Scientia
Interdisciplinary Studies in Computer Science
20(2): 83-93, July/December 2009
 – doi: 10.4013/sct.2009.20.2.02

Controle da diversidade da população em algoritmos genéticos aplicados na predição de estruturas de proteínas

Vinicius Tragante do Ó[1], Renato Tinós[2]

[1]Departement Computerwetenschappen, Katholieke Universiteit Leuven
Celestijnenlaan 200 A, 3001 Heverlee, Belgium

[2]Departamento de Física e Matemática, FFCLRP, Universidade de São Paulo
Av. Bandeirantes, 3900, 14040-901, Ribeirão Preto, SP, Brasil

vinicius.tragantedo@cs.kuleuven.be; rtinos@ffclrp.usp.br

Resumo
Apesar de utilizados em diversos problemas de otimização, Algoritmos Genéticos (AGs) tradicionais apresentam dificuldades, quando aplicados ao problema de predição de estruturas terciárias de proteínas. Isto ocorre porque o espaço de soluções é muito grande, e a convergência da população geralmente se manifesta antes que uma porcentagem razoável das soluções seja explorada. Assim sendo, este trabalho investiga o efeito que técnicas de incremento da diversidade da população em Algoritmos Genéticos tem sobre esta aplicação. Algoritmos Genéticos com Hipermutação e Imigrantes Aleatórios, técnicas tradicionais para o controle da diversidade da população, são comparados de acordo com seus resultados na determinação de estruturas das proteínas Crambina (PDB 1CRN), Met-Encefalina (PDB 1PLW) e DNA - Ligante (PDB 1ENH). Os resultados mostram uma significativa redução da energia mínima encontrada, graças ao aumento da diversidade da população, mas que não se reflete, necessariamente, em uma estrutura próxima da estrutura nativa.

PALAVRAS-CHAVE: computação evolutiva, algoritmos genéticos, hipermutação, imigrantes aleatórios, predição de estruturas de proteínas.

Abstract
Control of the population diversity in genetic algorithms applied to the protein structure prediction problem. Genetic Algorithms (GAs), a successful approach for optimization problems, usually fail when employed in the standard configuration in the protein structure prediction problem, since the solution space is very large and the population converges before a reasonable percentage of the possible solutions is explored. Thus, this work investigates the effect of increasing the diversity of the population on this problem by using Hypermutation and Random Immigrants, two traditional population diversity control schemes, in the structure prediction of the proteins Crambin (PDB 1CRN), Met-Enkephalin (PDB 1PLW), and DNA-Ligand (PDB 1ENH). Results show a significant reduction of the minimal energy found, thanks to the diversity, but this does not necessarily means a higher similarity to the original structure.

KEY WORDS: evolutionary computing, genetic algorithms, hypermutation, random immigrants, protein structure prediction.

1 Introdução

Proteínas são cadeias polipeptídicas compostas por quaisquer dos 20 aminoácidos existentes. Na Bioquímica, a estrutura terciária de uma proteína é sua conformação tridimensional, ou seja, tal estrutura indica como os átomos de cada aminoácido estão distribuídos espacialmente (Lehninger *et al.*, 2005). A função que cada proteína exerce está intimamente relacionada à sua estrutura; visto de outra forma, o formato no qual a estrutura se apresenta determina a sua funcionalidade, pois, de acordo com esse formato, podem ocorrer a formação de núcleos hidrofílicos ou hidrofóbicos, as interações de hidrogênio, as cargas elétricas. O resultado dessas características, quando em contato com outras moléculas, pode ativar ou inibir seu funcionamento.

A determinação de estruturas tridimensionais de proteínas está entre os problemas mais importantes da biologia molecular, visto que a ativação ou inibição de moléculas pode ocorrer quando há o perfeito "encaixe" entre um ativador/inibidor e a molécula-alvo. Assim, é possível desenvolver estruturas químicas que atendam especificamente a requisitos funcionais relacionados a um objetivo, seja este a criação de um novo fármaco, seja uma estrutura que dê melhor sabor ou cheiro a um alimento, entre outros, de acordo com uma determinada conformação estrutural.

Atualmente, os métodos experimentais mais eficazes na tarefa de determinação de estruturas de proteínas são a cristalografia e a ressonância magnética nuclear. Estes, apesar dos avanços, ainda apresentam limitações em relação a tamanho da molécula analisada e aos custos de aplicação (Han e Kambert, 2001).

Teoricamente, é possível determinar uma estrutura proteica a partir da sequência primária de aminoácidos que a compõe (Ginalski *et al.*, 2005), se utilizada uma simulação refinada de processos físicos (Anfinsen, 1973) em um processo de dobramento (*folding*) de proteínas. Com a capacidade de processamento atualmente disponível, é possível simular muitas das características presentes nas proteínas em relação a suas características físicas e de ligações químicas, apesar de nem todas as interações serem, atualmente, passíveis de modelagem computacional. Entretanto, a capacidade computacional ainda não é suficiente para que simulações com mecânica quântica, que seria o método mais oportuno, sejam realizadas (Anile *et al.*, 2006).

A predição de proteínas, contudo, pode ser vista como um problema de otimização, no qual, dada uma sequência de aminoácidos, deve-se encontrar a melhor estrutura dentre várias outras possíveis. No entanto, a determinação de estruturas de proteínas é um problema NP-completo (Pierce e Winfree, 2002), ou seja, existe uma explosão combinatória no número de soluções possíveis.

Recentemente, Algoritmos Genéticos (AGs) têm sido aplicados em diversos problemas de otimização reconhecidos como difíceis (e.g., os problemas NP-completo) para técnicas tradicionais (Goldberg, 1989), como, por exemplo, em problemas de otimização em seleção de atributos relevantes (Yang e Honavar, 1998), logística (Taniguchi *et al.*, 1999), sistemas elétricos (Fukuyama *et al.*, 1996), entre outros.

Semelhantemente, para o problema de predição de estruturas de proteínas, também tem havido um crescente interesse na aplicação de AGs (Pedersen e Moult, 1996; Schulze-Kremer, 1993). No entanto, para este problema, os AGs, em sua forma padrão, geralmente, não apresentam resultados satisfatórios, em razão da existência de muitos ótimos locais no espaço de busca e da dificuldade de escolha da função de energia a ser minimizada.

Como exemplo desse último problema, em Schulze-Kremer (1993) foram obtidos resultados com energia potencial menor que a do estado nativo da proteína estudada (Met-Encefalina), porém o estado nativo não foi encontrado.

