CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DO PARANÁ
Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial

---

DISSERTAÇÃO
apresentada ao CEFET-PR
para obtenção do título de

MESTRE EM CIÊNCIAS

por

MAURICIO PERRETTO

APLICAÇÃO DO ALGORITMO DE OTIMIZAÇÃO POR COLÔNIA DE FORMIGAS AOS PROBLEMAS DE RECONSTRUÇÃO DE ÁRVORES FILOGENÉTICAS E DOBRAMENTO DE PROTEÍNAS

Banca Examinadora:

Presidente e Orientador:

PROF. DR. HEITOR SILVÉRIO LOPES	**CEFET-PR**

Examinadores:

PROF. DR. FERNANDO JOSÉ VON ZUBEN	**UNICAMP**
PROFª. DRª. DENISE FUKUMI TSUNODA	**UFPR**
PROF. DR. CARLOS RAIMUNDO ERIG LIMA	**CEFET-PR**

Curitiba, 25 Fevereiro de 2005.

MAURICIO PERRETTO

APLICAÇÃO DO ALGORITMO DE OTIMIZAÇÃO POR COLÔNIA DE FORMIGAS AOS PROBLEMAS DE RECONSTRUÇÃO DE ÁRVORES FILOGENÉTICAS E DOBRAMENTO DE PROTEÍNAS

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial do Centro Federal de Educação Tecnológica do Paraná, como requisito parcial para a obtenção do título de "Mestre em Ciências" – Área de Concentração: Engenharia Biomédica.

Orientador: Prof. Dr. Heitor Silverio Lopes

Curitiba
2005

Ficha catalográfica elaborada pela Biblioteca do CEFET-PR – Unidade Curitiba

P455a Perretto, Mauricio

Aplicação do algoritmo de otimização por colônia de formigas aos problemas de reconstrução de árvores filogenéticas e dobramento de proteínas / Mauricio Perretto . – Curitiba : [s.n.], 2005.

xviii, 134 f. : il. ; 30 cm

Orientador : Prof. Dr. Heitor Silvério Lopes

Dissertação (Mestrado) – CEFET-PR. Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial. Curitiba, 2005.

Bibliografia: f. 95-100.

1. Bioinformática. 2. Algoritmos. 3. Otimização combinatória. 4. Filogenia. 5. Proteínas – Estrutura. 6. Biologia molecular – Processamento de dados. 7. Software – Desenvolvimento. 8. Engenharia biomédica. I. Lopes, Heitor Silvério, orient. II. Centro Federal de Educação Tecnológica do Paraná. Curso de Pós-Graduação em Engenharia Elétrica e Informática Industrial. III. Título.

CDD: 574.880285
CDU: 573.681.3.06

AGRADECIMENTOS

Aos meus pais, Mauricio e Edilia, que me apoiaram em todos os momentos do meu viver. Aos meus irmãos, avós e demais familiares que sempre estiveram prontos a me auxiliar e me incentivaram em cada fase dos meus projetos pessoais. A Athena por sua alegria e altruísmo fazendo com que me mantivesse sempre animado em busca dos meus objetivos.

A meu orientador, Heitor, por toda sua dedicação e auxílio durante a realização deste projeto. Aos demais professores do CPGEI pelo conhecimento transmitido.

Aos colegas do laboratório de Bioinformática do CEFET e aos amigos do laboratório de Alto Desempenho do Centro Universitário Positivo, especialmente: Éderson Cichazewski, Vanessa Lamoglia, Rodrigo Villaverde, Thiago Gabardo e Valéria Polli, pelos momentos de companhia no desenvolvimento dos mais variados projetos.

Aos colegas professores do Centro Universitário Positivo, especialmente: Edson Pedro Ferlin, José Carlos da Cunha, Valfredo Pilla Júnior e Nestor Cortez Saavedra, pelos conselhos e apoio durante todas as fases do mestrado.

Aos professores da banca que sugeriram diversas melhorias ao projeto, assim como prestaram auxílio na correção de informações equivocadas.

A todas as pessoas que não foram aqui mencionadas, não por serem de menor importância, mas apenas por culpa da minha falta de memória, que tiveram participação direta ou indireta neste projeto ou na formação de minha pessoa.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS

DNA	*Deoxyribonucleic acid* – Ácido Desoxirribonucléico
RNA	*Ribonucleic acid* – Ácido Ribonucléico
mtDNA	*Mitochondrial Deoxyribonucleic acid* – Ácido Desoxirribonucléico Mitocondrial
GRASP	*Greedy Randomized Adaptive Search Procedure* – Procedimento de Busca Melhorada, Adaptavia e Aleatória
AS	*AntSystem* – Sistema de Formigas
ACS	*ant Colony Systems* – Sistema por Colônia de Formigas
ACO	*Ant Colony Optimization* – Otimização por Colônia de Formigas
ANTS	*Approximated Non-Deterministic Tree Search* – Busca Não-Determinística de Árvore Aproximada
OTU	*Operational Taxonomic Unit* – Unidade Taxonômica Operacional
UPGMA	*Unweighted Pair-Group Method using an arithmetic Average* – Método de Par Agrupado sem Peso usando Média Aritmética
HP	*Hydrophobic – Polar* – Hidrofóbico – Polar

RESUMO

O ser humano tem uma grande estima pelo processo de raciocínio que desenvolveu durante a sua evolução. Uma das áreas da computação foi desenvolvida com o objetivo inicial de simular a inteligência humana dentro de programas computacionais. Esta área ficou conhecida como inteligência artificial. Nas últimas décadas a inteligência artificial tem se baseado nas mais diversas formas de organização que tenham padrões. Um desses métodos é o algoritmo de otimização por colônias de formigas, apresentado no início da década de 90, e que apresentou bons resultados para vários problemas que tiveram modelos implementados.

A biologia molecular visa analisar as estruturas moleculares contidas nos seres vivos, dentre elas as seqüências de DNA, RNA e os aminoácidos das proteínas. Devido o grande número de informações envolvidas nessa análise torna-se inviável em termos de tempo de processamento uma busca em todo o espaço de soluções possíveis, o que torna interessante o uso de algoritmos que percorram este espaço de busca de forma eficiente.

Um dos problemas da biologia molecular é a reconstrução de árvores filogenéticas. Ele visa relacionar de forma hereditária as diversas espécies através das informações contidas em suas seqüências. Desta forma é possível saber quais espécies são mais próximas em termos evolutivos..

Outro problema é o dobramento de proteínas. Uma proteína é um polímero que pode desempenhar as mais diversas funções em um ser vivo. A função que uma proteína desempenha esta diretamente relaciona a sua forma tridimensional. Uma proteína é codificada no DNA, e sintetizada no ribossomo de uma forma linear, a partir desta forma ela se dobra sobre a sua estrutura obtendo a sua forma final. Com a compreensão deste processo, seria possível a identificação de proteínas mal formadas e até mesmo o desenvolvimento de novas proteínas com funções específicas.

O presente trabalho visa descrever dos modelos, baseados na otimização por colônia de formigas, desenvolvidos para os problemas. Além disso, foram desenvolvidos recursos especiais que permitem percorrer o espaço de busca de forma mais efetiva obtendo melhores soluções.

Os resultados obtidos com as metodologias propostas apresentaram resultados similares ou até melhores que métodos já conhecidos que utilizaram o algoritmo de otimização por colônia de formigas para os mesmos problemas.

