Um Algoritmo Evolucionário para a Identificação de Seqüências de Aminoácidos na Estrutura Primária de Proteínas

Denise F. Tsunoda[1], Heitor S. Lopes[2] e Alex A. Freitas[3]

[1,2]Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial (CPGEI),
Centro Federal de Educação Tecnológica do Paraná (CEFET-PR), Brasil, 80230-901
Fone: +55 41 310 4694, Fax: +55 41 310 4683
[3] Programa de Pós-Graduação em Informática Aplicada (PPGIA),
Pontifícia Universidade Católica do Paraná (PUC-PR), Brasil, 80215-901
Fone: +55 41 330-1669, Fax: +55 41 330-1669
denise@cpgei.cefetpr.br, hslopes@cepgei.cefetpr.br, alex@ppgia.pucpr.br

Resumo - As proteínas podem ser agrupadas em famílias e estas em superfamílias de acordo com algumas características como hidrofobicidade, composição, carga elétrica (por exemplo, ponto isoelétrico), estrutura, comprimento ou função, com o objetivo de estabelecer as funções biológicas comuns. Este artigo apresenta o resultado do uso de um Algoritmo Genético (AG) modificado (uma vez que emprega operadores de busca global (mutação e *crossover*) e local (expansão)) na detecção e análise de seqüências de aminoácidos idênticas ou muito semelhantes (*motifs*), extraídas a partir do Protein Databank (PDB), objetivando a implementação de uma ferramenta capaz de classificar automaticamente proteínas. O acerto preditivo foi medido pelo uso do procedimento de validação cruzada de dez partições. A taxa de acerto de classificação do algoritmo implementado foi medida com a variação de alguns parâmetros, tendo oscilado entre 87,78% a 94,85%. Os testes foram efetuados sobre um subconjunto de classes de proteínas do PDB. Cada uma das classes utilizadas possuía, no mínimo, 10 proteínas.

Palavras-chave: Proteína, Seqüência de Aminoácidos, Estrutura Primária, Algoritmo Genético.

Abstract – Proteins can be grouped into families and these families into superfamilies according to some features such as hydrophobicity, composition, electric charge (e.g.: iso-electric point), structure, length or function, with the purpose of establishing the common biological functions. This paper shows the results of the use of a modified genetic algorithm (GA) once it uses global (mutation, crossover) and local search (expansion) operators) in the searching and analysis of identical or very similar amino acids sequences (called motifs) extracted from the proteins databank (PDB) with the purpose of implementing a tool capable of automatically classifying new proteins. The predictive accuracy rate was measured by the use of a well known validation procedure, named 10-cross validation. The classification accuracy of the implemented algorithm was measured varying some of its parameters. Classification accuracy rates between 87.78% and 94.85% were obtained, taken into account a subset of classes of proteins found in the PDB. Each one of them had, at least, 10 proteins.

Key-words: Protein, Amino Acids Sequence, Primary Structure, Genetic Algorithm.

Introdução

As proteínas são responsáveis por diversas funções biológicas: o transporte de pequenas moléculas (por exemplo, a hemoglobina), regulagem (por exemplo, a insulina), sustentação (colágeno), aumento da velocidade da reação (enzimas) e outras. Organismos biológicos possuem milhares de diferentes tipos de proteínas, as quais são constituídas basicamente de aminoácidos ligados em cadeias lineares por meio de ligações peptídicas. Forças intramoleculares ativas fazem com que a proteína assuma uma configuração tridimensional específica que está diretamente relacionada as suas funções biológicas (Lehninger *et al*, 1998).

Dentre as diversas abordagens que estão sendo estudadas para comparação de proteínas (Wang *et al*, 2000, Greenwood *et al.*, 1999, Piccolboni, Mauri, 1998; Lu, Fu, 1978), este artigo discorre sobre o desenvolvimento de uma ferramenta computacional chamada GAPC (*Genetic Algorithm-based Protein Classification*), bem como os resultados obtidos com seu uso. Esta ferramenta foi concebida para localizar seqüências de aminoácidos (padrões) que ocorrem muito

freqüentemente nas proteínas de uma dada classe (família), mas que raramente ocorrem em proteínas de outras classes. Tais padrões podem ser usados para proceder com a classificação automática de novas proteínas. De forma a localizar tais padrões, a ferramenta consiste de um algoritmo genético (GA) desenvolvido especificamente para a classificação de proteínas.