No trabalho aqui apresentado, o foco é a existência de muitos locais ótimos no espaço de busca do problema de predição da estrutura de proteínas, o que induz o AG a uma convergência prematura. Essa convergência, ao longo das gerações, origina-se na perda de diversidade da população: conforme o algoritmo vai sendo executado, os indivíduos se tornam mais e mais similares, em torno de um ponto ótimo, que, na maioria das vezes, é um ótimo local, sem que o ótimo global, ou seja, a melhor solução para o problema, seja encontrado.

Dadas essas bases, o objetivo principal deste trabalho é investigar se técnicas de manutenção e aumento de diversidade da população em AGs são úteis para este problema. Para o alcance dessa meta, foram empregadas as técnicas de Hipermutação e Imigrantes Aleatórios (Cobb e Grefenstette, 1993; Vavak e Fogarty, 1996).

O artigo se desenvolve da seguinte forma: a Seção 2 apresenta a metodologia empregada neste trabalho; os experimentos são descritos na Seção 3; a Seção 4 evidencia os resultados obtidos para três proteínas distintas para o AG tradicional e para o AG com Hipermutação ou com Imigrantes Aleatórios com diferentes taxas de substituição; por fim, a Seção 5 apresenta as conclusões do trabalho.

2 Metodologia

A abordagem escolhida para este estudo é a de *ab initio* com informações de base de dados. Por este método, os algoritmos não se fundam em uma estrutura pronta; a inicialização é aleatória, mas o uso de bases de dados (neste caso, de ângulos) evita combinações impossíveis entre ângulos. Para concretizar essa abordagem, foram montadas uma base de dados de ângulos de torção φ (phi) e ψ (psi) de cada um dos 20 aminoácidos existentes, a partir do projeto *Conformation Angle Database* (CADB) (Sheik *et al.*, 2003), e outra base de dados para a cadeia lateral dos aminoácidos, que pode possuir de nenhum a 5 ângulos χ, de acordo com as características de cada aminoácido. Esta base foi obtida a partir do trabalho de Tuffery *et al.* (1991).

Os dados são colocados como entrada para a formação de indivíduos em um AG (Mitchell, 1996), construído em sua forma padrão (Linden, 2006), e, depois, alterado em algumas características, a fim de incluir técnicas de controle de diversidade no conjunto de soluções.

O AG, conforme definido originalmente por Holland (1975), é uma estratégia de evolução, na qual uma população é constituída por um número predefinido de

indivíduos, os quais representam, um a um, uma solução em potencial para o problema estudado. Cada indivíduo é composto por um cromossomo com valores que podem representar a solução procurada. Esses cromossomos estão sujeitos à recombinação gênica e mutação, ao longo de gerações, formando novos indivíduos que mesclam características de seus antecedentes. Um processo de seleção inspirado na seleção natural se encarrega de eliminar os indivíduos pior adaptados ao problema e de permitir a sobrevivência daqueles que se adaptam melhor às condições oferecidas, de acordo com uma função de adaptação (*fitness*) ao problema apresentado.

Esse comportamento é muito importante para aplicação em problemas de otimização, nos quais diversos parâmetros devem ser combinados para gerar a melhor solução. Uma combinação de características ideal pode estar espalhada por vários indivíduos, e o objetivo necessário é combinar tais trechos, que são desconhecidos, em um único indivíduo. Com a seleção e recombinação gênica, novos conjuntos de soluções, que combinam partes das soluções anteriores, se formam, levando os indivíduos a uma melhor adaptação ao meio, geralmente, ao redor de um ponto de máximo, que pode ser um máximo local ou o máximo global, que é o objetivo buscado.

Em princípio, o cromossomo pode consistir de valores reais para os ângulos φ, ψ e χ entre –180º e 180º. No entanto, essa estratégia não respeita as restrições do mapa de Ramachandran (Ramachandran e Sasisekharan, 1968) e é insuficiente para se atingirem resultados satisfatórios em relação à redução da energia mínima do sistema, em vista da quantidade muito grande de combinações possíveis que podem ser formadas entre os ângulos de torção de cada aminoácido da proteína. Assim, será muito difícil que um dos indivíduos apresente todas as combinações ideais para todos os aminoácidos em um tempo computacionalmente aceitável. A Figura 1 explica os ângulos citados em relação a um aminoácido qualquer.

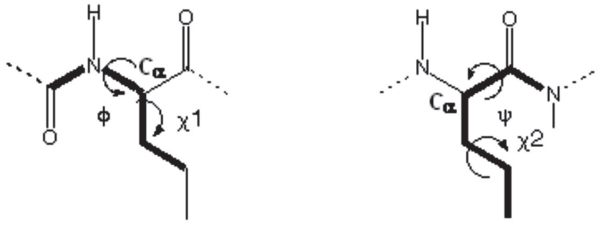

Figura 1. Ângulos que formam os cromossomos neste trabalho. A partir do carbono principal (Cα), os ângulos de torção ϕ e ψ da cadeia principal; e os ângulos da cadeia lateral χi, que podem chegar a 5, dependendo do aminoácido.
Figure 1. Angles that form the chromosomes in this work. From the main carbon (Cα), the torsion angles ϕ and ψ from the backbone chain, and the angles of the side chain χi, ranging up to 5, depending on the amino acid.

Neste trabalho, bases de dados de ângulos de torção são utilizadas para criar um cromossomo do AG. Essas bases de dados possuem combinações de ângulos válidas, pois foram retiradas de proteínas cujas estruturas já foram determinadas experimentalmente por ressonância magnética ou cristalografia.

Para os ângulos de torção da cadeia principal, utilizou-se o projeto CADB 2.0 (Mohan *et al.*, 2005), que foi desenvolvido mediante o uso de dois conjuntos de dados com proteínas com identidade de 25% e 90%, e que armazena cerca de 2,28 milhões de combinações de ângulos de torção da cadeia principal, de mais de 7.000 proteínas. O projeto CADB possui funcionalidades, como a exibição da cadeia principal e lateral para um aminoácido específico e um estudo de inter-relação entre a cadeia principal e a cadeia lateral. Possui limitações, conforme discutido em Dayalan *et al*. (2005), mas estas não se referem ao escopo deste trabalho. Todas as combinações de cada aminoácido foram inseridas em arquivos-texto, em que cada arquivo é relativo a um aminoácido.

Para os ângulos de torção da cadeia lateral, empregou-se o banco de dados de Tuffery *et al.* (1991). O projeto desse pesquisador analisou cadeias laterais de proteínas cujas estruturas já são conhecidas. Tal estudo também efetuou a distribuição de frequências de cada sequência, conforme foram encontradas, gerando duas bases de dados: a dependente da cadeia principal e a base independente da cadeia principal, esta última utilizada para este estudo. Outros trabalhos fazem uso da mesma abordagem (Koehl e Delarue, 1994; Holm e Sander, 1992). Todos os valores dos ângulos aos quais os índices se referem estão mantidos como um vetor dentro do algoritmo, diminuindo o tempo de acesso a estes valores, como demonstrado pela Figura 2. A Figura 3 mostra a relação entre os índices do cromossomo e a base de dados.

