	E
CV2974	L
CV3015	G e K
CV3040	G
CV3113	N
CV3525	M
CV3675	J
CV3715	J
CV3798	J
CV4250	P
CV4262	O
CV4324	E

Tabela 4.5: Sequências de aminoácidos da *Chlamydophila felis* que não possuem classificação nos banco de dados públicos (*Not in COG*) e que foram classificadas corretamente pelas RNAs treinadas com os vetores gerados pelo esquema de codificação *E-SCSW*. A primeira coluna mostra o código de cada seqüência de aminoácidos que não está classifica nos banco de dados públicos. A segunda coluna mostra a classificação de cada seqüência de aminoácidos obtida pelas RNAs e confirmadas pelo *CD-Search*.

Identificador da Seqüência	Classe Funcional do COG
CF0011	D
CF0019	H
CF0103	G
CF0114	D
CF0120	H
CF0151	D
CF0173	O
CF0195	J
CF0197	J
CF0245	J
CF0253	J
CF0261	D
CF0272	I
CF0291	E
CF0316	J
CF0317	J
CF0322	G
CF0329	C
CF0336	G
CF0354	J
CF0355	J
CF0356	J
CF0375	Q
CF0458	D
CF0468	G
CF0476	J
CF0477	J
CF0560	H
CF0566	I
CF0630	L
CF0636	M
CF0659	F
CF0692	J
CF0715	J
CF0767	L
CF0809	J
CF0810	J
CF0812	I
CF0817	J
CF0869	H

Tabela 4.5 - continuação

Identificador da Seqüência	Classe Funcional do COG
CF0885	F
CF0954	O
CF0959	J
CF0960	J
CF0998	L
CF1005	H

Todas as análises realizadas com as proteínas da *Chromobacterium violaceum* e com as proteínas da *Chlamydophila felis* referentes à codificação *E-SCSW* estão disponíveis em

www.dcc.ufla.br/∼thiago/e-scsw_chromo.htm e
www.dcc.ufla.br/∼thiago/e-scsw_chlamy.htm,

respectivamente.

Os Apêndices I e II mostram os resultados de todas as análises realizadas com as sequências de aminoácidos que foram classificadas de forma diferente em relação aos bancos de dados públicos pelas RNAs treinadas pelos vetores resultantes do esquema de codificação *E-SCSW*. O Apêndice I mostra o resultado das análises das sequências de aminoácidos da *Chromobacterium violaceum* e o Apêndice II mostra o resultado das análises das sequências de aminoácidos da *Chlamydophila felis*.

Sintetizando os dados das Tabelas 4.2 e 4.3 as Figuras 4.17 (a) e (b) mostram a quantidade de sequências de aminoácidos que tiveram sua classificação complementada pelas RNAs para a *Chromobacterium violaceum* e *Chlamydophila felis*, respectivamente. As Figuras 4.17 (a) e (b) fazem uma comparação entre os resultados das RNAs treinadas com os vetores gerados pelo esquema de codificação *SCSW* e *E-SCSW*.

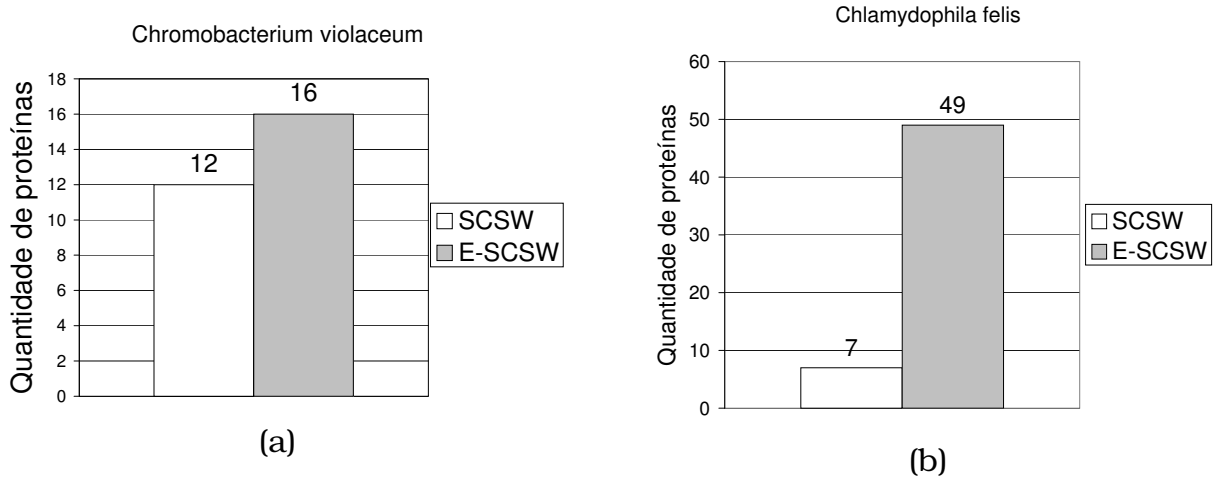
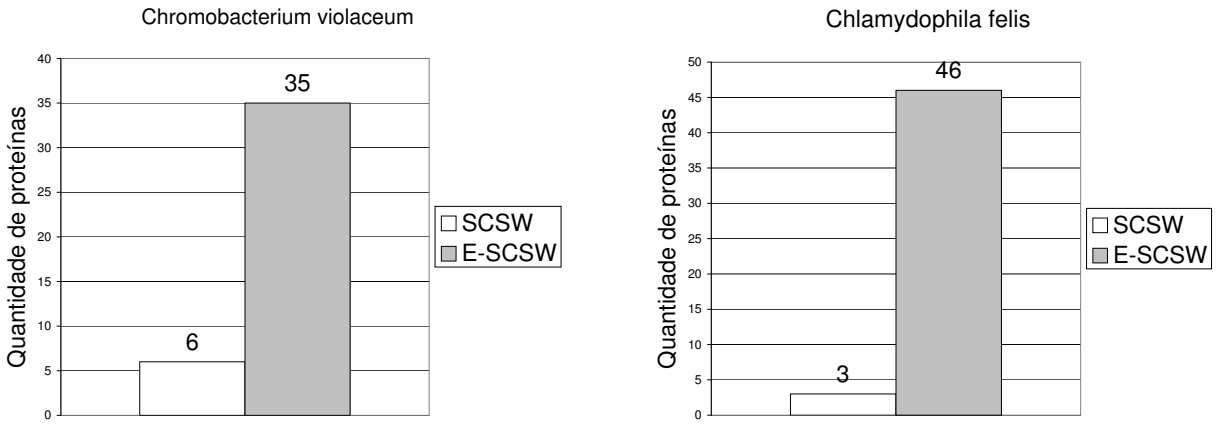


Figura 4.17: Em (a) é mostrada a quantidade de sequências de aminoácidos da *Chromobacterium violaceum* que tiveram sua classificação complementada pelas RNAs. Em (b) é mostrada a quantidade de sequências da *Chlamydophila felis* que tiveram sua classificação complementada pelas RNAs. As barras em branco indicam a quantidade de complementos de classificação realizados pelas RNAs treinadas com os vetores gerados pelo esquema de codificação *SCSW*. As barras em cinza indicam a quantidade de complementos de classificação realizados pelas RNAs treinadas com os vetores gerados pelo esquema de codificação *E-SCSW*.

Da mesma forma, as Figuras 4.18 (a) e (b) mostram a quantidade de sequências de aminoácidos que não possuíam classificação e que foram classificadas corretamente pelas RNAs para a *Chromobacterium violaceum* e *Chlamydophila felis*, respectivamente. As Figuras 4.18 (a) e (b) fazem uma comparação entre os resultados das RNAs treinadas com os vetores gerados pelo esquema de codificação *SCSW* e *E-SCSW*.