Os algoritmos genéticos foram inventados por Holland (1975) e subseqüentemente aprimorados por diversos pesquisadores, tais como Do Jong (1975), Goldberg (1989), Davis (1991) e Michalewicz (1996). Os AGs têm sua premissa nos processos adaptativos da natureza, como outros algoritmos evolucionários tais como a Programação Genética (Koza, 1982), a Programação Evolucionária (Fogel *et al.*, 1966), as Estratégias Evolutivas (Rechenberg, 1986) e outros.

Um Algoritmo Genético inicia com uma população formada de indivíduos gerados aleatoriamente que podem ser vistos como candidatos a soluções do problema alvo. Durante cada geração do processo evolucionário, cada indivíduo da população é avaliado por uma função própria chamada função de *fitness*, que mede quão boa é a solução candidata representada pelo indivíduo. De uma geração a outra, alguns indivíduos "pais" (normalmente aqueles que apresentam maior *fitness*) produzem uma nova geração, ou seja, novos indivíduos (soluções candidatas) que herdam algumas das boas características de seus antepassados, enquanto que outros (com *fitness* baixo) são descartados, seguindo a teoria da seleção natural de Darwin. Note-se que os melhores indivíduos da geração corrente são escolhidos para reprodução de forma que a próxima geração tende a ter novos indivíduos com melhores *fitness*, em média, que os da geração predecessora. A geração de novos descendentes a partir dos progenitores da geração corrente ocorre por meio de operadores genéticos. Após um processo seletivo, o GAPC faz uso dos operadores de mutação, *crossover* (permuta) e expansão, estudados posteriormente. Este processo é iterativamente repetido até que uma solução satisfatória seja encontrada ou um critério de parada atingido. Normalmente, este critério de parada é um número máximo de gerações (que é o caso do GAPC).

Os algoritmos genéticos foram usados nesse trabalho principalmente por sua habilidade em processos de buscas rápidas, robustas e adaptativas. Além disso, como uma técnica de computação evolucionária, eles operam em paralelo sobre a população de soluções candidatas, permitindo uma exploração simultânea de diferentes regiões do espaço de busca no domínio da solução.

Dentre os vários bancos de dados de proteínas disponíveis, foi feita a escolha pelo Protein Databank ou simplesmente PDB (Berman *et al.*, 2000) que é gerido pelo Research Collaboratory for Structural Bioinformatics (RCSB), o qual livremente provê informações a respeito do banco de dados a pesquisadores interessados. Para os propósitos deste artigo, foi utilizada a versão 97 do PDB.

Pré-Processamento do PDB

No PDB, os arquivos de proteínas são comprimidos usando-se o formato GZIP e seu uso pelo algoritmo genético motivou sua descompressão e a consecutiva extração de campos relevantes. Uma ferramenta automatizada foi desenvolvida para esse fim.

Um arquivo PDB tem informação dos três níveis das estruturas de proteínas, mas para os propósitos deste artigo, somente a informação da estrutura primária foi necessária: nome, classe e seqüência de aminoácidos.

O sistema permite a extração das informações relevantes e efetua a conversão de aminoácidos não padrão para a letra U (*unknown*), uma vez que esta letra não faz parte da convenção mundial de letras para a representação de aminoácidos.

Antes da carga do arquivo de entrada (criado a partir de milhares de arquivos PDB), é possível configurar a aplicação para descartar aquelas classes cujo número de proteínas está abaixo de um limite mínimo definido pelo usuário. Esta opção foi necessária porque existem muitas classes com pouquíssimas proteínas, o que tende a ser não estatisticamente relevante para efeitos de classificação. Como conseqüência, todos os experimentos discutidos e apresentados nesse artigo descartaram classe com menos de 10 proteínas. Houve, portanto, 183 classes que apresentavam pelo menos 10 proteínas. Todas as classes em conjunto somam 10531 proteínas.

Metodologia

Em um AG, um indivíduo (também chamado do cromossomo) representa um candidato a solução de um dado problema. No caso do GAPC, um indivíduo consiste de uma seqüência de letras, cada uma representando um aminoácido. Conforme descrito na introdução, o objetivo é o de encontrar uma seqüência de aminoácidos (um padrão) que tenha alto poder de classificação, ou seja, um padrão que ocorra em muitas proteínas de

uma classe e em poucas ou nenhuma proteína das classes remanescentes.

Os 20 aminoácidos são representados por 20 letras padrões, acrescidas da letra U (*unknown*) que representa um aminoácido desconhecido.