ABSTRACT

The human being has great esteem for the reasoning process developed during its evolution. One of the areas of the computation was developed with the initial objective to simulate human intelligence inside computational programs. This area is known as artificial intelligence. In the last decades artificial intelligence has been basing on the most diverse forms of organization that have standards. One of these methods is the ant colony optimization algorithm, presented in the beginning of nineties, and that achieved good results for some problems that had had implemented models.

Molecular biology aims to analyze the molecular structures present in living creatures, amongst them the sequences of DNA, RNA and protein aminoacids. Due to great number of information being confronted in this analysis it is impracticable in terms of processing time a search in the whole space of possible solutions, what makes interesting the use of algorithms that cover the search space efficiently.

One of the problems of molecular biology is phylogenetic trees reconstruction. It aims to relate hereditarily the several species through information present in its sequences. In that manner, it is possible to know which species are more closely related to one another and which are more distantly related.

Another problem is the proteins doubling. A protein is a polymer that can play several functions in a live being. The function that a protein plays directly relates its three-dimensional form. A protein is codified in the DNA, and synthesized in ribosome linearly, from this form it folds over its structure reaching its final form. Once the understanding of this process is achieved, it would be possible the identification of malformed protein and even though the development of new protein with specific functions.

The aim of this study is to describe the developed models based on the ant colony optimization. Moreover, special sources that allow covering the serch space more effectively had been developed leading to te achievement of better solutions.

The obtained results employing the proposed methodologies revealed similar or better results than those obtained from known methods which employed the ant colony optimization algorithm for resolving the same problems.

CAPÍTULO 1

INTRODUÇÃO

1.1 Motivações

O homem deu a sua espécie o nome de *homo sapiens*, o que evidencia a valorização das habilidades mentais humanas. Diversas ciências estudam este processo conhecido como inteligência, entre elas a pedagogia e a psicologia.

Na ciência da computação uma área foi iniciada tentando compreender e construir algoritmos que simulassem este processo, esta área ficou conhecida como inteligência artificial. Com a evolução das pesquisas a inteligência artificial se expandiu gradativamente abordando diversos problemas e técnicas, não compreendendo apenas simular o processo de inteligência do ser humano, mas buscar a resposta de problemas complexos das mais variadas maneiras. Os métodos antigos que procuram a simulação do ser humano ficaram conhecidos como inteligência artificial clássica, enquanto os novos algoritmos são conhecidos como inteligência computacional.

A maioria dos novos algoritmos são baseados em processos biológicos ou da natureza, como por exemplo, os algoritmos genéticos que foram baseados no predicado contido na evolução das espécies de que os melhores indivíduos têm maiores chances de produzirem descendentes.

Outros métodos se basearam no funcionamento de sistemas dentro do corpo humano como os sistemas imunológicos artificiais (ALVES, 2005), outros na análise de enxames ou colônias de animais como os algoritmos de enxames de partículas e de colônia de formigas (COLORNI e DORIGO, 1991).

O algoritmo de colônia de formigas foi proposto no início da década de 90, baseando-se nas atividades cotidianas de uma colônia de formigas. Mais especificamente, os primeiros sistemas de otimização por colônia de formigas se baseavam em apenas um aspecto deste cotidiano, a busca por alimentos. Foi biologicamente verificado que as formigas à medida que se movimentam liberam um composto químico conhecido como feromônio. Também foi verificado que no processo de busca as formigas analisam tanto os aspectos do ambiente quanto à quantidade deste composto no ambiente. Quanto maior estas quantidades, mais propensas as formigas

estarão de seguir esta trilha. Este processo, conhecido como estigmergia, se baseia em um sistema de realimentação positiva que torna os melhores caminhos mais atraentes para as formigas e um sistema de realimentação negativa que faz com que os piores caminhos sejam esquecidos com o passar do tempo. Como mencionado, este processo é semelhante à rotina de busca de alimentos das formigas.

No período de uma década o algoritmo de otimização por colônia de formigas tem sido aplicado em diversos problemas nos mais diversos campos de atuação, tendo apresentado resultados equivalentes ou melhores do que os principais algoritmos propostos até então.

Uma das áreas que tem tido grande aplicação dos métodos computacionais é a biologia, principalmente a área de biologia molecular. A biologia molecular visa analisar as estruturas moleculares contidas nos seres vivos, dentre elas as seqüências de DNA, RNA e os aminoácidos das proteínas. O estudo destas seqüências permite esclarecer como são codificadas as informações dos seres vivos e como são criadas as proteínas que executam as principais funções dentro dos organismos vivos.

A bioinformática é uma sub-área da computação que visa a aplicação de algoritmos computacionais para o estudo dos mais diversos problemas da biologia molecular. Dentre os diversos algoritmos que têm sido utilizados, os algoritmos de inteligência computacional têm obtido bons resultados, tendo sido aplicados em diversos problemas e codificados de diversas maneiras.

Dois problemas importantes da biologia molecular são: a reconstrução de árvores filogenéticas e o dobramento de proteínas.

A reconstrução de árvores filogenéticas é um dos problemas mais antigos da biologia que tem uma sub-área específica conhecida como filogenia. Neste processo é necessário, inicialmente, definir quais aspectos dos organismos vivos serão utilizados para realizar a classificação. Esta seleção inicial já produz uma série de mudanças nas árvores construídas. São utilizados seqüências de DNA, RNA, mtDNA, seqüências de genes específicos e seqüências de aminoácidos de proteínas. Por último, o maior problema na reconstrução de árvores é o grande número de árvores que podem ser produzidas com um pequeno número de espécies, por exemplo: para se construir a árvore correta de um conjunto inicial com 50 indivíduos seria necessária analisar $2{,}75 \times 10^{76}$ árvores candidatas (WEIR, 1996).

O dobramento de proteínas visa estudar o processo que ocorre com as seqüências de aminoácidos após a sua síntese no ribossomo. Este estudo é bastante

significativo porque a forma de uma proteína é de extrema importância para que ela desempenhe sua função. Se uma proteína se dobra de uma forma diferente a sua estrutura normal, esta não poderá desempenhar sua função corretamente, podendo até se tornar algo prejudicial ao organismo. Assim como no caso da reconstrução o número de soluções possíveis de serem obtidas cresce muito rápido com um pequeno número de resíduos na seqüência a ser analisada, até mesmo com os modelos mais simples que modelam o processo, sendo que a análise de todas as conformações possíveis para o dobramento de seqüências reais torna-se inviável.

O método de otimização de colônia de formigas tem se apresentado bastante robusto para a solução de diversos problemas computacionais podendo-se citar como problemas que tem modelos desenvolvidos sobre esta metodologia: problemas de busca do melhor caminho como o caixeiro viajante, otimização de agendamento de tarefas, roteamento estático e dinâmico de pacotes em rede. Porém na área da bioinformática o modelo de otimização por colônia de formigas tem sido pouco utilizado obtendo ainda resultados inferiores a outros métodos.

1.2 Objetivos

O presente trabalho tem como objetivo principal desenvolver metodologias baseadas no algoritmo de otimização por colônia de formigas para a solução de dois problemas de biologia molecular: a reconstrução de árvores filogenéticas e o dobramento de proteínas.

Como objetivos secundários tem-se a construção de dois *softwares* para utilização em estudos de biologia molecular que permitam a reconstrução de árvores filogenéticas e o estudo sobre o dobramento de proteínas. Deseja-se que estes *softwares* sejam ao mesmo tempo robustos do ponto de vista de obtenção de bons resultados, e que sejam rápidos em termos de tempo de processamento. Isto permitiria a sua utilização na análise destes problemas com um número maior de espécies no caso da reconstrução ou seqüências maiores no caso do dobramento.