(a) (b)

Figura 4.18: Em (a) é mostrada a quantidade de sequências de aminoácidos da *Chromobacterium violaceum* que foram classificadas pelas RNAs. Em (b) é mostrada a quantidade de sequências da *Chlamydophila felis* que foram classificadas pelas RNAs. No dois casos as sequências de aminoácidos estão classificadas como *Not in COG* nos bancos de dados públicos. As barras em branco indicam a quantidade classificações realizadas pelas RNAs treinadas com os vetores gerados pelo esquema de codificação *SCSW*. As barras em cinza indicam a quantidade de classificações realizadas pelas RNAs treinadas com os vetores gerados pelo esquema de codificação *E-SCSW*.

Para as proteínas classificadas e que tiveram sua classificação modificada (reclassificadas), 131 proteínas analisadas da *Chromobacterium violaceum* e 2 proteínas analisadas da *Chlamydophila felis* tiveram sua classificação modificada pelas RNAs treinadas com o esquema *E-SCSW*. As RNAs treinadas com o esquema *SCSW* não modificaram a classificação de nenhuma proteína analisada.

4.2.3 Teste com seqüências ambíguas

Todas as 70 seqüências que não foram utilizadas no treinamento pelo fato de serem ambíguas (Tabela 3.4) foram utilizadas para testar as RNAs.

A Tabela 4.6 mostra os resultados dos testes realizados com as seqüências ambíguas para os esquemas *SCSW* e *E-SCSW*.

- A primeira coluna mostra as classes funcionais do COG;
- A segunda coluna mostra a quantidade de sequências de aminoácidos ambíguas em cada classe funcional do COG, totalizando 70 sequências;
- A terceira coluna mostra a quantidade de proteínas que foram classificadas corretamente pelas RNAs treinadas com os vetores gerados pelos esquemas *SCSW* e *E-SCSW*;

Todas as sequências de aminoácidos ambíguas testadas foram analisadas individualmente através do *CD-Search*, onde não foi detectada nenhuma classificação de sequências de aminoácidos não-classificadas, nenhuma complementação à classificação e nenhuma reclassificação de sequências de aminoácidos já classificadas.

Tabela 4.6: Resultados dos testes com sequências de aminoácidos ambíguas. A primeira coluna mostra as classes funcionais do COG, a segunda coluna mostra a quantidade de sequências de aminoácidos ambíguas em cada classe funcional do COG, totalizando 70 sequências e a terceira coluna mostra a quantidade de proteínas que foram classificadas corretamente pelas RNAs treinadas com os vetores gerados pelos esquemas *SCSW* e *E-SCSW*.

Classes Funcionais do COG	Proteínas testadas	Classificação correta-RNAs	
		SCSW	E-SCSW
J	2	0	0
K	3	0	2
L	3	0	3
D	2	0	1
O	5	2	3
M	5	0	2
N	5	1	2
P	6	1	3
T	4	0	2
C	7	2	4
G	7	0	3
E	5	0	2
F	6	1	3
H	3	1	2
I	4	0	2
Q	1	0	0
R, S and Not in COG	2	0	0

A Figura 4.19 mostra a comparação entre a taxa de acerto das RNAs treinadas com os vetores gerados pelo esquema *SCSW* e *E-SCSW* para cada classe funcional do COG.

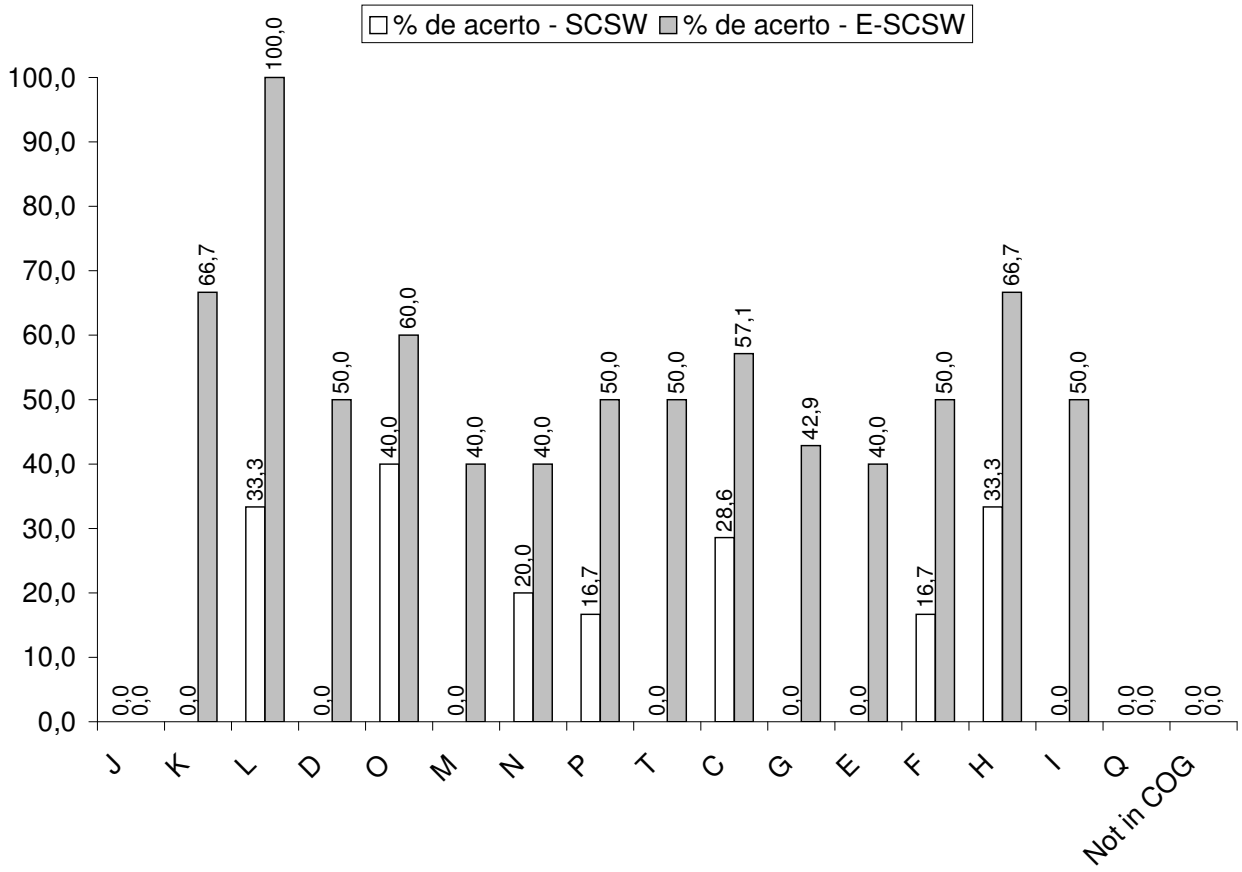


Figura 4.19: Comparação entre as taxas de acerto das RNAs treinadas com os vetores gerados pelos esquemas *SCSW* x *E-SCSW* referente às seqüências de aminoácidos ambíguas. As barras em branco mostram os resultados das RNAs treinadas com os vetores gerados pelo esquema de codificação *SCSW*. As barras em cinza mostram os resultados das RNAs treinadas com os vetores gerados pelo esquema de codificação *E-SCSW*. Sobre cada barra é mostrado a taxa de acerto da RNA correspondente.

No Capítulo 5 é realizada a discussão dos resultados encontrados e conclusões finais a respeito desse trabalho de tese.

CAPÍTULO 5

Discussão e Conclusões

Neste capítulo discutimos os resultados obtidos pelo teste realizado com o esquema de codificação *SCSW* assim como os resultados da comparação dos esquemas de codificação *SCSW* e *E-SCSW*. Na última seção é apresentada conclusão final desse trabalho de tese.