Duas abordagens para as bases de dados foram testadas: com os ângulos distribuídos de forma aleatória nas bases e com os ângulos ordenados de –180º a 180º. Esta última estratégia se justifica pelo fato de que, pelo operador de mutação, uma pequena mudança no índice dos ângulos pode significar uma grande mudança nos valores dos ângulos, quando a base não está ordenada. Isso ocorre porque os valores não possuem relação entre si, fazendo com que uma mudança de índice mude os ângulos para valores completamente diferentes, transformando, de forma dramática, a estrutura proteica, e, consequentemente, a energia potencial desta (Tragante e Tinós, 2008).

A ordenação é efetuada pelo ângulo φ, de forma crescente: primeiro vêm os ângulos mais próximos de –180º, depois, os ângulos mais próximos de 180º. Em caso de ângulos φ iguais, a ordenação segue para o ângulo ψ, nos mesmos moldes do anterior. Vale lembrar que nem todos os ângulos φ podem formar pares com os ângulos ψ existentes, e, assim, cada entrada na base de dados representa uma combinação única e válida, sem que estes valores possam ser misturados.

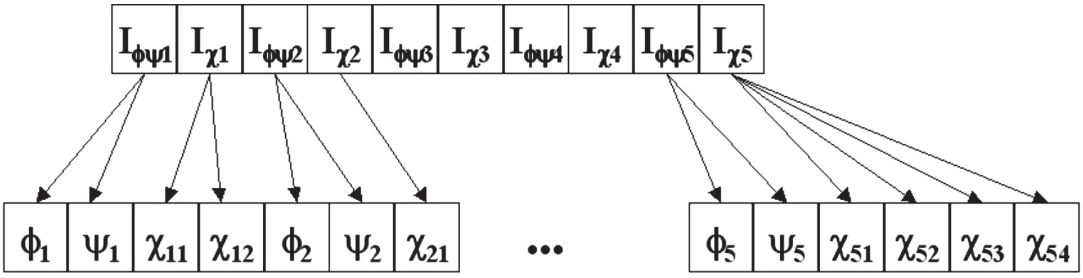

Figura 2. Esquema gráfico de um cromossomo para uma proteína de 5 aminoácidos. Cada aminoácido é representado por dois valores, $I_{\phi\psi}$ e I_{χ}, que são índices da base de dados para a cadeia principal e cadeia lateral, respectivamente. Um vetor auxiliar armazena os valores dos ângulos da base de dados, no índice relacionado.
Figure 2. Graphical schema of a chromosome for a protein 5 amino acids long. Each amino acid is represented by two values, $I_{\phi\psi}$ e I_{χ}, which are indexes of the database for the main chain and side chain, respectively. An auxiliary vector stores the values of the database angles, in the related index.

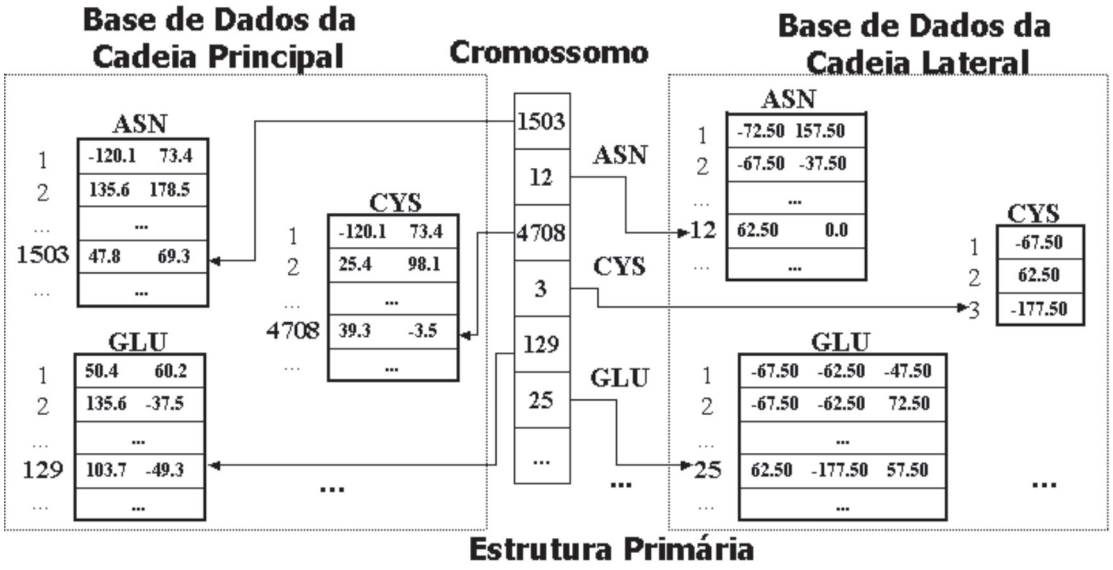

Figura 3. Representação gráfica da relação entre um cromossomo exemplo deste trabalho e a base de dados a que cada índice se liga.
Figure 3. Graphical representation of the relationship between a chromosome from this work and the database to which each index is connected.

No tocante ao operador de mutação do AG, a probabilidade de ocorrência é de $1/(2m)$, em que m é o número de aminoácidos da proteína, com a taxa de *crossover* fixada em 0,8. Na ocorrência de mutação, o índice relativo ao ângulo utilizado é alterado em uma posição, ou seja, +1 ou –1, com igual probabilidade em relação ao índice do ângulo do pai de origem na base de ângulos. Quando a mutação ocorre, os ângulos são alterados para um dado aminoácido (por exemplo, os dois ângulos de torção da cadeia principal ou os n ângulos da cadeia lateral), respeitando a regra de combinações de ângulos válidas. Dessa forma, quando as bases de dados não estão ordenadas, uma mutação de um número no índice leva a uma combinação de ângulos de torção completamente diferente da anterior, pois não há relação entre um par de ângulos e seus vizinhos. Empregando as bases ordenadas, no entanto, o índice vizinho apresentará uma combinação de ângulos próximos aos anteriores, o que pode significar uma pequena melhora sobre um ângulo que já é bom, e, assim, efetuar uma escalada no *fitness* do indivíduo.