Pretende-se implementar um modelo que permita a utilização do algoritmo original do ACO (*Ant Colony Optimization* – Otimização por Colônia de Formigas) e a partir de testes realizados com este modelo sugerir modificações ou mesmo acréscimos ao algoritmo que permitam a obtenção de melhores resultados sem que o desempenho de tempo seja acrescido significativamente.

1.3 Estrutura da dissertação

Esta dissertação está organizada da seguinte forma. No Capítulo 2 faz-se uma apresentação sobre o algoritmo de otimização por colônia de formigas, sua idéia básica e funcionamento. No Capítulo 3 é apresentado o problema de reconstrução de árvores filogenéticas. No Capítulo 4 é detalhado o problema do dobramento de proteínas bem como alguns dos métodos atualmente utilizados para modelagem destes problemas. No Capítulo 5 são apresentados trabalhos correlatos que utilizaram o algoritmo de otimização por colônia de formigas para os problemas propostos ou permitiram a construção de um modelo do algoritmo para o problema. No Capítulo 6 é apresentado em detalhes o desenvolvimento da metodologia proposta para os dois problemas. No Capítulo 7 relatam-se os resultados obtidos. E, finalmente, o Capítulo 8 apresenta a discussão dos resultados, as conclusões do trabalho e as propostas de trabalhos futuros.

No final da dissertação são incluídos diversos anexos que visam detalhar algumas informações ou métodos utilizados durante o decorrer do trabalho ou permitir a comparação entre o método apresentado e novos propostos. Para isso no anexo 1 são apresentados os 20 aminoácidos essenciais e sua classificação em hidrofóbico ou polar, no anexo 2 são apresentadas as 15 seqüências utilizadas para a análise do programa de dobramento de proteínas. O anexo 3 visa detalhar o processo de cálculo de distâncias evolutivas através do método de complexidade de Kolmogorov utilizado no programa de construção de árvores filogenéticas. No anexo 4 é apresentado como é feito o cálculo de distância topológicas entre árvores através do método de Robinson-Foulds, este método foi utilizado para analisar a semelhança entre a árvore produzida pelo método proposto e as árvores obtidas por outros métodos. Os anexos 5, 6 e 7 são os manuais de usuário dos dois softwares desenvolvidos e a documentação de desenvolvimento dos dois softwares respectivamente. Estas diversas informações foram anexadas ao trabalho visando um maior conhecimento de todo o trabalho e permitir a sua posterior continuação.

CAPÍTULO 2

OTIMIZAÇÃO POR COLÔNIA DE FORMIGAS

2.1 Inteligência de enxame

Uma das áreas primordiais de estudo da biologia tem como foco o estudo das formas de organização dos animais. Destas formas de organização uma que atraiu grande interesse são as colônias de insetos. Quando vistos isolados, os insetos parecem não conseguir desenvolver suas atividades de maneira eficaz. Porém, quando vistos em grupo, como uma colônia, esses insetos parecem ter uma organização bastante complexa, conseguindo elaborar planos, definir objetivos e funções.

Como exemplo destes fatos tem-se a organização de abelhas para construção da estrutura das colméias, busca de alimentos e guarda da colméia; os cupins através do seu intricado sistema de túneis e construção de torres; e as formigas na busca por alimento, organização de caminhos entre ninhos e divisão de tarefas conforme o tamanho.

Este processo de auto-organização através de colônias, observado por RABAUD (1937) é definido como inteligência de enxame, isto é, a capacidade de entidades simples conseguirem executar tarefas complexas a partir da sua organização.

Auto-organização é um processo dinâmico no qual uma estrutura global bem definida surge a partir de interações de componentes de baixo nível, neste caso os insetos. Para que este processo de auto-organização ocorra é necessário que algumas regras especifiquem as formas de interação dos componentes com o meio local onde estão localizados, sem referência a todo o sistema.

Este conjunto de regras, apresentado por BONABEAU, DORIGO e THERAULAZ (1999), pode ser resumido em quatro grupos principais:

- Auto-alimentação positiva (amplificação) – a amplificação dos melhores caminhos faz com que estes se tornem caminhos preferenciais na busca;
- Auto-alimentação negativa – utilizado para contrabalançar o efeito da alimentação positiva fazendo com que caminhos não freqüentemente utilizados sejam esquecidos com o passar do tempo, também utilizado para estabilizar um padrão;

- Aleatoriedade – a auto-organização surge da amplificação das flutuações, que permitem a localização de novas soluções superando mínimos locais;
- Interações múltiplas – Um indivíduo único pode produzir uma estrutura organizada através de uma trilha estável de feromônios. Porém, esta estrutura não será otimizada; para que um sistema auto-organizado encontre boas soluções é necessário que múltiplos indivíduos interajam entre si a partir dos seus resultados e através do ambiente comum.

2.2 Estigmergia

A interação dos insetos para produzir um sistema auto-organizado pode ocorrer de duas formas distintas:

- Direta: a qual ocorre através do contato táctil entre os animais por suas antenas ou mandíbulas, do contato visual ou mesmo quimicamente pelos odores de outros insetos;
- Indireta: esta forma de comunicação ocorre quando um indivíduo de uma população altera o seu meio próximo, fazendo com que o ambiente local onde ele está seja modificado, e que outros indivíduos que estejam no mesmo meio, em um momento posterior, tenham suas decisões afetadas por esta modificação individual;

Esta segunda forma de interação, chamada de estigmergia foi apresentada por GRASSÉ (1959) para explicar a coordenação de tarefas na reconstrução de ninhos de cupins.

A estigmergia em si não explica os mecanismos detalhados pela qual os indivíduos coordenam suas atividades. Porém, ela explica inicialmente a relação de indivíduo e colônia que existe nos sistemas de inteligência de enxame. Um exemplo bastante evidente da estigmergia em ação é a busca por alimentos realizada pelas formigas.

2.3 Formigas reais e a busca por alimentos

Uma das principais atividades realizadas por formigas é a localização de uma fonte de alimento e, a partir disto, a busca dos alimentos desta fonte. A maioria das

espécies de formigas utiliza um processo de criar uma trilha e segui-la para executar esta tarefa. Esta trilha é feita a partir de uma substância química denominada feromônio, que é depositada no caminho durante a movimentação das formigas entre a fonte de comida e o ninho. As formigas terão uma predisposição por seguir trilhas com uma maior quantidade de feromônios na sua movimentação.

O processo pelo qual uma formiga é influenciada a seguir um caminho a partir do feromônio depositado por outra formiga é chamado de recrutamento, e o processo de deixar feromônio em uma trilha já definida é chamado de reforço. Nestes dois processos fica clara a estigmergia como forma de comunicação indireta entre formigas.

A busca por alimentos se torna um processo de otimização visto que quanto menor o caminho que as formigas fizerem mais comida elas obterão com menor esforço e tempo. Na seqüência, serão mostrados dois experimentos propostos por DENEUBOURG (1990) que descrevem essa otimização dos caminhos.

Os dois experimentos buscam apresentar a maneira de auto-organização das formigas. No primeiro, o caminho entre o ninho e a fonte de comida é dividido em duas partes iguais, como mostrado na figura 1.