Fitness

O *fitness* de cada indivíduo é uma medida da qualidade do candidato a solução representada pelo indivíduo e seu cálculo é mostrado a seguir. Primeiro, para cada classe de proteína, calcula-se a freqüência relativa da ocorrência de uma seqüência de aminoácidos do indivíduo. Em seguida, o sistema calcula a média da habilidade de discriminação de classe da seqüência do indivíduo para uma dada classe *i*, onde *i = 1, ..., n*, e *n* é o número de classes. Seja *Disc(i)* a medida da abilidade de uma seqüência na discriminação da classe *i* para todas as outras classes *j*, *j = 1, ..., n*, *j* ≠ *i*. *Disc(i)* é calculado conforme Equação (1).

$$Disc(i) = F_i * \left(1 - \frac{\sum_{j=1}^{n} F_j}{k} \right) \quad (1)$$

Onde:

- *Fi* é a freqüência relativa da seqüência de aminoácidos do indivíduo na *i*-ésima classe;
- *Fj* é a freqüência relativa da seqüência de aminoácidos do indivíduo no *j*-ésima classe, onde *j = 1..n* e *j* ≠ *i*;
- *n* é o número total de classes;
- *k* é o número total de classes que contém pelo menos uma ocorrência da seqüência de aminoácidos.

Uma vez que o valor de *Disc(i)* tenha sido calculado para todas as classes *i*, *i = 1, ..., n*, o indivíduo é associado à classe que apresenta o maior valor de *Disc(i)*. Em outras palavras, a seqüência é considerada um padrão característico das proteínas da classe *i*. Como conseqüência, a existência daquele padrão em uma proteína de classe desconhecida será considerada uma evidência de que a mesma pertence à classe *i*.

A função de *fitness* proposta está normalizada e apresenta um domínio dentro do intervalo [0..1], o que torna fácil sua interpretação, sendo 1 o melhor valor possível e indicando máxima discriminação.

Operadores genéticos

A. Seleção

A seleção é uma operação que ocorre antes da aplicação dos operadores genéticos, não sendo propriamente um deles. O método de seleção adotado pelo GAPC é chamado de seleção por torneio. Este método toma dois indivíduos aleatoriamente da população e escolhe o que apresenta maior *fitness*. Este processo é repetido *P* vezes (com reposição), onde *P* é o tamanho da população. Então os *P* indivíduos podem sofrer as operações genéticas, descritas nas três próximas subseções.

B. Crossover

Utilizando dois indivíduos do conjunto selecionado (subseção anterior), o operador de *crossover* (ou permutação) é aplicado dentro de uma probabilidade pré-definida. No GAPC foi utilizado o *crossover* de um único ponto, de forma que parte de um indivíduo é permutada com parte de outro.

Considerando-se que o comprimento dos cromossomos é variável, o conceito original de ponto de *crossover* foi modificado e adaptado para a representação de indivíduo do GAPC. O ponto de *crossover* é um percentual que define o ponto inicial a partir de onde a permutação será feita. Por exemplo, suponha-se que dois cromossomos com comprimentos 7 e 8 sejam selecionado e que o ponto de *crossover* seja de 60% (um valor gerado aleatoriamente). Assumindo-se que os dois indivíduos geradores sejam conforme Tabela 1.

Tabela 1 – Indivíduos geradores.

Cromossomo A	Cromossomo B
RAYLEGT	HEATRLCW
1234567	12345678

A posição 60% é calculada para os cromossomos. Para o cromossomo A, o ponto de *crossover* será 4 e para o cromossomo B, será 5. Assim, após a realização do *crossover* entre os dois cromossomos, os descendentes terão a estrutura representada na Tabela 2.

Tabela 2 – Indivíduos descendentes.

Descendente A	Descendente B
RAY**RLCW**	HEAT**LEGT**
1234567	12345678

O operador de *crossover* no GAPC é um operador "inteligente" no seguinte sentido: é implementado de tal forma que os descendentes são avaliados imediatamente após sua produção e se seus *fitness* são inferiores aos dos ancestrais, a operação é desfeita. Esse mecanismo torna o processo evolucionário mais rápido, pois todos os descendentes gerados serão sempre melhores que os ancestrais (ou não serão gerados).

C. Mutação

O operador de mutação é utilizado para promover mais exploração do espaço de busca e para evitar perdas irrecuperáveis de material genético que leva à convergência prematura a algum mínimo local.