O operador de *crossover* mescla as características de dois indivíduos selecionados por torneio. Por esse método, dois indivíduos são escolhidos aleatoriamente, e o indivíduo com melhor *fitness* possui 75% de chances de ser escolhido para efetuar o *crossover.* Por conseguinte, o indivíduo de pior *fitness* entre os dois possui os 25% restantes de possibilidade de ser selecionado (Mitchell, 1996). O método é o *crossover* de um ponto, no qual um ponto aleatório é selecionado, a parte do cromossomo à esquerda deste ponto é retirada de um pai, e o restante é obtido a partir do cromossomo do outro pai. As partes restantes dos cromossomos são aplicadas para a obtenção do segundo filho.

Um processo de elitismo garante que os dois melhores indivíduos de uma geração sejam automaticamente transportados à geração seguinte, sem passar por *crossover* ou mutação, de forma que o avanço atingido até a dada geração não seja perdido.

Após o término do processo de geração de novos indivíduos, o AG cria arquivos com as informações de ângulos de cada indivíduo e os envia ao algoritmo *protein* do pacote de modelagem molecular Tinker (Ponder, 1998), o qual converte estes ângulos de torção em coordenadas espaciais e no formato do *Protein DataBase* (http://www.rcsb.org/). Na sequência, o algoritmo *analyze,* do mesmo pacote, faz a avaliação dos indivíduos, com base nas interações presentes na proteína e retorna a energia, que será o *fitness* deste indivíduo. O AG tenta reduzir esta energia para o menor valor possível.

A avaliação da energia do pacote Tinker depende do campo de força escolhido. Neste trabalho, empregou-se o campo de força CHARMM27. Esta função de energia é composta pelos seguintes parâmetros:

$$E_{tot} = E_{bond} + E_{ang} + E_{tors} + E_{urey} + E_{impr} + E_{VdW} + E_{ch} \quad (1),$$

em que:

E_{tot} = Energia potencial total (usado aqui como *fitness*);

E_{bond} = Energia do comprimento de ligação (*bond stretching*), que varia conforme a distância da ligação;

E_{ang} = Energia de ângulo de ligação (*angle bending*), que varia conforme a abertura do ângulo;

E_{tors} = Energia de ângulo de torção (*torsion angle*);

E_{urey} = Energia Urey-Bradley, que se refere à interação entre átomos não diretamente vizinhos;

E_{impr} = Energia imprópria (*improper torsion*), associada a deformações nos ângulos de torção;

E_{VdW} = Energia Van der Waals, referente à perturbação não covalente;

E_{ch} = Energia Eletrostática (*charge-charge*), denotada pelo potencial de Coulomb.

Os parâmetros padrão deste campo de força foram mantidos, exceto pela constante dielétrica, que foi alterada para 78,7 para simular o efeito da água no ambiente e dar maior fidelidade à simulação. Em seguida, os algoritmos de Hipermutação e Imigrantes Aleatórios foram codificados, independentemente, com o objetivo de manter ou aumentar a diversidade da população do AG durante as gerações.

Hipermutação (Cobb e Grefenstette, 1993) é uma estratégia de incremento da probabilidade de mutação, de acordo com um critério específico. Um exemplo deste tipo de critério é a baixa diversidade da população, o que pode ser avaliado pela média do *fitness* da população. Caso a média esteja próxima do valor do melhor indivíduo, pode-se concluir que todos os indivíduos estão semelhantes. Outra abordagem empregada neste trabalho é o aumento da taxa de mutação, periodicamente, durante um dado número de gerações, com retorno ao normal, pelo mesmo número de gerações, em um ciclo contínuo. O valor escolhido empiricamente foi de 5 gerações de incremento na probabilidade de mutação e 5 gerações sob probabilidade padrão. O pseudocódigo apresentado no Algoritmo 1 descreve esta abordagem.

Nos Imigrantes Aleatórios (Cobb e Grefenstette, 1993), uma porcentagem dos indivíduos da população é substituída por novos indivíduos criados aleatoriamente, a cada geração. Os indivíduos a serem substituídos são escolhidos de forma aleatória.

Algoritmo 1:

```
                Hipermutaçao
procedimento geracao ( )
  inicio
para(contador=0;contador<tamanho_populacao;contador+2)    //2 filhos
      pai_1 = torneio( )                  //seleção do pai 1
    pai_2 = torneio( )                    //seleção do pai 2
    filho = pai_1.crossover(pai_2)        //crossover
    se (flag == 0)
      taxa_mutacao = taxa_normal  //taxa_normal=(1/2m), onde 2m é o
   //tamanho do cromossomo
      contador++                          // conta gerações com hipermutação
      se (contador = 5)
        flag = 1
        contador = 0
        pare
      fim_se
    senao
      taxa_mutacao = taxa_alta      //taxa_alta=80% de probabilidade
      contador++
      se (contador = 5)                   //5 gerações com mutação alta
        flag = 0
        contador =0
      fim_se
    fim_se
    filho[0].mutacao(taxa_mutacao)      //envia o filho 1 para ser mutado
    filho[1].mutacao(taxa_mutacao)      //envia o filho 2 para ser mutado
 fim_para
 fim.
```

Neste trabalho, os novos indivíduos aleatórios são gerados em uma geração t e automaticamente inseridos na geração t+1, sem passar por avaliação de *fitness*. A avaliação desses indivíduos só é efetuada na geração t+1, e os indivíduos da geração t, assim, revelam uma chance menor de serem escolhidos para *crossover*, pois parte da população já foi preenchida. O pseudocódigo que descreve o procedimento de criação dos indivíduos no AG com Imigrantes Aleatórios é apresentado no Algoritmo 2.

Além disso, testou-se a possibilidade de proceder à inserção de novos indivíduos após um número de gerações predefinido, de modo que o procedimento de *crossover* permitisse a convergência da população, e, só então, novas características fossem inseridas. Este procedimento pode ser particularmente proveitoso para manter a diversidade

da população, uma vez que, em todas as gerações, há a inserção de novos indivíduos, os quais, possivelmente, carregam em si ângulos que nunca foram usados na execução do algoritmo, e que, quando combinados a indivíduos já existentes, podem gerar bons descendentes.

Algoritmo 2:

```
                Imigrantes Aleatórios
procedimento geracao ( )
  inicio
    total_imigrantes = 0
    enquanto (total_imigrantes < taxa_subst) //taxa_subst=numero de novos
      filho = novo_individuo( )                    //cria novo individuo aleatório
      nova_pop.adicionar(filho)                    //inclui novo à nova população
      total_imigrantes++
    fim_enquanto
    para(contador=taxa_subst;contador<tamanho_populacao;contador+2)//2 filhos
     pai_1 = torneio( )                          //seleção do pai 1
   pai_2 = torneio( )                           //seleção do pai 2
   filho=pai_1.crossover(pai_2)                 //crossover
   filho[0].mutacao(taxa_mutacao)               //envia o filho 1 para ser mutado
   filho[1].mutacao(taxa_mutacao)               //envia o filho 2 para ser mutado
 fim_para
  fim.
```

3 Experimentos

Nos experimentos realizados, os indivíduos da população inicial são criados aleatoriamente, com distribuição uniforme de posições nas bases de ângulos para cada aminoácido da proteína e para cada indivíduo.