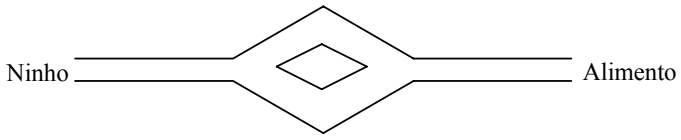


Figura 1. Caminho dividido igualmente para o primeiro experimento

As formigas são liberadas e procuram a fonte de comida através de um caminho aleatório no início, mas que vai rapidamente convergindo para uma das pontes. Após poucos minutos a maioria das formigas já convergiu para um caminho único. Porém, algumas formigas ainda fazem um caminho aleatório, para procurar novos caminhos que ainda possam ser descobertos.

No gráfico da figura 2, é apresentado o percentual de formigas seguindo cada um dos caminhos em um espaço de tempo definido.

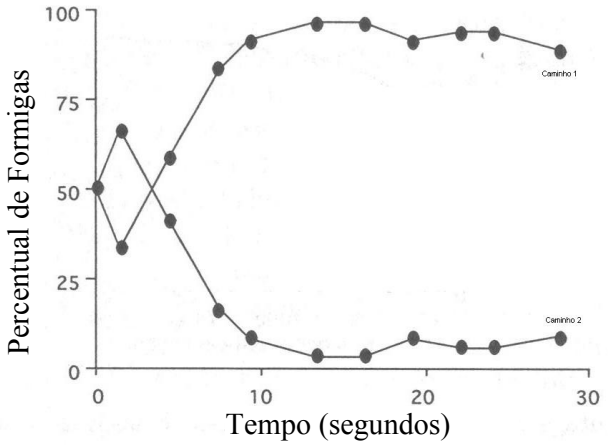


Figura 2. Relação entre porcentual de formigas no caminho durante o tempo

O segundo experimento tinha como objetivo verificar como as formigas se comportariam a mudanças no ambiente global, visto que isto normalmente ocorre no mundo real. Para isso foi desenvolvido um novo caminho entre o ninho e a fonte de alimento. Neste caminho existem duas partes mais longas e duas partes mais curtas, conforme a figura 3. Inicialmente quando apresentado os dois caminhos ao mesmo tempo as formigas rapidamente convergem para o menor caminho. Na segunda parte do experimento as formigas só podem percorrer os caminhos mais longos. Após a deposição de feromônio neste pior caminho e conseqüente fixação da trilha, são liberados os caminhos mais curtos para serem percorridos.

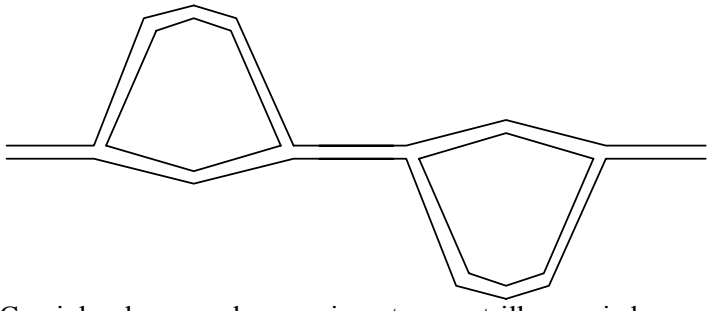


Figura 3. Caminho do segundo experimento com trilhas mais longas e curtas

Na figura 4 são apresentados dois gráficos de barra. O primeiro apresenta o número de formigas na menor trilha após quatro minutos, quando apresentados os dois caminhos simultaneamente. No segundo gráfico apresenta-se o número de formigas no

menor caminho na segunda situação onde uma trilha de feromônios já foi formada no pior caminho. É possível ver que as formigas não trocam de caminho imediatamente quando apresentado o novo menor caminho, porém este já foi encontrado e é cada vez mais utilizado.

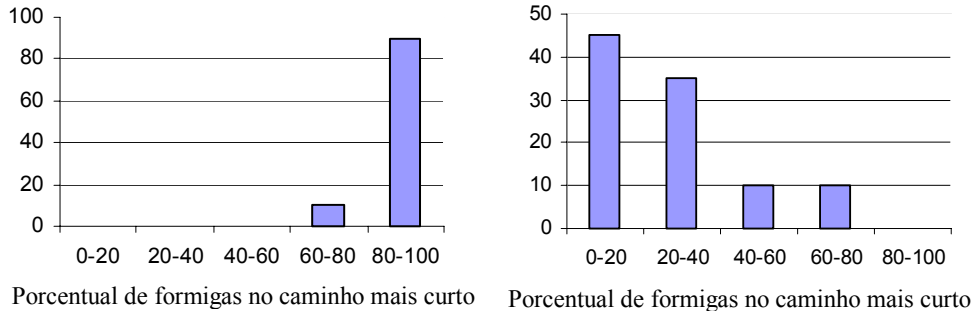


Figura 4. Porcentagem de formigas que escolheram o menor caminho; 1º - os dois caminhos apresentados simultaneamente; 2º - melhor caminho apresentado 30 minutos após o pior caminho

Além disto, formigas reais na busca por alimentos conseguem achar estes recursos sem informações visuais, pois são praticamente cegas, e adaptar-se a alterações no ambiente otimizando o caminho entre o ninho e a fonte de alimentos (DORIGO, GAMBARDELLA, 1998).

Este fato se deve a um fenômeno químico conhecido como deposição de feromônio. Feromônio é uma substância química que influencia o comportamento de outros animais da mesma espécie. Desta forma, as formigas tendem a seguir uma trilha.

Um exemplo clássico da constituição de uma trilha de feromônios e busca por um caminho alternativo é apresentado na figura 5. No primeiro caso há um caminho, já conhecido pelas formigas, do ninho até a fonte de alimentos. Na segunda figura é inserido um obstáculo no meio desta trilha e as formigas se dispersam para os dois lados, pois não existe ainda uma trilha clara para contornar o novo obstáculo. Conforme as formigas contornam o objeto e novamente chegam à fonte de alimento vai se formando uma nova trilha que será mais forte no menor caminho. Esta situação é apresentada na terceira parte da mesma figura. Este exemplo foi apresentado por DORIGO e DI CARO (1999).

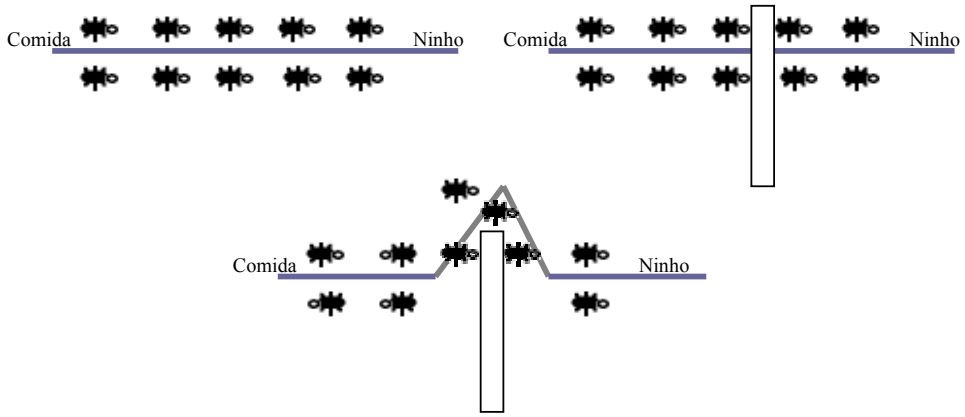


Figura 5. a) Formigas com trilha de feromônios já formada entre comida e ninho; b) Um obstáculo é inserido no meio da trilha, com melhor e pior caminho; c) nova trilha de feromônios formada através do melhor caminho.