Em geral, a mutação é implementada pela mudança de valor de alguma posição (um alelo) de um indivíduo, dentro de uma dada probabilidade, chamada probabilidade de mutação.

No GAPC existem quatro tipos de mutação, cada uma com a mesma probabilidade de ocorrência:

a) *adição à esquerda*: um caracter aleatoriamente gerado é adicionado à esquerda da seqüência de aminoácidos.
b) *adição à direita*: um caracter aleatoriamente gerado é adicionado à direita da seqüência de aminoácidos.
c) *mudança aleatóri***a**: os aminoácidos partindo de uma posição aleatória até o final da seqüência são alterados, excetuando-se as primeira e última posições.
d) *remoção*: um aminoácido cuja posição é aleatoriamente gerada é removido da seqüência.

Da mesma forma que o operador de *crossover*, o operador de mutação também é "inteligente" porque também é implementado de forma que sua avaliação ocorre imediatamente após sua geração. Se o *fitness* do descendente não for melhor que o do antepassado, a operação será desfeita.

D. Expansão

O operador de expansão foi especificamente desenvolvido para uso no GAPC. Este operador inicia fazendo acesso a primeira proteína da classe associada ao indivíduo (esta classe foi determinada durante o cálculo do fitness, como explicado na seção 3.1) e localizando a posição da seqüência na mesma. Em seguida o aminoácido imediatamente anterior ao início da seqüência (na proteína) é adicionado à seqüência original e sua freqüência relativa é recalculada. Se esta nova freqüência for maior ou igual à anterior, o aminoácido é efetivamente adicionado à seqüência original, o que corresponde a expandir o genoma do indivíduo. Este processo é repetido até que a freqüência relativa seja inferior ou não existam mais aminoácidos para serem adicionados. Neste ponto, o processo descrito acima passa a ser aplicado para aminoácidos subseqüentes na proteína. Finalmente, todo o processo (expansão para a esquerda e para a direita) é repetido para todas as proteínas da classe do indivíduo. A Tabela 3 mostra um exemplo da aplicação deste operador.

Tabela 3 – Exemplo do operador de expansão.

Proteína		Cromossomo	Comp	*Fitness*
Glycogen Phosphorylase	CO	GNM	3	0,9899
	CR	APGYHMAK MIIKLITAIGD VVNHDPVVG DRLRVIFLEN YRVSLAEKVI PAADLSEQIS TAGTEASGT GNM	70	0,9998

CO: Cromossomo original
CR: Cromossomo resultante após expansão

Designação de resultado

Um AG convencional retorna, como resultado, o melhor indivíduo (o que apresenta maior *fitness*) gerado durante sua execução. No caso do GAPC, no entanto, o resultado desejado não é um único indivíduo, mas um conjunto deles. O motivo é que cada indivíduo representa um único padrão (seqüência de aminoácidos) associado a uma única classe de proteínas. De forma a classificar proteínas, é necessário descobrir muitos padrões associados com o máximo de classes possível.

Descreve-se a seguir a abordagem para a descoberta de padrões. Em cada geração, após o cálculo do *fitness* de cada indivíduo, uma cópia dos padrões de alta qualidade é armazenada em um conjunto auxiliar chamado de conjunto de padrões descobertos. Um padrão é eleito para ser colocado no conjunto de padrões descobertos se seu fitness for maior ou igual a um limiar definido pelo usuário. Nos experimentos realizados, foram utilizados 0,7 e 0,9 como limiares (ressalta-se que o domínio do *fitness* está entre 0 e 1, inclusive).

Esta política resulta na descoberta de muitos padrões, associados a várias classes de

proteínas, como desejado. O procedimento que utiliza o conjunto de padrões descobertos para classificar proteínas desconhecidas é explicado na próxima seção.

Classificação de proteínas desconhecidas

De forma a classificar uma nova e desconhecida proteína, o sistema leva em conta todos os padrões descobertos (seqüências de aminoácidos de indivíduos com alto *fitness*). É importante lembrar que cada padrão descoberto está associado a uma classe e que a ocorrência de um padrão em uma dada proteína pertence à esta classe. Portanto, de forma a classificar uma nova proteína, o sistema faz uso de um modelo de classificação no qual os padrões que caracterizam uma classe são procurados. Aquela classe que contiver o maior número de padrões localizados dentro da proteína desconhecida será escolhida para classificá-la, desde que supere um limiar mínimo de ocorrência de 60%. Este liminar pode ser definido pelo usuário.