Para todos os algoritmos e proteínas, 100 indivíduos por geração são empregados, e este valor não muda, pois, no caso de inserção de imigrantes aleatórios, estes substituem os indivíduos anteriores. Para as proteínas Crambina e DNA - Ligante, 500 gerações de 100 indivíduos foram empregadas em cada execução do algoritmo. Testes com números maiores de gerações mostraram resultados similares para a comparação dos algoritmos. Para a proteína Met-Encefalina, 50 gerações de 100 indivíduos foram aplicadas, dado o menor tamanho do polipeptídio. Cada algoritmo foi executado dez vezes, com uma semente aleatória diferente por execução, sendo as sementes aleatórias iguais para todos os algoritmos, de forma que a primeira população fosse sempre igual para a mesma execução dos três algoritmos.

Todos os algoritmos usados foram testados com as bases de ângulos ordenadas e desordenadas. No caso dos Imigrantes Aleatórios, três taxas de substituição de indivíduos foram testadas: 2%, 6% e 10% dos indivíduos por geração.

Além disso, testou-se a inserção de novos imigrantes a partir de 10% das gerações transcorridas, ou seja, a partir da 6ª geração, para a Met-Encefalina e, a partir da 51ª geração, para as demais. A taxa de inserção de novos indivíduos, neste algoritmo, foi de 10%.

Os algoritmos descritos acima foram aplicados sobre três proteínas, escolhidas por suas características e por serem amplamente empregadas na literatura, como se vê a seguir.

3.1. Crambina (1CRN)

Proteínas possuem, em média, cerca de 350 resíduos. A Crambina, no entanto, possui apenas 46 aminoácidos. Ela é encontrada nas sementes do repolho abissínio, e sua função biológica é desconhecida, apesar de se saber que ela não está relacionada a nenhuma doença humana. Possui duas alfa-hélices e duas folhas-beta antiparalelas. Além disso, a Crambina possui seis resíduos de Cisteína (cerca de 13% da estrutura), o que é incomum, quando se compara a outras proteínas. É muito utilizada tanto teórica quanto experimentalmente, pois os cristais de Crambina possuem uma difração muito boa, pois apresenta a estrutura de melhor resolução já determinada até hoje, a 0,54 Å (Protein Data Bank Japan, 2008). Devido a este fato, é uma proteína útil para efetuar testes e *benchmarking*, e, por essa razão, foi utilizada por diversos trabalhos, como os de Schulze-Kremer e Tiedemann (1994), de Pedersen e Moult (1996) e Lima (2006).

Sua energia potencial total, quando analisada pelo pacote Tinker, utilizando o campo CHARMM, é de 465,538 kcal/mol.

3.2. Met-Encefalina (1PLW)

A Met-Encefalina é um neurotransmissor narcótico, dotado de atividade analgésica semelhante à da morfina. Ela se fixa nos receptores de certas células nervosas pela extremidade da sua cadeia tirosina N-terminal, cuja conformação é semelhante à dos opiáceos (MDP, 2008). Pode terminar sua cadeia com uma Metionina ou uma Leucina. Por sua reduzida estrutura, de apenas 5 aminoácidos, é muito útil como prova de funcionamento de algoritmos; por isso é empregada em muitos trabalhos, entre eles, os realizados por Kaiser *et al.* (1997), Bindewald *et al.* (1998) e Nicosia e Stracquadanio (2008).

Esta estrutura apresenta uma energia potencial total de 345,978 kcal/mol, segundo o pacote Tinker e empregado o campo CHARMM.

3.3. DNA Ligante (1ENH)

A proteína DNA Ligante (1ENH) representa o homeodomínio granulado da Drosophila e integra uma importante família de proteínas ligantes ao DNA (Clarke *et al*., 1994). Sua principal característica é ser formada por 3 alfa-hélices e 55 aminoácidos, o que constitui um bom representante do domínio α e um bom estudo de caso, empregado também em Lima (2006).

O pacote Tinker, sob o campo de força CHARMM, apresentou uma energia potencial total de 427,305 kcal/mol para esta proteína.

4 Resultados

As tabelas a seguir apresentam o *fitness* do melhor indivíduo, obtido ao final das gerações entre todas as execuções. Assim sendo, nenhum outro indivíduo será considerado, apenas o melhor entre as 10 execuções. A média e o desvio-padrão entre os valores dos melhores indivíduos obtidos em cada execução são também apresentados. O *fitness* original da proteína é apresentado como termo de comparação.

4.1. Crambina

A Tabela 1 apresenta os resultados para a proteína Crambina. É possível notar que tanto a Hipermutação quanto os Imigrantes Aleatórios atingiram melhores resultados que o AG padrão. Esse fato pode ser comprovado por um teste T de *student*. Comparando estatisticamente os resultados da Hipermutação contra o AG padrão, temos um valor de p=0,019 para a base ordenada, e p=0,11 para a base desordenada. Em relação aos Imigrantes Aleatórios, todas as taxas de substituição apresentam melhora estatisticamente significante contra o AG padrão, com valores p={10^{-3}; 10^{-6};10^{-3};10^{-3}}, respectivamente, para as taxas de substituição de 2%, 6%, 10% e 10% de substituição a partir de 10% das gerações já transcorridas, sempre com as bases ordenadas.

4.2. Met-Encefalina

A Tabela 2 apresenta os resultados para a proteína Met-Encefalina. Este caso, particularmente, é interessante, pois todos os algoritmos, inclusive o AG padrão, atingem níveis de energia menores que o estado nativo da proteína.

Pelos resultados, todos os novos algoritmos testados atingiram valores de energia menores que o AG padrão, com significância estatística. Comparando Hipermutação com o AG padrão por um teste T, obtemos p=0,15 e p=0,14 para a base ordenada e desordenada, respectivamente. Para os Imigrantes Aleatórios, os p-valores são p={0,33;0,17;0,04;0,12}, respectivamente,

Tabela 1. Resultados para a proteína Crambina após 500 gerações.
Table 1. Results for protein Crambin after 500 generations.