Apesar da seleção por um caminho menor, é necessário notar que algumas formigas ainda escolhem o caminho mais longo, não permitindo que o algoritmo convirja. No caso de uma alteração no sistema, este caminho pouco atraente pode se tornar um caminho válido para a solução.

2.4 A heurística de otimização por colônia de formigas

A otimização por colônia de formigas é uma maneira de desenvolver algoritmos metaheurísticos para problemas de otimização combinatorial.

Algoritmos metaheurísticos são algoritmos que utilizam mais de uma informação na definição da qualidade de uma resposta, normalmente as variáveis utilizadas são: uma forma de cálculo heurístico e uma variável interna ao método que inicia com valor zero e é incrementada no decorrer do método melhorando o resultado. Estes métodos se utilizam deste recurso para construir uma solução escapando de máximos locais.

A principal característica de um ACO (*Ant Colony* Optimization) é o uso explícito de elementos de soluções anteriores para cálculo das próximas soluções. Como um GRASP (*Greedy Randomized Adaptive Search Procedure*) (FEO, RESENDE; 1995), porém ele utiliza um método aleatório de estado inicial como o Monte Carlo.

O algoritmo de otimização por colônia de formigas, inicialmente proposto por COLORNI e DORIGO (1991), e conhecido como AntSystem (AS), foi modelado para o problema do caixeiro viajante. Este problema bastante conhecido visa buscar o caminho

que minimize o trajeto entre diversos nós interconectados através de arcos em uma estrutura semelhante a um grafo, como apresentado na figura 6.

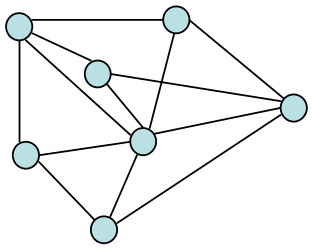


Figura 6. Estrutura de nós interconectados por arcos utilizada no problema do caixeiro viajante

Para cada arco i,j do grafo, é definida uma variável τ_{ij} , conhecida como trilha artificial de feromônio. Esta variável é incrementada conforme o passar das formigas e decrementada a cada ciclo. A intensidade da trilha de feromônio definirá a utilidade da trilha para as formigas, isto é, quanto maior a quantidade de feromônio, melhor é esta trilha e as formigas a utilizarão em detrimento dos outros arcos.

A cada nó, a formiga artificial executa uma função estocástica para calcular a probabilidade de utilização dos arcos. Inicialmente a função de probabilidade foi desenvolvida relacionanda apenas com o processo de estimergia ocasionado pelo acúmulo de feromônio pelas trilhas. Desta forma, a função continha apenas um termo relacionado à deposição deste composto, como apresentado na equação 1.

$$p(i,j) = [\tau(i,j)] \tag{1}$$

Esta abordagem, apesar de obter alguns resultados para problemas específicos, não obtinha bons resultados para diversas instâncias do problema do caixeiro viajante. Além disto, ela não tinha uma grande confiabilidade e repetibilidade dos resultados. Assim, decidiu-se pela implementação de um segundo termo que seria um valor heurístico relacionado à natureza do problema, esta nova forma é apresentada na equação 2.

$$p(i,j) = [\tau(i,j)] \cdot [\eta(i,j)] \tag{2}$$

Por último, foram definidos expoentes para os dois termos fazendo com que fosse possível definir qual teria uma preponderância na probabilidade de cada caminho. A equação 2 também foi normalizada dividindo-se os termos desta equação pelo somatório de todas as probabilidades possíveis para o movimento a partir deste ponto, como apresentado na equação 3:

$$p(i,j) = \frac{[\tau(i,j)]^{\alpha} \cdot [\eta(i,j)]^{\beta}}{\sum_{l \in N_i^k} [\tau(i,j)]^{\alpha} \cdot [\eta(i,j)]^{\beta}} \tag{3}$$

onde:

η(i,j): é um valor heurístico relacionado a natureza do problema modelado. No caso do problema do caixeiro viajante *η(i,j) = 1/d(i,j)*, sendo *d(i,j)* a distância entre os nós *i* e *j*;

N_i^k: é o conjunto de nós conectados a *i* que ainda não foram visitados;

τ(i,j): é o valor da trilha de feromônios no arco que conecta os dois nós;

α: é um parâmetro que pondera a importância relativa da trilha de feromônios na decisão de movimentação da formiga;

β: é um parâmetro que pondera a influência relativa da distância no processo de decisão.

Na equação 3 se *α*=0 as formigas seguirão a heurística do vizinho mais próximo para resolução, enquanto que se *β*=0 as formigas selecionarão um caminho aleatório.

No algoritmo original, uma formiga iniciava em um nó qualquer e continha uma memória da rota já realizada, para armazenar os nós já visitados. Após isto, a formiga iniciava um caminho probabilístico percorrendo todos os nós do grafo apenas uma vez. Em cada nó *i*, a formiga *k* calcula a probabilidade de seguir para um próximo nó não visitado *j* a partir equação 3 já apresentada.

Após a execução de um ciclo que ocorre quando todas as formigas já terminaram o caminho, calcula-se o escore de cada caminho através da equação 4. Esta equação visa minimizar o caminho obtido pela formiga.

$$\Delta\tau(i,j) = \begin{pmatrix} \frac{1}{L_k}, se(i,j) usado \\ 0, caso \quad contrário \end{pmatrix} \tag{4}$$

Na equação 4, L_k é o escore do caminho percorrido pela formiga k durante o ciclo.

Na seqüência inicia-se o processo de deposição de feromônio conforme a regra apresentada na equação 5, onde m é o número de formigas que percorreram o grafo naquele ciclo.

$$\tau(i,j) = \rho.\tau(i,j) + \sum_{k=1}^{m} \Delta\tau(i,j) \tag{5}$$

Nesta equação, tem-se:

ρ : representando a taxa de evaporação de feromônio. Este parâmetro é utilizado para que os caminhos que são menos utilizadas sejam esquecidas com o passar do tempo

Δτ : é a taxa de incremento do feromônio definida pela equação 4.

O algoritmo inicial tinha como único critério de parada o número de vezes que as formigas percorreriam o grafo.

2.5 Sistemas de otimização por colônia de formigas

Como apresentado por DORIGO e DI CARO (1999), a principal tarefa de uma formiga artificial é a busca do menor caminho em um par de nós de um grafo no qual o problema foi mapeado.

Os sistemas de otimização por colônia de formigas são formas de resolução de problemas que simulam o comportamento de um grupo de formigas na busca pelo melhor caminho.

O incremento da trilha de feromônio pode ocorrer de duas formas distintas. No primeiro caso, a cada movimento da formiga artificial é incrementado o arco relativo ao movimento (*online step-by-step pheromone trail updating*). No segundo caso, os arcos das trilhas obtidas são incrementados ao final de um ciclo (*delayed pheromone trail updating*) que seria o tempo que todas as formigas levaram para completar seus caminhos (CORDON, VIANA, HERRERA *et al.*, 2000).

Existem diversas diferenças entre as formigas reais e as geradas em um sistema artificial (PARPINELLI, LOPES, FREITAS, 2001), como por exemplo:

- Formigas artificiais têm memória;
- O tempo é discreto.

Em contrapartida, os ACS (*Ant Colony Systems*) permitem a simulação de características reais dos enxames, como por exemplo:

- As formigas artificiais tendem a preferir caminhos com uma quantidade maior de feromônio;
- Os menores caminhos têm um crescimento maior na taxa de feromônios;
- É utilizada uma forma de comunicação indireta, a trilha de feromônios, para se descobrir o melhor caminho.