Resultados

Experimentos foram executados com o objetivo de medir a taxa de acerto da classificação, usando-se um subconjunto do PDB (como especificado na seção 2). Este subconjunto contém 10531 proteínas, pertencentes a 183 classes.

Como normalmente feito na literatura, a taxa de acerto na classificação é calculada em um conjunto de teste sem intersecção com o conjunto de treinamento (Hand, 1997) de forma a avaliar a generalidade dos padrões identificados. A taxa de acerto da classificação é medida como descrito a seguir. Uma vez que o AG concluiu sua execução, tem-se como resultado um conjunto de padrões que supostamente identificam as proteínas de uma dada família ou classe de proteínas. Neste instante, o sistema escolhe um subconjunto do conjunto de teste contendo apenas proteínas pertences a classes para a qual pelo menos um padrão foi identificado. A taxa de acerto é calculada pelo uso unicamente das proteínas deste conjunto de teste. As proteínas que não pertencem ao subconjunto não podem ser confiavelmente classificadas, pois nenhum padrão foi descoberto para elas. Esta é uma limitação da atual versão deste método, que será abordada em pesquisas futuras. Notar que para uma mesma classe, várias seqüências podem ser encontradas, já que apresentam, individualmente, *fitness* mais altos que o estabelecido pelo limiar mínimo de qualidade, definido pelo usuário.

O acerto preditivo foi medido pelo uso de um bem conhecido procedimento de validação cruzada de dez partições, no qual o conjunto de dados é dividido em 10 partes mutuamente excludentes. Uma das partições é utilizada como conjunto de teste e as outras 9 formam um conjunto de teste. Este processo é repetido até que todas as partições tenham sido usadas como conjunto de teste. A taxa de acerto média sobre as 10 execuções do algoritmo (sobre os dez conjuntos de teste) é a taxa retornada como resultado desta validação cruzada.

A tabelas 4 e 5 mostram os resultados obtidos com o sistema GAPC, os parâmetros usados, as taxas de acerto e o número de classes identificadas, ou seja, o número classes para as quais pelo menos um padrão foi identificado pelo GAPC. Em todos os experimentos, a população foi dimensionada para conter 100 indivíduos.

A tabela 4 mostra a influência do número de gerações sobre a taxa de acerto e sobre o número de classes descobertas. Se o número de gerações sobe, o número de padrões descobertos também sobe, mas o número de diferentes classes identificadas permanece praticamente o mesmo. Os resultados da tabela 2 foram obtidos com um limiar de qualidade mínima de 0,9 (ver seção 3). Em todos os experimentos, a probabilidade de *crossover* foi de 20%.

Tabela 4 – Resultados – 80% de mutação.

Gerações	Taxa de acerto (%)	#Classes
100 generations	94,85 ± 1,13	46
300 generations	93,69 ± 1,11	48

A Tabela 5 mostra a influência da probabilidade de mutação. Se esta probabilidade aumenta, o número de classes identificadas também aumenta, mas a taxa de acerto diminui. Todos os resultados da tabela 3 foram obtidos com o limiar de qualidade mínima de 0,7. É importante lembrar que o operador de mutação nunca reduz o *fitness* de um indivíduo (ver subseção 3.2.3).

Tabela 5 – Resultados – 100 gerações.

Prob. de mutação	Taxa de acerto (%)	#Classes
20%	89,89 ± 1,33	63
50%	88,02 ± 1,92	66
70%	87,82 ± 1,75	68
100%	87,78 ± 1,64	69

O uso do operador de expansão foi muito benéfico no sentido de permitir ao GA a obtenção de seqüências mais longas com *fitness* mais alto que as seqüências mais curtas encontradas em

experimentos anteriores. Inicialmente, sem o uso do operador de expansão, a maioria das seqüências apresentava 3 aminoácidos de comprimento. Com o operador de expansão foi possível localizar seqüências com mais de 200 aminoácidos. Um exemplo está apresentado na figura 1, na qual o sistema encontrou uma seqüência contendo 231 aminoácidos para a classe *Glycogen Phosphorylase*, mostrada na figura 1:

```
AGVENVTELKKNFNRHLHFTLVKDRNVATPRDYY
FALAHTVRDHLVGRWIRTQQHYYEKDPKRIYYLSL
EFYMGRTLQNTMVNLALENACDEATYQLGLDME
ELEEIEEDAGLGNGGLGRLAACFLDSMATLGLAA
YGYGIRYEFGIFNQKICGGWQMEEADDWLRYGN
PWEKARPEFTLPVHFYGRVEHTSQGAKWVDTQV
VLAMPYDTPVPGYRNNVVNTMRLWSAKAP
```

Figura 1 – Um exemplo de um padrão (motif) para a classe *Glycogen Phosphorylase.*

A seqüência acima apresenta um *fitness* muito alto, de 0,9998 (recordando que o mais alto valor para *fitness* é 1).