Algoritmo	Melhor *Fitness*	*Fitness* Médio	Desvio-padrão
Hipermutação (ord.)	586,178	716,465	88,462
Hipermutação (desord.)	581,893	672,237	87,112
Imig. Aleat. 2% (ord.)	561,596	574,746	11,978
Imig. Aleat. 2% (desord.)	559,022	590,557	30,941
Imig. Aleat. 6% (ord)	**506,252**	**525,767**	16,054
Imig. Aleat. 6%(desord)	517,040	538,819	11,936
Imig. Aleat. 10% (ord)	519,987	538,219	18,476
Imig. Aleat. 10% inseridos depois (ord)	507,320	552,867	23,347
AG padrão (ord)	695,754	831,733	110,018
AG padrão (desord)	626,908	816,237	247,777
Estrutura original	465,538		-

Tabela 2. Resultados para a proteína Met-Encefalina após 50 gerações.
Table 2. Results for protein Met-Enkephalin after 50 generations.

Algoritmo	Melhor *Fitness*	*Fitness* Médio	Desvio-padrão
Hipermutação (ord)	43,736	46,237	1,50
Hipermutação (desord)	44,492	46,797	1,078
Imig. Aleat. 2% (ord)	43,420	46,577	1,284
Imig. Aleat. 2% (desord)	44,602	46,618	0,899
Imig. Aleat. 6% (ord)	44,86	46,439	0,979
Imig. Aleat. 6% (desord)	**43,404**	**45,737**	1,246
Imig. Aleat. 10% (ord)	44,847	46,160	0,848
Imig. Aleat. 10% inseridos depois (ord)	43,746	46,252	1,230
AG padrão (ord)	45,599	47,107	1,092
AG padrão (desord)	46,203	47,598	1,223
Estrutura original	345,978		-

para as taxas de substituição de 2%, 6%, 10% e 10% de substituição, a partir de 10% das gerações já transcorridas, sempre com as bases ordenadas. Esta proteína, devido a seu tamanho reduzido, facilita a produção de melhores resultados por todos os algoritmos; dessa forma, o avanço das novas abordagens não é tão significativo, ainda que existente.

4.3. DNA - Ligante

Esta proteína é a de maior custo computacional, por seu tamanho. A Tabela 3 apresenta seus resultados.

Novamente, todos os algoritmos foram capazes de atingir valores de energia menores que o AG padrão, exceto pela Hipermutação com a base ordenada. Os p-valores para os Imigrantes Aleatórios com taxa de reposição de 2%, 6%, 10% e a partir de 10% de gerações transcorridas foram p={0,31;0,01;10^{-5};0,01}. Assim sendo, em praticamente todos os casos, as novas abordagens foram superiores ao AG padrão, e o algoritmo de Imigrantes Aleatórios com 6% de reposição de indivíduos foi o de melhor resultado para as três proteínas, em metade dos casos com a base ordenada e na outra metade com a base desordenada.

4.4. Cálculo de distância de *Root Mean Square Deviation* (RMSD)

Nesta seção, é feita a comparação da estrutura final obtida pelo AG com a estrutura nativa. Isto pode ser obtido de diversas formas; a forma escolhida foi o cálculo de RMSD, que leva em consideração a posição de cada átomo da proteína obtida em relação às posições em coordenadas espaciais da proteína original. A equação é dada por

$$RMSD = \sqrt{\frac{\sum_i d_1^2}{n}} \quad (2),$$

na qual n é o número de átomos e d_i é a distância entre dois átomos i correspondentes das duas estruturas, predita e real (Verli, 2008). Este cálculo é realizado pelo software VMD (Humphrey *et al.*, 1996), desenvolvido pela Universidade de Illinois e disponível para download gratuito[1].

Este método não é empregado diretamente como *fitness* dos indivíduos porque o objetivo final do algoritmo é ser capaz de prever estruturas de proteínas ainda não conhecidas, quando não haveria comparação a ser efetuada; no entanto, como os testes são feitos com proteínas já conhecidas, este cálculo pode ser empregado ao final da execução do AG, de maneira a definir a proximidade da estrutura predita em relação à estrutura original. A Tabela 4 apresenta o melhor e o pior RMSD obtido para cada proteína.

Tabela 4. Melhor e pior RMSD obtido pelo melhor indivíduo de cada algoritmo.
Table 4. Best and worst RMSD obtained by the best individual of each algorithm.

Proteína	Melhor RMSD	Pior RMSD
Crambina (1CRN)	17,512 Å (Imig. Aleat. 6% desord.)	25,667 Å (Imig. Aleat. Ap. 10% ord.)
Met-Encefalina (1PLW)	6,008 Å (Imig. Aleat. 10% ord.)	9,824 Å (Imig. Aleat. 6% desord.)
DNA-Ligante (1ENH)	30,410 Å (Imig. Aleat. 2% desord.)	59,990 Å (Imig. Aleat. 6% ord.)

Tabela 3. Resultados para a proteína DNA–ligante após 500 gerações.
Table 3. Results for the protein DNA–ligand after 500 generations.

Algoritmo	Melhor *Fitness*	*Fitness* Médio	Desvio-padrão
Hipermutação (ord)	1018,911	4920,488	4226,027
Hipermutação (desord)	1053,500	2073,168	986,010
Imig. Aleat. 2% (ord)	795,085	1047,238	183,626
Imig. Aleat. 2% (desord)	704,036	1196,289	458,129
Imig. Aleat. 6% (ord)	691,593	**713,582**	12,922
Imig. Aleat. 6% (desord)	**673,558**	728,646	60,821
Imig. Aleat. 10% (ord)	746,979	868,154	101,005
Imig. Aleat. 10% inseridos depois (ord)	749,063	839,056	81,739
AG padrão (ord)	1446,176	3721,321	2794,986
AG padrão (desord)	1077,668	4290,645	5047,524
Estrutura original	427,305		-

[1]Download disponível em http://www.ks.uiuc.edu/Development/Download/download.cgi?PackageName=VMD.

Por estes resultados, duas análises importantes podem ser efetuadas: (a) a primeira é a de que não há, necessariamente, aparente relação entre a menor energia obtida e o menor RMSD obtido, visto que, em dois dos três casos, o algoritmo de melhor energia conseguiu o resultado estrutural mais distante da proteína original; (b) a segunda é que as populações convergem para mínimos locais (ou globais), cujas estruturas correspondentes divergem da estrutura nativa.

4.5. Manutenção de Diversidade

No tocante à manutenção da diversidade, com o passar das gerações, os indivíduos do AG tendem a ficar cada vez mais parecidos, devido aos operadores de seleção e aos sucessivos *crossovers*, já que as taxas normais de mutação não são capazes de aumentar suficientemente a diversidade da população. Assim, quanto mais gerações transcorridas, mais a média dos indivíduos está próxima do valor do melhor indivíduo. Este efeito pode ser visto na Figura 4.