Em um sistema por colônia de formigas, as duas principais tarefas são:

- A construção de uma representação do problema de uma forma verossímil, normalmente uma estrutura na forma de um grafo e que permita uma regra probabilística de transição entre nós baseada na trilha de feromônios e no valor de cada arco;
- O desenvolvimento de uma heurística para transição de nós que possa avaliar a qualidade dos caminhos tomados.

2.6 Contribuições importantes

Após estes trabalhos iniciais, surgiram diversos trabalhos que utilizam a metodologia de construção de algoritmos do ACO como forma básica para a resolução de problemas das mais diversas áreas, como: problemas de ordenação seqüencial, (GAMBARDELLA, DORIGO, 2000), e localização de rotas para veículos. (BULLNHEIMER, HARTL, STRAUSS, 1999) (BULLNHEIMER, HARTL, STRAUSS 1999.2). Também foram propostas diversas modificações junto ao algoritmo inicial.

Na seqüência, serão apresentadas as principais contribuições realizadas por alguns trabalhos que serão exploradas nos próximos capítulos.

A primeira alteração do ACS para resolução de problemas do caixeiro viajante com um grande número de nós foi proposta por GAMBARDELLA e DORIGO (1996). O método proposto visava diminuir o número de nós que serão analisados a cada

movimento da formiga sobre o grafo. Para isso, para cada nó do grafo é criada uma lista de nós candidatos que contêm os melhores valores heurísticos de transição. A formiga, quando em um nó, verifica apenas a probabilidade de transição dos nós contidos na lista e define o próximo movimento.

ANTS (*Approximated Non-Deterministic Tree Search*) é uma extensão, proposta por MANIEZZO (1999), do ACO original. Ele é uma hibridização entre o ACO e os algoritmos de busca em árvores. As principais diferenças implementadas no ANTS foram o parâmetro de atratividade e o mecanismo de incremento da trilha.

No parâmetro de atratividade, é definido um limite inferior e a cada movimento é computado um valor de limite com uma solução completa. Quanto menor este limite, melhor será a solução final. Porém se o limite ultrapassar o valor da solução completa o caminho é descartado.

ANTS foi o primeiro algoritmo de ACO que implementou uma maneira de prevenir a estagnação, que é a convergência de todas as formigas para um caminho único, não havendo mais caminhos alternativos. Para isto, o procedimento de incremento foi alterado para a equação 6.

$$\Delta\tau(i,j) = \tau_0.(1 - \frac{z_{curr} - LB}{z - LB}) \tag{6}$$

Nela, é executada uma avaliação de cada trilha em relação às últimas k trilhas percorridas. A média de movimentos destas trilhas (z) é calculada a cada nova solução z_{curr} que é comparada com a média obtida. Se z_{curr} é menor que a média o valor de deposição de feromônio na trilha para as soluções é aumentado, senão este valor é diminuído. Isto significa que, se ocorrer apenas soluções ótimas, os valores de feromônios não terão incremento nem decremento.

O algoritmo Max-Min AS foi proposto por STÜTZLE e HOOS (2000), sendo que testes preliminares mostraram ser a melhor alternativa na solução de problemas como o caixeiro viajante.

Neste algoritmo foi implementada uma forma agressiva de incremento de feromônio, na qual apenas o melhor caminho obtido em um ciclo teria sua trilha de feromônio acrescida.

Este tipo de abordagem normalmente leva o algoritmo a uma convergência prematura, perdendo variedade e estabilizando facilmente em um máximo local. Porém, além desta alteração implementou-se uma outra forma de avaliação das taxas de feromônio prevenindo a convergência prematura.

Por último o algoritmo Max-Min utiliza métodos de busca local, utilizando para isso um método bastante conhecido em algoritmos genéticos, chamado de correlação distância-fitness (JONES, FORREST, 1995).

CAPÍTULO 3

FILOGENIA

A filogenia é a construção da sucessão genética das espécies orgânicas, isto é, a representação da seqüência de evolução das espécies animais e vegetais existentes e que foram extintas. O termo filogenia foi criado pelo naturalista alemão Ernst Heinrich Haeckel (1834 – 1919).

A idéia da construção de árvores filogenéticas nasceu naturalmente da teoria da evolução de Charles Darwin, pois justamente se os animais sofrem uma seleção natural e mutações para adaptar-se ao meio, isto tornaria possível o mapeamento da seqüência destas alterações nas espécies e seria possível a localização da origem das espécies. Apesar de Darwin ter trabalhado nesta área, não foi seu objetivo desenvolver uma seqüência filogenética geral, preferindo trabalhar com grupos previamente classificados, para validar que as diferenças dentro destes grupos viriam de uma evolução e adaptação ao meio.

As principais representações filogenéticas foram possivelmente as feitas por Lamarck em 1809 e utilizadas por Darwin em sua obra *A Origem das Espécies* (1859). Porém, a primeira pessoa a tentar mapear todas as espécies construindo uma árvore ligando as diversas espécies foi Haeckel em 1866.

Nesta árvore, Haeckel abrangeu todos os grupos de seres vivos conhecidos, entre eles microorganismos, vegetais e animais. Porém, esta classificação foi realizada a partir de aspectos questionáveis cientificamente e a árvore apresentada não tem nenhuma semelhança com as árvores atualmente construídas.

Segundo REIJMERS, WEHRENS, DAEYAERT *et al.* (1999), os estudos de filogenia visam obter uma estrutura em forma de árvore que define um relacionamento ancestral entre um conjunto de objetos. Este conjunto de objetos é normalmente conhecido como OTU (*Operational Taxonomic Unit*). Como OTU's pode-se considerar as características físicas como: número de patas, órgãos internos, estrutura óssea; ou gênicas, como: seqüências de DNA, proteínas.

Uma árvore pode ser construída a partir de apenas uma OTU ou agrupar diversas OTUs. Porém, quando são agrupadas é necessário definir quais atributos têm maior importância na classificação, o que pode se tornar um critério arbitrário.

Uma distinção que se faz necessária é apresentada por NEI e KIMURA (2000): os evolucionistas normalmente procuram em uma árvore a história evolucionária de um grupo de espécies ou de uma população e esta árvore normalmente é chamada de árvore de espécie. Entretanto, as árvores construídas a partir das seqüências de genes das espécies não necessariamente corresponderão às respectivas árvores de espécies. Desta forma, convencionou-se chamar as árvores produzidas através deste método de árvores gênicas.

A primeira distinção destas foi apresentada por TATENO (1982), e pode ser vista na figura 7. A figura 7a é a árvore de espécie correspondente obtida através de características biológicas. As árvores gênicas apresentadas em 7b e 7c foram obtidas através de seqüências diferentes e, por isso, apresentam pequenas diferenças, sendo que em 7b as espécies X e Y contêm genes mais proximamente relacionados, o que faz uma quebra diferente no topo da árvore. Este fato também ocorre em 7c, porém isto se deve apenas à aproximação de Y com Z, o que tornou o elemento X totalmente diferente das outras espécies.