5.1 Discussão

O teste realizado com o esquema *SCSW* utilizando janela deslizante de tamanho $n = 2$ (Seção 3.1) mostrou que, apesar desse esquema de codificação ser útil para a determinação de similaridade entre seqüências, ele não possui a mesma acurácia dos métodos tradicionais de alinhamento par-a-par, pois alguns dos agrupamentos não foram coerentes com o resultado do *ClustalW*. A Tabela 4.1 mostra os agrupamentos encontrados pelo *K-means* que foram coerentes com os domínios do *PFAM*. O agrupamento *Taeniidae* apresenta 18 seqüências de aminoácidos com um domínio em comum e, adicionalmente, 3 seqüências sem nenhum domínio que não foram mostradas na tabela. Este agrupamento pode ser um indício de que essas 3 seqüências possam fazer parte da família *Taeniidae* mas o agrupamento não foi confirmado pelo *ClustalW*. Ainda na Tabela 4.1 um agrupamento contendo 5 seqüências sem domínios foi encontrado pelo *K-means*. Este agrupamento pode indicar que estas 5 seqüências pertencem à mesma família e são funcionalmente relacionadas entretanto não existe nenhum domínio que comprove essa relação.

A ambigüidade das seqüências foi levantada como um dos prováveis motivos do

agrupamento incorreto de algumas seqüencias. Os testes com janelas deslizantes de tamanhos $n = 3$ à $n = 10$ mostraram que os agrupamentos se mantinham para janelas deslizantes de tamanho até $n = 6$ e que para janelas deslizantes maiores os resultados pioravam. Logo, verificamos que, com o aumento do tamanho da janela deslizante a ambiguidade era minimizada, entretanto pequenas regiões de similaridade deixavam de ser consideradas, ou seja, era necessário considerar mais de um tamanho de janela deslizante. Além disso, deve ser dada uma maior relevância às janelas deslizantes maiores pois estas indicam uma maior identidade entre as seqüências comparadas, logo, se utilizarmos mais de uma janela deslizante, deve ser atribuído um peso proporcional ao seu tamanho, no caso do presente trabalho, o peso foi o tamanho da janela deslizante. Uma outra questão diz respeito ao resultado do agrupamento ter se mantido para janelas deslizantes de tamanho $n = 2$ à $n = 6$. O motivo pode estar no fato de que somente aminoácidos idênticos eram considerados na comparação entre os vetores resultantes. A similaridade entre aminoácidos diferentes deve ser levada em consideração pois proteínas com a mesma função não necessariamente possuem a mesma seqüência de aminoácidos, e sim, podem ter aminoácidos similares em posições específicas que caracterizam os domínios da seqüência.

Os resultados do teste com o esquema de codificação *SCSW* mostraram que esse esquema é útil para a determinação de similaridade entre seqüências, como mostrado em outros trabalhos (Petrilli, 1993), (Blaisdell, 1986), (Blaisdell, 1989b), (Blaisdell, 1989a). Entretanto os resultados não possuem uma acurácia compatível com os métodos tradicionais de alinhamento par-a-par, *FASTA* (Pearson, 1990) e *BLAST* (Altschul et al., 1990), como destacado em (Wu et al., 1992).

Visto isto, propusemos o esquema *E-SCSW* como uma alternativa ao esquema *SCSW*. A comparação entre os dois esquemas de codificação (Seção 3.3) mostrou que o método proposto é superior ao método *SCSW*, quando os vetores resultantes são utilizados para treinar RNAs. O treinamento com os vetores resultantes do esquema proposto possibilitou que as RNAs realizassem uma melhor separação quando consideramos as classes funcionais do COG. Analisando o resultado dos testes realizados, podemos verificar que a taxa de acerto das RNAs treinadas com os vetores gerados pelo esquema *E-SCSW* é superior à das RNAs treinadas com os vetores gerados pelo esquema *SCSW* (Figuras 4.5 e 4.9). A única exceção diz respeito à classe *Q* da *Chlamydophila felis*, para a qual ambos os métodos resultaram na mesma taxa de acerto (Figura 4.9).

Utilizado-se o mesmo tamanho de janelas deslizantes, o mesmo alfabeto e sendo evitado a ambigüidade para os dois esquemas de codificação, o esquema *E-SCSW*

proporcionou um resultado superior, em média 30% para as seqüências da *Chromobacterium violaceum* e 18% em média para as seqüências da *Chlamydophila felis*. Dois motivos aparentes podem ser levantados: o primeiro diz respeito ao peso atribuído às janelas deslizantes, onde uma maior relevância passou a ser dada às janelas maiores. Isto possibilita que seqüências de aminoácidos com domínios similares irão possuir um maior peso nas posições que tiverem em comum nos vetores resultantes correspondentes. Desta forma, possivelmente um melhor agrupamento de seqüências de aminoácidos similares é obtida. O segundo motivo diz respeito à dimensionalidade dos vetores resultantes. Quando é utilizado mais de um tamanho de janela deslizante, o esquema *SCSW* gera vetores com maior dimensionalidade que os gerados pelo esquema *E-SCSW*. Além disso, os vetores gerados pelo esquema *SCSW* são mais esparsos. Portanto as RNAs treinadas com os vetores gerados pelo esquema *SCSW* têm uma maior dificuldade em realizar a separação das classes em relação àquelas treinadas com os vetores gerados pelo esquema *E-SCSW*.

É importante notar que as proteínas pertencentes a cada classe funcional do COG não são rigorosamente similares entre si. Cada classe funcional do COG é formada por grupos de seqüências de aminoácidos que possuem a mesma função, onde cada grupo é denominado COG, possuindo uma identificação particular. Na base de dados do COG somam-se 138.458 seqüências de aminoácidos que são agrupadas em 4.873 COGs (Tatusov et al., 2003). Os COGs com funções correlatas estão agrupados em superclasses formando as 18 classes funcionais[1]. Conseqüentemente, os vetores gerados pelos dois esquemas de codificação referentes às seqüências de cada classe funcional do COG não estão distribuídos em um único agrupamento. A Figura 5.1 mostra, esquematicamente, uma visão *incorreta* da distribuição dos vetores gerados a partir dos membros de duas classes quaisquer do COG, onde os vetores pertencentes à *Classe 1*, representados por ◯ e os vetores pertencentes à *Classe 2*, representados por □, estão agrupados de acordo com a similaridade. Na verdade os membros de uma classe não são, necessariamente, similares entre si.

Uma visão mais realista da distribuição dos vetores gerados pelos dois esquemas de codificação referentes às proteínas de cada classe funcional é a disposição destes vetores em pequenos clusters, como mostrado na Figura 5.2. Os vetores pertencentes à *Classe 1*, representados por ◯, e os vetores pertencentes à *Classe 2*, representados por □, estão agrupados em pequenos grupos (branco, preto, cinza e listrado), onde cada um destes grupos corresponde a um COG da classe funcional.

[1] http://www.ncbi.nlm.nih.gov/COG/old/palox.cgi?fun=all

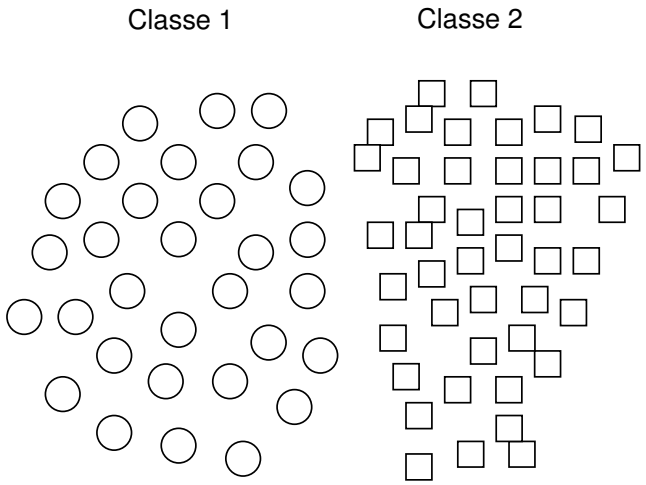


Figura 5.1: Distribuição *incorreta* dos vetores gerados pelos esquemas de codificação referentes às duas classes funcionais do COG. As seqüências de uma classe qualquer do COG não são, necessariamente, similares entre si. Portando os vetores correspondentes a *Classe 1*, representados por ◯, e os vetores correspondentes à *Classe 2*, representados por □, não se apresentam , necessariamente, agrupados como na figura.