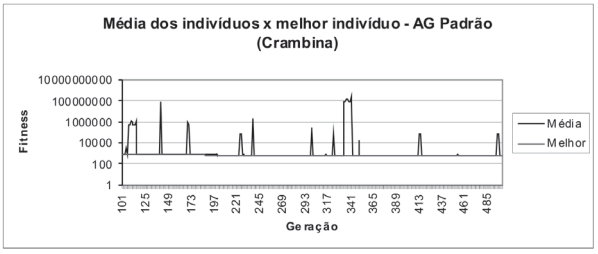

Figura 4. Média dos indivíduos por geração *versus* melhor indivíduo da geração para a sétima execução do AG padrão (a partir da 101ª geração). Devido à falta de diversidade, durante a maioria das gerações, a média da população é muito próxima ao valor do melhor indivíduo.
Figure 4. Average fitness of the individuals per generation versus best individual of the generation for the seventh run of the standard GA (from the 101st generation). Due to the lack of diversity in most generations the average fitness of the population is very close to the value of best individual.

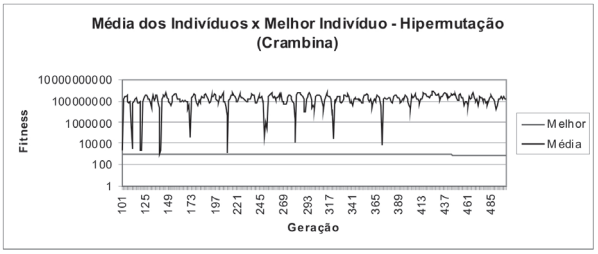

Figura 5. Média dos indivíduos por geração *versus* melhor indivíduo da geração para a terceira execução do AG com Hipermutação.
Figure 5. Average individual fitness per generation versus best individual of generation to the third run of the GA with hypermutation.

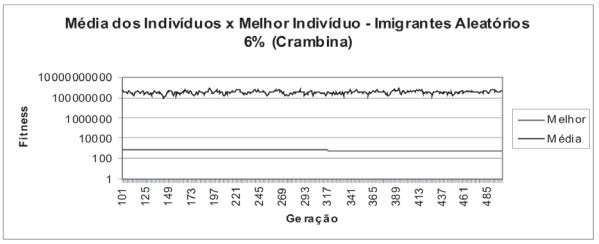

Figura 6. Média dos indivíduos por geração *versus* melhor indivíduo da geração para a segunda execução do AG com Imigrantes Aleatórios com 6% de substituição.
Figure 6. Average individuals per generation versus best individual of generation for the second run of the GA with Random Immigrants with 6% of substitution.

As Figuras 5 e 6 mostram a comparação entre o melhor indivíduo e a média da população para os algoritmos que aumentam a diversidade na população durante a sua execução. Nota-se que, para a Hipermutação, o aumento periódico da taxa de mutação leva à criação de descendentes bastante diferentes dos indivíduos originais. No caso dos Imigrantes Aleatórios, a introdução de indivíduos aleatórios, que geralmente têm valor de *fitness* ruim, faz com que a média do *fitness* da população seja bastante diversa do *fitness* do melhor indivíduo.

Em ambos os casos, nota-se que o *fitness* médio da população e o *fitness* do melhor indivíduo não possuem valores próximos, o que indica que a diversidade da população é muito maior do que para o AG padrão. Observa-se que a média do *fitness* da população aumenta devido à criação de indivíduos menos aptos, mas estes trazem consigo informação diversa da carregada pela população atual. Esta informação, ao ser mesclada com a informação dos melhores indivíduos, eventualmente, permite uma melhora no melhor indivíduo.

5 Conclusões

Este trabalho investigou o uso de duas técnicas de manutenção e o aumento da diversidade de populações em Algoritmos Genéticos para o problema de Predição de Estruturas de Proteínas. As técnicas analisadas foram a Hipermutação e os Imigrantes Aleatórios, escolhidas num universo de outras técnicas de controle de diversidade que podem ser empregadas.

De fato, técnicas que permitam que um maior espaço do conjunto de soluções seja explorado são bem-vindas, principalmente em problemas como o de predição de estruturas proteicas. Esta possui um conjunto NP-completo de soluções, mas apenas uma parte deste conjunto é útil, e uma região ainda menor é a que representa a melhor solução, fazendo com que, facilmente, a população do AG fique presa em um dos muitos ótimos locais existentes,

característicos deste problema.

Para as técnicas investigadas, percebe-se a importância de uma correta escolha da taxa de substituição de indivíduos no algoritmo de Imigrantes Aleatórios, pois, como constatado, taxas muito baixas ou muito altas se mostram ineficazes. Com uma taxa muito baixa de substituição, há uma chance menor de novas características serem aproveitadas, pois há o risco de os novos indivíduos não serem escolhidos para o *crossover* (lembrando que nem todos os indivíduos são escolhidos para *crossover*, uma vez que novos indivíduos ocupam alguns lugares). Além disso, a geração de muitos indivíduos novos não permite a fixação, na população, das informações dos melhores indivíduos.

Para os experimentos apresentados aqui, foram empregadas proteínas já largamente utilizadas na literatura científica, por possuírem características que proporcionam desafios diversos aos algoritmos. Em geral, os resultados são satisfatórios em relação à minimização da energia. No entanto, a estrutura final das proteínas ainda não se apresenta em um patamar satisfatório, o que indica que o conhecimento apresentado para o AG não parece suficiente para que a estrutura completa seja determinada para o número de gerações máximo considerado. Assim, propõe-se a investigação de técnicas que adicionem conhecimento ao AG, para que ele possa atingir melhores resultados a partir de alguma informação prévia.

Agradecimentos

Os autores agradecem o apoio financeiro da FAPESP (projeto 04/04289-6) e CAPES, por meio do Programa de Pós-Graduação em Física Aplicada à Medicina e Biologia da FFCLRP/USP.

Referências

ANFINSEN, C.B. 1973. Principles that govern the folding of protein chains. *Science*, **181**:223-230.

ANILE, A.M.; CUTELLO, V.; NARZISI, G.; NICOSIA, G.; SPINELLA, S. 2006. Lipschitzian pattern search and immunological algorithm with Quasi-Newton Method for the protein folding problem: An innovative multistage approach. *Lecture Notes in Computer Science,* **3931**:307-323.

BINDEWALD, E.; HESSER, J.; MANNER, R. 1998. Implementing genetic algorithms with sterical constrains for protein structure prediction. *In*: INTERNATIONAL CONFERENCE ON PARALLEL PROBLEM SOLVING FROM NATURE (PPSN V), Amsterdam, Netherlands, 1998. *Anais*... Amsterdam, p. 959-967.

CLARKE, N.D.; KISSINGER, C.R.; DESJARLAIS, J.; GILLILAND, G.L.; PABO, C.O. 1994. Structural studies of the engrailed homeodomain. *Protein Science,* **3:**1779-1787.