Todos os resultados apresentados foram obtidos em um computador equipado com um processador Intel© Pentium™ III de 550Mhz e 256Mb de memória principal. Cada execução tomou, em média, 4 horas. O sistema foi desenvolvido usando-se o Borland Delphi™ versão 5 e executado sobre o sistema operacional Microsoft© Windows 2000™ Professional.

Discussão e Conclusões

Foi apresentado um sistema baseado em um Algoritmo Genético modificado, voltado à classificação de proteínas, cuja avaliação se deu sobre um subconjunto das proteínas encontradas no Protein Databank (PDB). O AG proposto tem operadores especialmente projetados para o problema de classificação de proteínas, o que contribuiu para o sucesso no uso do mesmo durante os experimentos. A monitoração das taxas de acertos durante a execução dos experimentos mostrou que as mesmas variam de 87,78% a 94,85%.

Algumas direções para pesquisas futuras são colocadas a seguir. A função de *fitness* utilizada pelo AG do GAPC poderá provavelmente ser melhorada, por exemplo, levando-se em conta o comprimento dos padrões, bem como sua posição relativa dentro da proteína.

Adicionalmente, de forma a elevar a flexibilidade na busca pelos padrões, pode ser interessante investigar o uso do conceito de hiato (do inglês, *gap*), onde os padrões encontrados não precisariam ser exatamente iguais aos presentes nas proteínas, mas apenas muito semelhantes (Nicholas *et al.*, 1998). Este aprimoramento parece ter grande importância, pois algumas famílias não apresentam padrões característicos associados a elas.

Referências

Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E. (2000), "The Protein Data Bank", *Nucleic Acids Res.*, v. 28p. 235-242.

DavisS, L. (1991), *Handbook of Genetic Algorithms*, New York: Van Nostrand Reinhold.

De Jong, K. A. (1975), *An Analysis of the Behavior of a Class of Genetic Adaptative Systems,* Ph.D. Thesis, University of Michigan, Michigan.

De Jong, K. A. (1987), "On Using Genetic Algorithms to Search Program Spaces", *Proc. 2nd Intl Conf. on Genetic Algorithms and Their Applications*, p. 210-216.

De Jong, K. A. (1993), "Genetic Algorithms are not Function Optimizers?", In: *Foundations of Genetic Algorithms 2*, San Mateo: Morgan Kaufmann, p. 5-17.

Goldberg, D.E. (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*, Massachusetts: Addison Wesley.

Greenwood**,** G.W., Lee, B., Shin, J., Fogel, G.B. (1999), "A Survey of Recent Work on Evolutionary Approaches to the Protein Folding Problem", *Proc. Congress of Evolutionary Computation*, p. 488-495.

Holland, J. (1973), "Genetic Algorithms and the Optimal Allocations of Trials", *SIAM J. Sci. Comp.*, v. 2, n. 2, p. 88-105.

Lehninger, A.L., Nelson, D.L., Cox, M.M. (1998), *Principles of Biochemistry*, 2ª ed., New York: Worth Publishers.

Lu, S.Y., Fu, K.S. (1978), "A Sentence-to-Sentence Clustering Procedure for Pattern Analysis", *IEEE Trans. Syst., Man, Cyb.*, v. SMC-8, p. 381-389.

Nicholas, H.B., Deerfield, D.W. and Ropelewski, A.J. (1998), "A Tutorial on Searching Sequence Databases and Sequence Scoring Methods", *Sequence Analysis Tutorials*, Carnegie Mellon University.

Piccolboni, A., Mauri, G. (1998), "Application of Evolutionary Algorithms to Protein Folding Prediction", *Proc. ICONI '97*, New York: Springer Verlag.

Wang, J.T.L., Sasha, D., Wu, C.H. (2000), "Application of neural networks to biological data mining: a case study in protein sequence classification", *Proc. KDD* , p. 305-309.