O resultado dos testes mostrou que as RNAs treinadas com os vetores resultantes do esquema de codificação *E-SCSW* foram capazes de realizar uma melhor separação do conjunto de agrupamentos pertencentes a cada classe funcional do COG, em comparação ao esquema *SCSW*. Conseqüentemente, o esquema proposto é mais adequado em reter as informações de um conjunto de seqüências de modo que RNAs possam realizar sua classificação de maneira mais eficiente. Mesmo para seqüências ambíguas onde o esquema de codificação proposto possibilitou que $48,5\%$ das seqüências testadas fossem classificadas corretamente pelas RNAs contra $12,8\%$ para as RNAs treinadas com o esquema *SCSW*. Entretanto podemos observar pelos testes que a taxa de acerto de todas as RNAs é inferior quando utilizamos seqüências ambíguas em comparação à utilização de seqüências não-ambíguas. Este resultado já era esperado pois a composição de uma proteína e a ordem em que os aminoácidos aparecem é o que determina sua função. Quando existe ambigüidade o vetor resultante não corresponde a uma seqüência de aminoácidos única comprometendo, em alguns casos, a configuração de alguns domínios existentes na seqüência original e conseqüentemente a determinação da sua função.

Considerando o caso deste trabalho de tese onde uma RNA mapeia uma classe contra todas as outras, a indefinição na configuração de alguns domínios pode fazer com que três casos ocorram quando seqüências ambíguas são utilizadas:

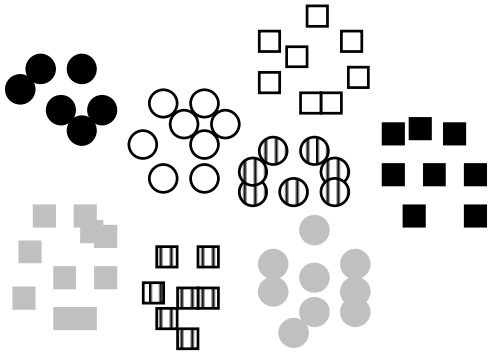


Figura 5.2: Distribuição mais realista dos vetores gerados pelos esquemas de codificação referentes à duas classes funcionais do COG. Um classe funcional é composta de vários COG's, os quais contém um conjunto de seqüências similares. Portando os vetores correspondentes à *Classe 1*, representados por ◯, e os vetores correspondentes à *Classe 2*, representados por □, se apresentam em pequenos grupos correspondentes às seqüências similares.

1. os vetores correspondentes às seqüências ambíguas ficam na margem de separação entre as duas classes resultando em uma indefinição na pertinência em uma das classes;

2. os vetores correspondentes às seqüências ambíguas passam a pertencer à classe que não é a original, indicando a pertinência em qualquer classe não mapeada pela RNA em questão;

3. os vetores correspondentes às seqüências ambíguas conseguem manter a informação de parte do domínio que caracteriza funcionalmente a seqüências correspondente. Esta informação pode ser suficiente para classificar corretamente a seqüência.

É importante notar que a ambigüidade pode afetar a configuração de qualquer parte da seqüência original. Se o domínio que classifica essa seqüência não for afetado, provavelmente o vetor resultante vai ser classificado corretamente pelas RNAs, como ocorreu em parte das 70 seqüências ambíguas testadas Seção 4.2.3.

O esquema proposto também se mostrou superior em determinar as inconsistências encontradas nos bancos de dados públicos quando todas as seqüências foram analisadas utilizando o *CD-Search* ($3,7\%$ do total das proteínas da *Chromobacterium violaceum*, (Tabela 4.2) e $4,7\%$ do total das proteínas da *Chlamydophila felis*, (Tabela 4.3)).

Se compararmos o número de proteínas que tiveram sua classificação complementada pelas RNAs, ou seja, aquelas seqüências que pertencem a uma classe funcional do COG nos bancos de dados públicos mas possuem domínios que a

classificam em outras classes (Figuras 4.17 (a) e (b)), pode-se perceber que, para as duas bactérias, o número de proteínas foi maior para as RNAs treinadas com os vetores gerados pelo esquema de codificação proposto. O esquema de *E-SCSW* possibilitou a complementação na classificação de 54 sequências contra 19 complementações na classificação obtidas a partir do esquema *SCSW*.

Para as proteínas que haviam sido classificadas pelos autores dos estudos sobre o genoma da *Chromobacterium violaceum* e da *Chlamydophila felis* e que tiveram sua classificação modificada utilizando o nosso método (reclassificadas), as RNAs treinadas com os vetores gerados pelo esquema *E-SCSW* foram mais eficientes. Das seqüências analisadas, 131 da *Chromobacterium violaceum* e 2 da *Chlamydophila felis* tiveram sua classificação modificada. Nenhuma modificação foi sugerida pelas RNAs treinadas pelo esquema de codificação *SCSW*. Adicionalmente, das 131 seqüências da *Chromobacterium violaceum* que sofreram modificação na classificação, 99 seqüências foram classificadas, pelo nosso método, como não pertencentes ao *COG*, sendo este resultado comprovado pela análise individual das seqüências utilizando o *CD-Seach*.

Uma última observação sobre as análises realizadas, diz respeito às seqüências classificadas como não pertencentes a nenhuma classe do *COG* e que foram classificadas neste trabalho como pertencente a uma das classes funcionais (Figuras 4.18 (a) e (b)). Para ambas as bactérias, esse número é maior para as RNAs treinadas com os vetores resultantes do esquema de codificação *E-SCSW*, sendo estas novas classificações também comprovadas pela análise individual utilizando o *CD-Seach*.

Antes dos testes serem realizados, esperava-se que um número maior de seqüências da *Chromobacterium violaceum* pudessem ter sua classificação modificada quando comparadas com seqüências da *Chlamydophila felis*. Isso porque as RNAs foram treinadas com seqüências depositadas em 2005, ou seja, um banco de dados bem mais atualizado se comparado ao banco de dados na época em que as seqüências do genoma da *Chromobacterium violaceum* foram anotadas e depositadas (2003). As seqüências do genoma da *Chlamydophila felis* foram anotadas e depositadas em 2006. Era portanto de se esperar que novos domínios proteicos pudessem ter sido acrescentados aos bancos de dados neste intervalo de tempo e que a presença de novos domínios pudessem facilitar a classificação de algumas seqüências. Entretanto os testes mostraram que nos dois casos, várias seqüências sofreram modificações na classificação.

A anotação das seqüências da *Chromobacterium violaceum* foi realizada através da busca de similaridade de cada seqüência contra toda a base de dados do COG

(Vasconcelos et al., 2003) utilizando um programa denominado SABIA (Almeida et al., 2004) o qual possui vários módulos baseados no BLAST. Da mesma forma, a análise individual de cada seqüência classificada de forma incongruente pelas RNAs foi realizada pelo *CD-Search* utilizando toda a base do COG, a qual sofreu uma atualização em 2003 (Tatusov et al., 2003), no mesmo ano em que os dados do genoma da *Chromobacterium violaceum* foram publicados. Portanto, o motivo para o número maior de complementação, reclassificação e classificação de seqüências da *Chromobacterium violaceum* pelas RNAs possivelmente se deve à inserção de novas seqüências no banco de dados público do COG, cujos domínios não estavam disponíveis quando o genoma da *Chromobacterium violaceum* foi anotado.

Para as seqüências complementadas, reclassificadas e classificadas da *Chlamydophila felis* pelas RNAs a mesma justificada não pode ser utilizada pelo fato das seqüências terem sido anotadas após a atualização sofrida pelo COG. Uma possível causa pode ser o uso inadequado de alguma ferramenta de anotação, onde a utilização de valores pouco rígidos de alguns parâmetros torne possível o aparecimento de falsos positivos. No caso do *CD-Search*, por exemplo, um valor inferior à $0,01$ para *Expected Value* pode resultar em alinhamentos inconsistentes[2]. Como em (Vasconcelos et al., 2003) em (Azuma et al., 2006) não são apresentados detalhes sobre a anotação dos genomas da *Chromobacterium violaceum* e da *Chlamydophila files*, isso impossibilita uma investigação mais detalhada sobre os motivos das complementações, reclassificações e classificações realizadas pelas RNAs.