COBB, H.G.; GREFENSTETTE, J.J. 1993. Genetic algorithms for tracking changing environments. *In*: INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS*,* 5, Urbana-Champaign, IL, 1993. *Anais*... Urbana-Champaign, IL, p. 523-530.

DAYALAN, S.; BEVINAKOPPA, S.; SCHRODER, H. 2005. Homology based structure extractor for protein structure prediction. *International Journal of Lateral Computing,* **2**(1):56-61.

FUKUYAMA, Y.; CHIANG, H.; MIU, K. 1996. Parallel genetic algorithm for service restoration in electric power distribution systems. *International Journal of Electrical Power and Energy Systems*, **18**(2):111-119.

GINALSKI, K.; GRISHIN, N.V.; GODZIK, A.; RYCHLEWSKI, W. 2005. Practical lessons from protein structure prediction. *Nucleic Acids Research*, **33**:1874-1891.

GOLDBERG, D.E. 1989. *Genetic algorithms in search, optimization and machine learning*. Boston, Addison-Wesley Longman Publishing Co. Inc., 432 p.

HAN, J.; KAMBERT, M. 2001. *Data mining: Concepts and techniques*. San Francisco, Morgan Kaufmann, 550 p.

HOLLAND, J.H. 1975. *Adaptation in natural and artificial systems*. Ann Arbor, University of Michigan Press, 228 p.

HOLM, L.; SANDER, C. 1992. Fast and simple Monte Carlo algorithm for side-chain optimization in proteins: Application to model building by homology. *Proteins: Structure, Function and Genetics*, **14**:213-223.

HUMPHREY, W.; DALKE, A.; SCHULTEN, K. 1996. VMD - Visual Molecular Dynamics. *Journal of Molecular Graphics*, 14:33-38.

KAISER JR, C.E.; LAMONT, G.B.; MERKLE, L.D.; GATES JR, G.H.; PATCHER, R. 1997. Polypeptide structure prediction: Real-valued versus binary hybrid genetic algorithms. *In*: ACM SYMPOSIUM ON APPLIED COMPUTING (SAC), San Jose, CA, 1997, *Anais*... San Jose, CA, p. 279-286.

KOEHL, P.; DELARUE, M. 1994. Application of a self-consistent mean field theory to predict protein side-chains conformation and estimate their conformational entropy. *Journal of Molecular Biology*, **239**:249-275.

LEHNINGER, A.L.; NELSON, D.L.; COX, M.M. 2005. *Principles of Biochemistry.* 4ª ed., New York, Freeman, 1100 p.

LIMA, T. 2006. *Algoritmos Evolutivos para Predição de Estruturas de Proteínas*. São Carlos, SP. Dissertação de Mestrado. Universidade de São Paulo, 124 p.

LINDEN, R. 2006. *Algoritmos Genéticos*. 2ª ed., Rio de Janeiro, Ed. Brasport, 428 p.

MÉDICOS DE PORTUGAL (MDP). 2008. *Glossário*. Disponível em: http://medicosdeportugal.saude.sapo.pt/action/10/glo_id/4531/menu/2/. Acesso em: 20/12/2008.

MITCHELL, M. 1996. *An Introduction to Genetic Algorithms*. Cambridge, MIT Press, 221 p.

MOHAN, S.; SHEIK, S.S.; RAMESH, J.; BALAMURUGAN, B.; JEYASIMHAN, M.; MAYILARASI, C.; SEKAR, K. 2005. CADB-2.0: Conformation Angles Database. *Biological Crystallography*, **D61**:637-639.

NICOSIA, G.; STRACQUADANIO, G. 2008. Generalized pattern search algorithm for peptide structure prediction. *Biophysical Journal,* **95**(10):4988-4999.

PEDERSEN, J.; MOULT, J. 1996. Genetic algorithms for protein structure prediction. *Current Opinion in Structural Biology*, **6**(2):227-231.

PIERCE, N.A.; WINFREE, E. 2002. Protein Design is NP-hard. *Protein Engineering*, **15**(10):779-782.

PONDER, J. 1998. *TINKER: Software tools for molecular design*. St. Louis, Department of Biochemistry and Molecular Biophysics, Washington University School of Medicine, versão 4.2.

PROTEIN DATA BANK JAPAN (PDBJ). 2008. *Encyclopedia of Protein Structures*. Disponível em: http://eprots.protein.osaka-u.ac.jp/eProtS/Chain.do?from=group&lang=en&pdb_id=1CRN. Acesso em: 20/12/2008.

RAMACHANDRAN, G.N.; SASISEKHARAN, V. 1968. Conformation of polypeptides and proteins. *Advances in Protein Chemistry*, **23**:283-438.

SCHULZE-KREMER, S.; TIEDEMANN, U. 1994. Parameterizing genetic algorithms for protein folding simulation. *System Sciences*, **5**:345:354.

SCHULZE-KREMER, S. 1993. Genetic algorithms for protein tertiary structure prediction. *Lecture Notes in Computer Science: Machine Learning*, **ECML-93**:262-279.

SHEIK, S.S.; ANANTHALAKSHMI, P.; BHARGAVI, G.R.; SEKAR, K. 2003. CADB: Conformation Angles DataBase of proteins. *Nucleic Acids Research*, **31**(1):448-451.

TANIGUCHI, E.; NORITAKE, M.; YAMADA, T.; IZUMITANI, T. 1999. Optimal size and location planning of public logistics terminals. *Transportation Research Part E*, **35**(3):207-222.

TRAGANTE, V.; TINÓS, R. 2008. Impact of database sorting on the efficiency of genetic algorithms in protein structure prediction. *In*: BIOMAT INTERNATIONAL SYMPOSIUM ON MATHEMATICAL AND COMPUTATIONAL BIOLOGY, VIII, Campos do Jordão, 2008. *Anais*... Campos do Jordão, 12 p.

TUFFERY, P.; ETCHEBEST, C.; HAZOUT, S.; LAVERY, R. 1991. A new approach to the rapid determination of protein side chain conformations. *Journal of Biomolecular Structure Dynamics*, **8**(6):1267-1289.

VAVAK, F.; FOGARTY, T.C. 1996. A comparative study of steady state and generational genetic algorithms for use in nonstationary environments. *Lecture Notes in Computer Science*, 1143:297-304.

VERLI, H. 2008. *Bioinformática Estrutural*. Slides de aula. Disponível em: http://www.cbiot.ufrgs.br/bioinfo/SAEF_03.pdf. Acesso em: 19/01/2009.

YANG, J.; HONAVAR, V. 1998. Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems,* **13**(2):44-49.

Submitted on November 9, 2009.
Accepted on December 16, 2009.