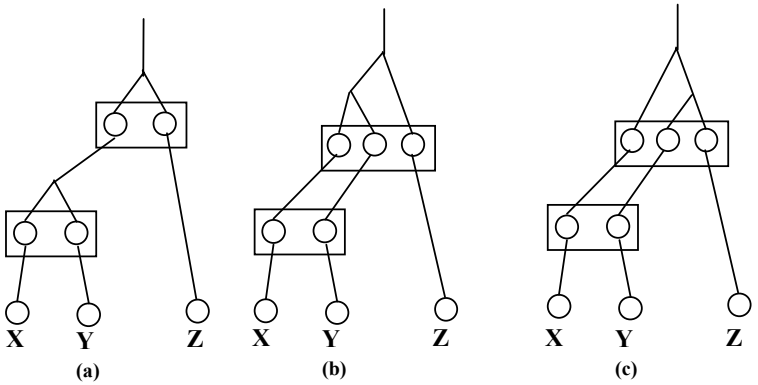


Figura 7. Três modelos de árvores gênicas possíveis a partir da mesma árvore de espécie

3.1 Árvores com raiz ou sem raiz

Na construção da árvore filogenética correspondente a um grupo de espécies, deve-se analisar se a árvore gerada conterá ou não raiz ou se será um cladograma.

Um cladograma é apenas um agrupamento seqüencial entre as espécies, formando, desta forma, uma árvore em forma de escada, como a apresentada na figura

8. Além disso, no cladograma não é possível levar em consideração os ancestrais comuns entre espécies.

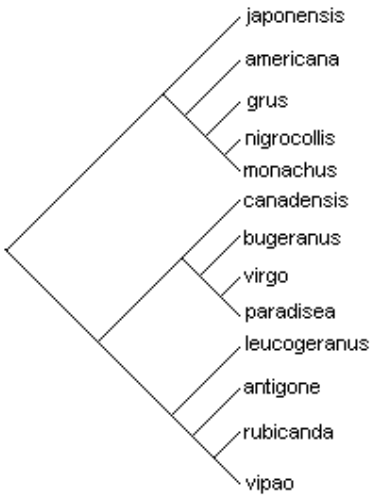


Figura 8. Exemplo de um cladograma

Uma árvore sem raiz, como apresentada na figura 9, apresenta a distância evolutiva entre espécies e os ancestrais comuns, porém não tem uma noção de ordem temporal dos eventos evolutivos.

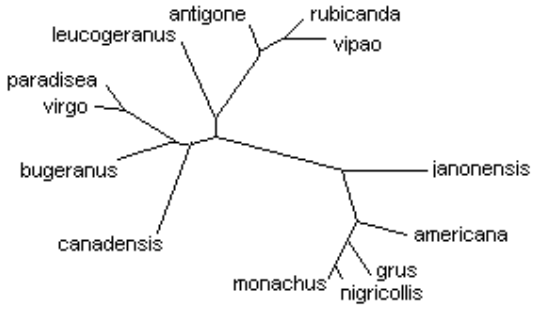


Figura 9. Construção de uma árvore sem raiz

As árvores com raiz são as mais completas, podendo fornecer as informações temporais entre espécies e ancestrais, assim como as distâncias evolutivas. O principal problema desse tipo de árvore é o elevado processamento para obtenção da melhor localização para a raiz. Para isto devem ser avaliados todos os ramos obtidos na construção da árvore. Na figura 10 é apresentada a localização do ponto onde seria a raiz da árvore da figura 9.

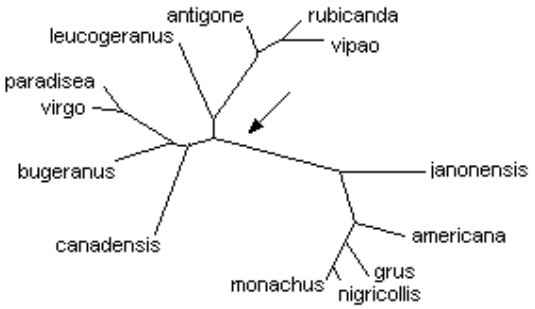


Figura 10. Melhor localização da raiz

Segundo NEI e KUMAR (2000), o número de topologias de árvores que podem ser produzidas apenas com a bifurcação de espécies e sem raiz e com tamanho igual a *n elementos* é dado pela equação 7a. Já no caso de árvores com raiz o número de árvores a serem analisadas é dado pela equação 7b.

$$\frac{(2n-3)!}{2^{n-2}(n-2)!} \tag{7a}$$

$$\frac{(2n-5)!}{2^{n-3}(n-3)!} \tag{7b}$$

Desta forma, o número de árvores que necessitam ser avaliadas para apenas 10 espécies seria de aproximadamente 34 milhões. Atualmente uma busca por todas as árvores envolvendo apenas 60 espécies é inviável em termos de tempo de processamento.

3.2 Métodos de reconstrução de árvores

Diversos métodos foram propostos para a reconstrução de árvores filogenéticas. A primeira classificação destes métodos foi proposta por FELSENSTEIN (1981) em três classes distintas: métodos baseados em matriz de distância, método da máxima parcimônia e máxima verossimilhança. Existem outros dois tipos de classificações propostas. Porém, neste trabalho, optou-se por esta classificação pelas diferenças claras dos dados de entrada dos três grupos.

3.2.1 Métodos baseados em matriz de distância

Os métodos baseados em matriz de distância utilizam a matriz obtida pela distância evolutiva entre cada par de espécies. Na construção da árvore filogenética, é utilizada uma forma de relacionamento entre estas distâncias.

Estes métodos podem ser resumidamente considerados como um processo de agrupamento ou clusterização. Agrupamento pode ser definido como o processo de formar grupos com objetos que tenham características similares.

3.2.1.1 UPGMA

O primeiro algoritmo a ser classificado como baseado em matriz de distâncias foi o UPGMA (*Unweighted Pair-Group Method using an arithmetic Average)*, proposto por SOKAL e MICHENER (1958). Este método foi inicialmente proposto com o intuito da análise de similaridades das características fenotípicas de espécies na taxonomia tradicional. Por este motivo, a árvore obtida normalmente é chamada de fenograma. Características fenotípicas são as formas externas ou funções do organismo analisado.

Entretanto, este método também pode ser utilizado na reconstrução de árvores a partir de seqüências moleculares, desde que a taxa de substituição de bases entre todas as espécies analisadas se mantenha constante. Quando esta taxa é constante o UPGMA produz respostas rápidas e confiáveis comparados com outros métodos similares.

O algoritmo do UPGMA é bastante simples e começa com uma matriz quadrada *n*x*n*, onde *n* é o número de espécies. Seleciona-se a célula (*i*, *j*) com o menor valor da matriz. Ela contém a distância evolutiva entre as espécies *i* e *j* que serão agrupadas formando uma única espécie *u*. A distância evolutiva de cada ramo à nova espécie é dada pela metade da distância entre as espécies $d_{(1|2)u} = \frac{d_{1,2}}{2}$, e recalculam-se os valores de distância para todos os outros *k* elementos a partir da equação 8.

$$d_{uk} = \frac{d_{1k} + d_{2k}}{2} \tag{8}$$

Esse procedimento é executado até que sejam agrupados todos os elementos da matriz. Um exemplo do método UPGMA é mostrado na figura 11.