5.2 Conclusões finais

Os testes realizados mostraram que existem seqüências depositadas nos bancos de dados públicos que estão classificadas de maneira incosistente (Kyrpides and Ouzounis, 1999), (Pallen et al., 1999) e (Karp, 1998). O principal motivo é que cada nova seqüência depositada tem sido anotada levando em consideração as próprias seqüências nos bancos de dados públicos, possibilitando uma transição de erros de anotação (Karp, 1998).

Verificar a acurácia da anotação de um genoma completo ou até mesmo de algumas poucas seqüências não é uma tarefa simples. As primeiras publicações onde foram descritos estudos de genomas normalmente não trazem detalhes dos procedimentos utilizados na etapa de anotação das seqüências, quais foram os métodos computacionais utilizados (embora o BLAST seja o mais comum), qual o valor dos

[2]http://www.ncbi.nlm.nih.gov/Structure/cdd/cdd_help.shtml

parâmetros utilizados, ou qualquer informação na qual o pesquisador interessado possa medir a confiabilidade da anotação. Entretanto, vale ressaltar que nos últimos anos está se tornando comum a disponibilização de material suplementar que possibilita aos autores disponibilizar dados que não podem ser acomodados nas publicações (Santos et al., 2005), como em (Vasconcelos et al., 2003) onde todos os detalhes do resultado da anotação da *Chromobacterium violaceum* estão disponibilizados em *http://www.brgene.lncc.br/cviolaceum*.

Da mesma forma é difícil estimar a confiabilidade de alguns bancos de dados de seqüências pois estes fornecem pouca informação de como foram criados e como as seqüências são avaliadas antes de serem inseridas nestes bancos de dados. Por outro lado, alguns bancos de dados de seqüências são construídos de maneira mais rigorosa e possuem seqüências cuja classificação é mais confiável. O COG é um exemplo onde, em sua maioria, a função das proteínas ou é conhecida através de experimentos ou pela significância da similaridade de seqüências com proteínas já classificadas (Tatusov et al., 1997).

Uma tentativa de se evitar a propagação de anotações incorretas de novas seqüências deve ser realizada com ferramentas e um conjunto com banco de dados confiáveis os quais devem ser utilizados para verificar a classificação de seqüências já depositadas assim como classificar as novas entradas. Os métodos tradicionais de alinhamento par-a-par, especificamente o *BLAST* e suas variações (Altschul et al., 1997), são tidos como a melhor solução para busca de similaridade e posterior classificação funcional de proteínas. Entretanto, anotações de seqüências que utilizaram ferramentas baseadas no *BLAST* estão sujeitas a falhas e devem ser inspecionadas manualmente.

O esquema de *E-SCSW* se mostrou superior ao esquema *SCSW* no que tange à extração de informações da seqüência de aminoácidos original. O método *E-SCSW* se mostrou mais capaz de gerar vetores de modo que esses facilitem o mapeamento das classes por parte das RNAs. Esta mapeamento proporciona um melhor resultado no treinamento e testes das RNAs que tiveram como entrada os vetores gerados pelo esquema *E-SCSW* em comparação ao esquema *SCSW*. Logo, o esquema de codificação de seqüências *E-SCSW* e posterior classificação dos vetores resultantes por RNAs é apresentado aqui como um complemento aos métodos tradicionais de alinhamento par-a-par, capaz de detectar vária incoerências geradas por anotações realizadas com base no uso do *BLAST*.

Portanto o uso em conjunto do método proposto e de ferramentas tradicionais de anotação baseadas em alinhamento par-a-par se mostra-se extremamente útil a ser utilizado em uma etapa de verificação de seqüências já anotadas assim como

para evitar erros de anotação em novas seqüências. Os resultados combinados das duas metodologias podem resultar em uma maior confiabilidade na classificação ou na necessidade de uma análise mais detalhada da classificação realizada.

Propostas de Continuidade

Sugere-se como propostas para continuação deste trabalho de tese, investir nos seguintes problemas relacionados ao tema:

- Analisar cada seqüência antes de utilizá-las para treinamento das Redes Neurais Artificiais;
- Selecionar um conjunto de treinamento que mapeie o maior número de *COGs* possível de modo que seqüências de outros organismos, além de bactérias, possam ser aplicadas à metodologia;
- Expandir o método de modo que as RNAs mapeiem as classes do *KOG*, *Clusters of orthologous groups* para seqüências de eucariotos, de modo que seqüências de eucariotos possam ser aplicadas à metodologia;
- Implementar o método de modo que seqüências possam ser aplicadas através de uma interface web;
- Implementar um sistema que possa realizar a verificação de um conjunto de seqüências já depositados nos bancos de dados públicos de maneira automatizada, através da metodologia proposta;
- Investigar o problema de divergência entre seqüências já que é um problema ainda em aberto;

Referências

Almeida, L. G. P., Paixão, R., Souza, R. C., Costa, G. C., Barrientos, F. J. A., Santos, M. T., Almeida, D. F., and Vasconcelos, A. T. R. (2004). A system for automated bacterial (genome) integrated annotation sabia. *Bioinformatics*, 20:2832–2833.

Altschul, S. F., Gish, W., Miller, W., Meyers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410.

Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402.

Arthur, M. (2002). *Introduction to Bioinformatics*. Oxford University Press Inc., New York.

Azuma, Y., Hirakawa, H., Yamashita, A., Cai, Y., Rahman, M. A., Suzuki, H., Mitaku, S., Toh, H., Goto, S., Murakami, T., Sugi, K., Hayashi, H., Fukushi, H., Hattori, M., Kuhara, S., and Shirai, M. (2006). Genome sequence of the cat pathogen, chlamydophila felis. *DNA Research*, 13:15–23.

Baldi, P. and Brunak, S. (2001). *Bioinformatics, the machine learning approach*. Massachusetts Institute of Technology, 2 edition.

Blaisdell, B. E. (1986). A measure of the similarity of sets of sequences not requiring sequence alignment. *Proc. Natl. Acad. Sci. USA*, 83.

Blaisdell, B. E. (1989a). Average values of a dissimilarity measure not requiring sequence alignment are twice the averages of conventional mismatch counts requiring sequence alignment for a computer-generated model system. *Journal of Molecular Evolution*, 29.

Blaisdell, B. E. (1989b). Effectiveness of measures requiring and not requiring prior sequence alignment for estimating the dissimilarity of natural sequences. *Journal of Molecular Evolution*, 29.

Bohr, H., Bohr, J., Brunak, S., Cotteril, R. M. J., Lautrup, B., Norskov, L., Oslen, O. H., and Petersen, S. B.

Braga, A. P., Carvalho, A. F., and Ludermir, T. B. (2000). *Redes Neurais Artificiais: Teoria e Aplicações*. Livros Técnicos e Científicos.

Brunak, S., Engelbrecht, J., and Knudsen, S. (1991). Prediction of human mrna donor and acceptor sites from the dna sequence. *J. Mol. Biol.*, 220:49–65.

Cherkassky, V. and Mulier, F. (1998). *Learning From Data: Concepts, Theory, and Methods*. John Wiley Sons.

Childers, S. E., Ciufo, S., and Lovley, D. R. (2002). Geobacter metallireducens accesses insoluble fe(iii) oxide by chemotaxis. *Nature*, 416:767–769.

Dayhoff, M. O. (1978). Survey of new data and computer methods of analysis. *Atlas of protein sequence and structure*, 5.

Eidhammer, I., Jonassen, I., and Taylor, W. R. (2004). *Protein Bioinformatics An Algorithmic Approach to Sequence and Structure Analysis.* John Willey.

Ewens, W. J. and Grant, G. R. (2001). *Statistical Methods in Bioinformatics*. Springer-Verlag.

Gibas, C. and Jambeck, P. (2001). *Developing Bioinformatics Skills*. O'Reilly.

Gibbis, A. J. and Cohen, M. A. (1970). The diagram, a method for comparing sequences. *Eur. J. Biochem*, (16):1–11.

H Nielsen, J Engelbrecht, S. B. G. v. H. (1997). A neural network method for identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. *Int J Neural Syst*, (8):581–599.

Hart, P. E. (1968). The condensed nearest neighbour rule. *IEEE Transactions Information Theory*, 1(14).

Haykin, S. (1999). *Neural Networks: a comprehensive foundation*. 2 edition.

Henikoff, S. and Henikoff, J. G. (1992). Amino acid substitution matrices from protein blocks. *Proceedings of the national academy of sciences of the united states of america*, 22(89):10915–10919.

Hide, W., Burke, J., and Davison, D. B. (1994). Biological evaluation of d2, an algorithm for high-performance sequence comparison. *Journal of Computional*

Biology, 3(1):199–215.

Holdena, M. T. G., Titballb, R. W., Peacockd, S. J., Cerdeño-Tárraga, A. M., Atkinsb, T., Crossmana, L. C., Pittf, T., Churchera, C., Mungalla, K., Bentleya, S. D., Sebaihiaa, M., Thomsona, N. R., Basona, N., Beachamg, I. R., Brooksa, K., Brownh, K. A., Browng, N. F., Challisi, G. L., Cherevacha, I., Chillingwortha, T., Cronina, A., Crossetth, B., Davisa, P., DeShazerj, D., Feltwella, T., Frasera, A., Hancea, Z., Hausera, H., Holroyda, S., Jagelsa, K., Keithh, K. E., Maddisona, M., Moulea, S., Pricea, C., Quaila, M. A., Rabbinowitscha, E., Rutherforda, K., Sandersa, M., Simmondsa, M., Songsivilaik, S., Stevensa, K., Tumapae, S., Vesaratchaveste, M., Whiteheada, S., Yeatsa, C., Barrella, B. G., Oystonb, P. C. F., , and Parkhill, J. (2004). Genomic plasticity of the causative agent of melioidosis, burkholderia pseudomallei. *Proceedings of National Academy of Science of the United States of America*, 101:14240–14245.

Holley, L. H. and Karplus, M.

Hsu, C. and Lin, C. (2002). A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425.

Jeong, H., Yim, J. H., Lee, C., Choi, S., Park, Y. K., Yoon, S. H., Hur, C., Kang, H., Kim, D., Lee, H. H., Park, K. H., Park, S., Park, H., Lee, H. K., Oh, T. K., and Kim, J. F. (2005). Genomic blueprint of hahella chejuensis, a marine microbe producing an algicidal agent. *Nucleic Acids Res*, 33(22):7066–7073.

Joardar, V., Lindeberg, M., Jackson, R. W., Selengut, J., Dodson, R., Brinkac, L. M., Daugherty, S. C., DeBoy, R., Durkin, A. S., Giglio, M. G., Madupu, R., Nelson, W. C., Rosovitz, M. J., Sullivan, S., Crabtree, J., Creasy, T., Davidsen, T., Haft, D. H., Zafar, N., Zhou, L., Halpin, R., Holley, T., Khouri, H., Feldblyum, T., White, O., Fraser, C. M., Chatterjee, A. K., Cartinhour, S., Schneider, D. J., Mansfield, J., Collmer, A., and Buell1, C. R. (2005). Whole-genome sequence analysis of pseudomonas syringae pv. phaseolicola 1448a reveals divergence among pathovars in genes involved in virulence and transposition. *Journal of Bacteriology*, 187(18):6488–6498.

Kanehisa, M. and Bork, P. (2003). Bioinformatics in the post-genomic era. *NATURE*, 33:305–310.

Karp, P. D. (1998). What we do not know about sequence analysis and sequence databases. *Bioinformatics*, 14:753–754.

Kim, H. S., Schell, M. A., Yu, Y., Ulrich, R. L., Sarria, S. H., Nierman, W. C., and DeShazer, D. (2005). Bacterial genome adaptation to niches: Divergence of the

potential virulence genes in three burkholderia species of different survival strategies. *BMC Genomics*, 6:1–13.

Kork, I., Yandell, M., and Bedell, J. (2003). *BLAST*. O'Reilly.

Kyrpides, N. C. and Ouzounis, C. A. (1999). Whole-genome sequence annotation: Going wrong with confidence. *Molecular Microbiology*, 32:886–887.

Likas, A., Vlassis, N., and Verbeek, J. J. (2003). The global k-means clustering algorithm. *Pattern Recognition*, (36-2):451–461.

Mackay, D. (1992). Bayesian interpolation. *Neural Computation*, 4(3):415–447.

Marchler-Bauer, A. and Bryant, S. H. (2004). Cd-search: protein domain annotations on the fly. *Nucleic Acids Res*, 32(Web Server issue):W327–31.

Matsunaga, T., Okamura, Y., Fukuda, Y., Wahyudi, A. T., Murase, Y., and Takeyama, H. (2005). Complete genome sequence of the facultative anaerobic magnetotactic bacterium magnetospirillum sp. strain amb-1. *DNA Research*, 12(3):157–166.

Methé, B. A., Nelson, K. E., Deming, J. W., Momen, B., Melamud, E., Zhang, X., Moult, J., Madupu, R., Nelson, W. C., Dodson, R. J., Brinkac, L. M., Daugherty, S. C., Durkin, A. S., DeBoy, R. T., Kolonay, J. F., Sullivan, S. A., Zhou, L., Davidsen, T. M., Wu, M., Huston, A. L., Lewis, M., Weaver, B., Weidman, J. F., Khouri, H., Utterback, T. R., Feldblyum, T. V., and Fraser, C. M. (2005). The psychrophilic lifestyle as revealed by the genome sequence of colwellia psychrerythraea 34h through genomic and proteomic analyses. *Proc Natl Acad Sci USA*, 102:10913–10918.

Mongodin, E. F., Nelson, K. E., Daugherty, S., DeBoy, R. T., Wister, J., Khouri, H., Weidman, J., Walsh, D. A., Papke, R. T., Perez, G. S., Sharma, A. K., NesbÃ¸, C. L., MacLeod, D., Bapteste, E., Doolittle, W. F., Charlebois, R. L., Legault, B., and Rodriguez-Valera, F. (2005). The genome of salinibacter ruber: Convergence and gene exchange among hyperhalophilic bacteria and archaea. *Proc Natl Acad Sci USA*, 102(50):18147–18152.

Mount, D. W. (2004). *Bioinformatics, Sequence and Genome Analisys*. Cold Spring Harbor Laboratory Press, New York.

Needleman, S. and Wunsch, C. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453.

Nielsen, H., Engelbrecht, J., Brunak, S., and von Heijne, G.

Pallen, M., Wren, B., and Parkhill, J. (1999). Going wrong with confidence: misleading sequence analyses of ciab and clpx. *Molecular Microbiology*, 34(1):195.

Paulsen, I. T., Press, C. M., Ravel1, J., Kobayashi, D. Y., Myers, G. S. A., Dmitri V Mavrodi4, Robert T DeBoy1, R. S. Q. R. R. M., Dodson1, R. J., Durkin, A. S., Brinkac, L. M., Daugherty, S. C., Sullivan, S. A., Rosovitz, M. J., Gwinn, M. L., Zhou, L., Schneider, D. J., Cartinhour, S. W., Nelson, W. C., Weidman, J., Watkins, K., Tran, K., Khouri, H., Pierson, E. A., III, L. S. P., Thomashow, L. S., and Loper, J. E. (2005). Complete genome sequence of the plant commensal pseudomonas fluorescens pf-5. *Nature Biotechnology*, 23:873–878.

Pearson, W. R. (1990). Rapid and sensitive sequence comparison with fastp and fasta. *Methods Enzymol*, (183):63–98.

Pearson, W. R., Wood, T., Zang, Z., and Miller, W. (1997). Comparison of dna sequence with protin sequences. *Genomics*, (46):24–36.

Pedersen, A. G. and Nielsen, H. (1997). Neural network prediction of translation initiation sites in eukaryotes: perspectives for est and genome analysis. *Proc Int Conf Intell Syst Mol Biol*, (5):226–233.

Petrilli, P. (1993). Classification of protein sequences by their dipeptide comnposition. *CABIOS*, (2):205–209.

Pevzner, P. A. (1995). Dna physical mapping and alternating eulerian cycles in colored graphs. *Algorithmica*, 13:77–105.

Qian, W., Jia1, Y., Ren, S.-X., He, Y.-Q., Feng, J.-X., Lu, L.-F., Sun, Q., Ying, G., Tang, D.-J., Tang, H., Wu, W., Hao, P., Wang, L., Jiang, B.-L., Zeng, S., Gu, W.-Y., Lu, G., Rong, L., Tian, Y., Yao, Z., Fu, G., Chen, B., Fang, R., Qiang, B., Chen, Z., Zhao, G.-P., Tang, J.-L., and He, C. (2005). Comparative and functional genomic analyses of the pathogenicity of phytopathogen xanthomonas campestris pv. campestris. *Genome Research*, 15:757–767.

Reinert, G., Schbath, S., and Waterman, M. S. (2000). Probabilistic and statistical properties of words: An overview. *Journal of Computational Biology*, 7(1-2):1–46.

Rodrigues, T. S., Braga, A. P., Pacífico, L. G., Teixeira, S. M. R., and Oliveira, S. C. (2003a). Amino acid coding with sliding window technique. Procedings of Workshop of Bioinformatics.

Rodrigues, T. S., Braga, A. P., Pacífico, L. G., Teixeira, S. M. R., and Oliveira, S. C. (2003b). Clustering and artificial neural networks: Classification of variable

lengths of helminth antigens in set of domains. Procedings of International Conference of Bioinformaticas and Computational Biology.

Rodrigues, T. S., Braga, A. P., Pacífico, L. G., Teixeira, S. M. R., and Oliveira, S. C. (2004). Clustering and artificial neural networks: Classification of variable lengths of helminth antigens in set of domains. *Genetics and Molecular Biology*, 4(27):673–678.

Rodrigues, T. S., Braga, A. P., Teixeira, S. M. R., and Oliveira, S. C. (2005). Protein classification with extended sequence coding by sliding window. In *Research in Computational Molecular Biology*. Broad Institute of MIT and Harvard / Boston University's Center for Advanced Genomic Technology. Poster aceito para apresentação oral.

Santos, C., Blake, J., and States, D. J. (2005). Supplementary data need to be kept in public repositories. *Nature*, 438:8.

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

Snyder, E. E. and Stormo, G. D. (1995). Identification of protein coding regions in genomic dna. *J. Mol. Biol.*, 248:1–18.

Stormo, G. D., Schneider, T. D., Gold, L., and Ehrenfeucht, A.

Stormo, G. D., Schneider, T. D., and Gold, L. M. (1982). Characterization of translational initiation sites in e. coli. *Nucleid Acid Research*, 19:2971–2996.

Stryer, L., Berg, J. M., and Tymoczko, J. L. (2002). *Biochemistry*. Freeman, New York, 5 ediï¿½o edition.

Tatusov, R. L., Fedorova, N. D., Jackson, J. D., Jacobs, A. R., Kiryutin, B., Koonin, E. V., Krylov, D. M., Mazumder, R., Mekhedov, S. L., Nikolskaya, A. N., Rao, B. S., Smirnov, S., Sverdlov, A. V., Vasudevan, S., Wolf, Y. I., Yin, J. J., and Natale, D. A. (2003). The cog database: an updated version includes eukaryotes. *BMC Bioinformatics*, 4:1–14.

Tatusov, R. L., Koonin, E. V., and Lipman, D. J. (1997). A genomic perspective on protein families. *Science*, (278):631–637.

Tettelin, H., Masignani, V., Cieslewicz, M. J., Donati, C., Medini, D., Ward, N. L., Angiuoli, S. V., Crabtree, J., Jones, A. L., Durkin, A. S., DeBoy, R. T., Davidsen, T. M., Mora, M., Scarselli, M., y Ros, I. M., Peterson, J. D., Hauser, C. R., Sundaram, J. P., Nelson, W. C., Madupu, R., Brinkac, L. M., Dodson, R. J., Rosovitz, M. J., Sullivan, S. A., Daugherty, S. C., Haft, D. H.,

Selengut, J., Gwinn, M. L., Zhou, L., Zafar, N., Khouri, H., Radune, D., Dimitrov, G., Watkins, K., O'Connor, K. J. B., Smith, S., Utterback, T. R., White, O., Rubens, C. E., Grandi, G., Madoff, L. C., Kasper, D. L., Telford, J. L., Wessels, M. R., Rappuoli, R., and Fraserabkm, C. M. Genome analysis of multiple pathogenic isolates of streptococcus agalactiae: Implications for the microbial.

Thompson, J. D., Higgins, D. G., and Gibson, T. J. (1994). Clustalw: imporving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specifc gap penalties and weight matrix choice. *Neural Computation*, 22(22):4673–4680.

Vasconcelos, A. T. R., Almeida, D. F., Hungria, M., Guimarães, C. T., Antônio, R. V., and et. al. (2003). The complete genome sequence of chromobacterium violaceum reveals remarkable and exploitable bacterial adaptability. *Proceedings of National Academy of Science of the United States of America*, 100(20):11660–11665.

Vinga, S. and Almeida, J. (2003). Alignment-free sequence comparison-a review. *Biometrics*, (4):513–523.

Wu, C., Ermongkonchai, A., and Chang, T. (1991a). Protein classification using a neural network protein database (nnpdb) system. *Proc. Anal. Neural Net. Appl. Conf.*, pages 29–41.

Wu, C., McLarty, J., and Whitson, G. (1991b). Neural networks for molecular sequence database management. *Proc. ACM 19th Comp. Sci. Conf.*, pages 588–594.

Wu, C., Whitson, G., McLarty, J., Ermongkonchai, A., and Chang, T. (1992). Protein classification artificial neural system. *Protein Science*, (1):667–677.

Wu, C. H. (1997). Artificial neural networks for molecular sequence analysis. *Computers Chemistry*, 21(4):237–256.

Wu, M., R., Q., Durkin, A. S., Daugherty, S. C., Brinkac, L. M., Dodson, R. J., Madupu, R., Sullivan, S. A., Kolonay, J. F., Nelson, W. C., Tallon, L. J., Jones, K. M., Ulrich, L. E., Gonzalez, J. M., Zhulin, I. B., Robb, F. T., and Eisen1, J. A. (2005). Life in hot carbon monoxide: The complete genome sequence of carboxydothermus hydrogenoformans z-2901. *PLoS Genetics*, 1:563–574.

Wu, T. J., Burke, J., and Davison, D. B. (1997). A measure of dna sequence dissimilarity based on mahalanobis distance between frequencies of words. *Biometrics*, 53:1431–1439.

Yang, F., Yang, J., Zhang, X., Chen, L., Jiang, Y., Yan, Y., Tang, X., Wang, J., Xiong, Z., Dong, J., Xue, Y., Zhu, Y., Xu, X., Sun, L., Chen, S., Nie, H., Peng, J., Xu, J., Wang, Y., Yuan, Z., Wen, Y., Yao, Z., Shen, Y., Qiang, B., Hou, Y., Yu, J., and Jin, Q. (2005). Genome dynamics and diversity of shigella species, the etiologic agents of bacillary dysentery. *Nucleic Acids Res*, 33(19):6445–6458.

Zvelebil, M. J., Barton, G. J., Taylor, W. R., and Sternberg, M. J. E. (1987). Prediction of protein secondary structure and active sites using the alignment of homologous sequences. *Journal of Molecular Biology*, 4:957–961.

Apêndice I - Resultado das análises realizadas com as sequências de aminoácidos da Chromobacterium violaceum

A Tabela abaixo mostra o resultado das análises realizadas com as sequências de aminoácidos da *Chromobacterium violaceum* que foram classificadas de forma incoerente com os bancos de dados públicos pelas RNAs treinadas com os vetores resultantes do esquema de codificação *E-SCSW*. Somente os resultados corretos por parte das RNAs são mostrados na tabela abaixo. A Tabela está dividida da seguinte forma:

- A primeira coluna mostra a classe funcional do COG na qual a seqüência de aminoácidos em questão está classificada nos bancos de dados públicos.
- A segunda coluna mostra a código identificador da ORF correspondente à seqüência de aminoácidos analisada.
- A terceira coluna mostra o resultado classificação realizada pelas RNAs e confirmadas pelo *CD-Search*.

Classe funcional do COG nos repositórios públicos	Identificador da ORF	Resultado classificação realizada pelas RNAs
C	CV2151	Not in COG
C	CV2777	Not in COG
C	CV2986	Not in COG
C	CV3166	Not in COG
C	CV3543	D
C	CV4114	Not in COG
C	CV4201	L
D	CV0647	Not in COG
D	CV1477	Not in COG
D	CV2149	Not in COG
D	CV2155	Not in COG
D	CV2264	Not in COG
D	CV2285	Not in COG
D	CV2668	Not in COG
D	CV2971	Not in COG
E	CV1340	Not in COG
E	CV1554	Not in COG
E	CV1715	Not in COG
E	CV1824	Not in COG
E	CV1888	Not in COG
E	CV2908	Not in COG
E	CV2948	E and T
E	CV4130	E and J
E	CV4213	Not in COG
E	CV4298	Not in COG
E	CV4306	Not in COG
E	CV4367	Not in COG
E	CV4370	Not in COG

Classe funcional do COG nos repositórios públicos	Identificador da ORF	Resultado classificação realizada pelas RNAs
F	CV0279	Not in COG
F	CV3746	E and F
F	CV4082	Not in COG
F	CV4248	Not in COG
F	CV4330	O
G	CV2434	Not in COG
G	CV3990	Not in COG
H	CV3955	Not in COG
H	CV4210	Not in COG
H	CV4231	Not in COG
H	CV4313	E
H	CV4320	E
H	CV4335	Not in COG
I	CV0538	Not in COG
I	CV2450	Not in COG
I	CV4291	G
I	CV4315	Not in COG
J	CV0467	Not in COG
J	CV0474	Not in COG
J	CV2011	Not in COG
J	CV3529	E and J
J	CV3609	Not in COG
J	CV4265	Not in COG
K	CV0333	E
K	CV0468	J
K	CV0532	M
K	CV1438	E
K	CV1536	Not in COG
K	CV1731	Not in COG
K	CV1836	E
K	CV2076	E
K	CV2190	Not in COG
K	CV2337	Not in COG
K	CV2374	Not in COG
K	CV2444	Not in COG
K	CV2469	E
K	CV2584	F and K

Classe funcional do COG nos repositórios públicos	Identificador da ORF	Resultado classificação realizada pelas RNAs
K	CV2785	Not in COG
K	CV2952	Not in COG
K	CV3126	Not in COG
K	CV3388	M
K	CV3622	Not in COG
K	CV4116	M
K	CV4321	Not in COG
K	CV4331	Not in COG
K	CV4366	E
L	CV0364	Not in COG
L	CV1399	Not in COG
L	CV1405	L and F
L	CV1928	L and F
L	CV1939	Not in COG
L	CV2805	Not in COG
L	CV2995	Not in COG
L	CV3076	Not in COG
L	CV3385	L and F
L	CV3398	Not in COG
L	CV3590	Not in COG
L	CV4072	L and F
L	CV4223	Not in COG
M	CV0108	Not in COG
M	CV0348	Not in COG
M	CV1971	Not in COG
M	CV1983	Not in COG
M	CV2185	Not in COG
M	CV2263	Not in COG
M	CV2912	Not in COG
M	CV3179	Not in COG
M	CV3353	M and D
M	CV3538	I
M	CV3617	Not in COG
M	CV4254	Not in COG
M	CV4302	Not in COG
M	CV4349	D
M	CV4351	Not in COG

Classe funcional do COG nos repositórios públicos	Identificador da ORF	Resultado classificação realizada pelas RNAs
N	CV0414	Not in COG
N	CV0772	M
N	CV1859	Not in COG
N	CV1916	N and T
N	CV2065	Not in COG
N	CV2120	Not in COG
N	CV2218	E
N	CV2593	Not in COG
N	CV2947	Not in COG
N	CV3874	Not in COG
N	CV4054	M
N	CV4079	Not in COG
N	CV4080	Not in COG
N	CV4083	Not in COG
O	CV1175	E
O	CV1960	Not in COG
O	CV1990	Not in COG
O	CV2490	Not in COG
O	CV3460	D
P	CV3937	E
P	CV3981	Not in COG
P	CV4245	Not in COG
P	CV4251	Not in COG
P	CV4284	Not in COG
P	CV4389	L
Q	CV0334	H
Q	CV0463	J
Q	CV0466	M
Q	CV1045	Not in COG
Q	CV1255	H
Q	CV1545	Q and H
Q	CV1741	Not in COG
Q	CV2028	Not in COG
Q	CV2749	H
Q	CV3474	Q and H
Q	CV4293	Not in COG
Q	CV4378	T
Q	CV4398	Not in COG
Q	CV4400	Not in COG
T	CV0439	K and T
T	CV2931	K and T
T	CV4260	Not in COG

Apêndice II - Resultado das análises realizadas com as sequências de aminoácidos da Chlamydophila felis

A Tabela abaixo mostra o resultado das análises realizadas com as sequências de aminoácidos da *Chlamydophila felis* que foram classificadas de forma incoerente com os bancos de dados públicos pelas RNAs treinadas com os vetores resultantes do esquema de codificação *E-SCSW*. Somente os resultados corretos por parte das RNAs são mostrados na tabela abaixo. A Tabela está dividida da seguinte forma:

- A primeira coluna mostra a classe funcional do COG na qual a seqüência de aminoácidos em questão está classificada nos bancos de dados públicos.
- A segunda coluna mostra a código identificador da ORF correspondente à seqüência de aminoácidos analisada.
- A terceira coluna mostra o resultado classificação realizada pelas RNAs e confirmadas pelo *CD-Search*.

Classe funcional do COG nos repositórios públicos	Identificador da ORF	Resultado classificação realizada pelas RNAs
C	CF0108	C and O
C	CF0679	C and G
C	CF0789	C and I
E	CF0064	E and J
E	CF0648	E and I
F	CF0254	F and H
F	CF0358	F and P
G	CF0193	G and M
G	CF0371	G and K
G	CF0457	G and E
G	CF0576	G and I
G	CF0673	G and T
G	CF0753	G and C
H	CF0017	H and K
H	CF0118	H and F
H	CF0137	H and C
H	CF0170	H and E
H	CF0295	H and E
H	CF0297	H and M
H	CF0491	H and O
H	CF0803	H and G
I	CF0199	I and E
I	CF0454	I and C
I	CF0522	I and Q
I	CF0620	Q
I	CF0699	I and G and H
I	CF0845	I and H and M

Classe funcional do COG nos repositórios públicos	Identificador da ORF	Resultado classificação realizada pelas RNAs
J	CF0024	J and F
J	CF0482	J and C
K	CF0876	K and N and C
L	CF0164	L and F and I
L	CF0217	D
M	CF0147	M and H
M	CF0152	M and G
M	CF0225	M and E
M	CF0836	M and E
N	CF0970	N and C
O	CF0108	O and C
O	CF0231	O and E
O	CF0765	O and I
P	CF0167	P and G
P	CF0268	P and F
P	CF0813	P and C
T	CF0157	T and P