	A	B	C	D	E
A	0				
B	19	0			
C	27	31	0		
D	**8**	18	41	0	
E	33	36	32	17	0

➔

	AD	B	C	E
AD	0			
B	**18,5**	0		
C	34	31	0	
E	25	36	32	0

	ADB	C	E
ADB	0		
C	32,5	0	
E	**30,5**	32	0

	ADBE	C
ADBE	0	
C	32,25	0

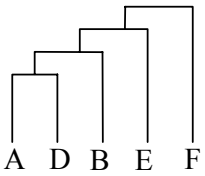


Figura 11. Exemplo de reconstrução de árvore usando UPGMA

Como apresentado anteriormente, o método UPGMA não é confiável quando a taxa de substituição de nucleotídeos não é constante. Para resolver isto é necessário utilizar um método que utilize taxas variadas para cada ramo. Entre estes métodos se destacam os que utilizam a regra dos quadrados mínimos. Existem diversos métodos baseados em quadrados mínimos, sendo os mais utilizados o ordinário e o ponderado (NEI, KUMAR, 2000).

No método dos quadrados mínimos ordinários, é calculada a soma residual das distâncias da matriz original e dos valores obtidos na reconstrução da árvore, equação 9.

$$R_s = \sum_{i<j} (d_{ij} - e_{ij})^2 \qquad (9)$$

Para as topologias obtidas na reconstrução deve-se calcular o valor residual de R_s, onde: d_{ij} é a distância obtida entre duas espécies na árvore e e_{ij} é a distância evolutiva real. O objetivo é a minimização de R_s.

O método ponderado, proposto por FITCH e MARGOLIASH (1967), transforma o erro absoluto obtido através do método ordinário em um erro relativo com a adição do termo $1/d_{ij}$ no somatório, obtendo-se a equação 10.

$$R_s = \sum_{i<j} \frac{(d_{ij} - e_{ij})^2}{d_{ij}} \qquad (10)$$

3.2.1.2 NEIGHBOR-JOINING

O método de *neighbor-joining*, proposto por SAITOU e NEI (1987), procura reconhecer os pares mais próximos dos elementos, de forma a minimizar o comprimento da árvore construída. Este método é considerado um dos métodos baseados em matriz de distâncias que produz as árvores evolutivas mais coerentes.

O método é considerado uma simplificação do método da evolução mínima proposto por EDWARDS e CAVALLI-SFORZA (1963). Ele procura identificar os pares mais próximos de elementos, chamados de vizinhos (*neighbors*), agrupando os vizinhos mais próximos tende-se a minimizar o comprimento total da árvore produzida. Um par de vizinhos é formado por dois elementos conectados por apenas um ancestral em uma árvore bifurcada e sem raiz. Por exemplo, na figura 12 as espécies homem e chimpanzé são vizinhos, porém homem e gorila não são, pois há mais de um nó interno entre eles.

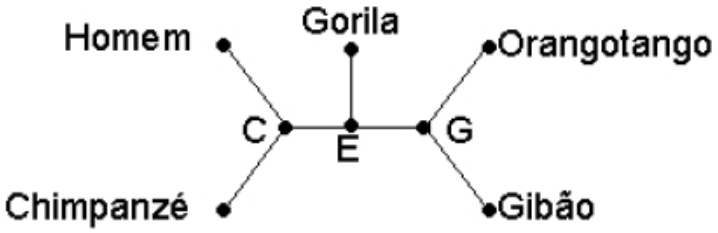


Figura 12. Árvore obtida pelo método de neighbor-joining, as espécies com apenas um nó interno são considerados vizinhos (Weir; 1996).

Este método inicia com uma estrutura em forma de estrela onde todos os elementos estão ligados em um ponto único, figura 13a. Deste grupo, procura-se o par de vizinhos mais próximos, isto é, o que tem a menor distância entre si, e que irá produzir a árvore com menor comprimento total. Estes elementos são unidos formando uma nova unidade, figura 13b. Este procedimento deve ser repetido até que existam dois elementos combinados na estrutura inicial (WEIR, 1996). A figura 13c apresenta

os passos seguintes necessários para o agrupamento dos demais indíviduos, para este agrupamento são considerados como vizinhos mais próximos os seguintes pares: *E* com *D*, *C* com *B* e *A* com *H*.

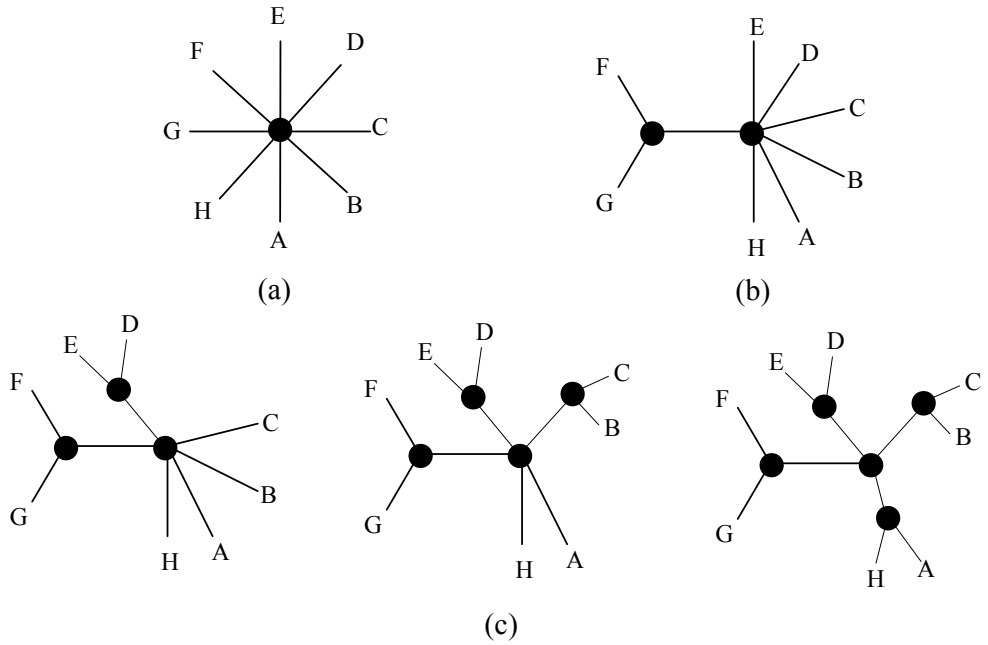


Figura 13. Método *neighbor-joining* de construção de árvores: (a) Início do processo onde todas as espécies estão ligadas por um ancestral comum; (b) agrupamento isolado das duas espécies mais próximas com separação de ancestral; (c) passos necessários para reconstruir a árvore segundo o método *neighbor-joining*

3.2.2 Máxima Parcimônia

A máxima parcimônia é um método numérico para reconstrução de árvores filogenéticas a partir de seqüências moleculares.

O método da máxima parcimônia se baseia em um princípio simples da ciência enunciado por Ockham que diz que as hipóteses mais simples devem ser preferidas em detrimento das mais complexas.

A aplicação deste conceito para a reconstrução de árvores filogenéticas consiste na análise de substituições de base para cada uma das topologias que a árvore pode assumir. O número total de alterações obtida em uma árvore é declarado como o valor de parcimônia da árvore. Deve-se encontrar a topologia e a seqüência de substituições que minimizem o valor de parcimônia. A árvore obtida desta maneira é conhecida

como árvore mais parcimoniosa. Se ocorrer mais de uma árvore mais parcimoniosa deve-se optar por uma das árvores usando algum critério de desempate (SWOFFORD 1996).

Para ilustrar um exemplo de máxima parcimônia, considere as seqüências apresentadas na figura 14a. Dada uma árvore qualquer, como a apresentada na figura 14b, é necessário verificar o menor número de alterações que gere essa árvore. Este processo deve ser repetido para cada árvore candidata que se deseja avaliar. A árvore que conter o menor número de alterações é considerada a árvore mais parcimoniosa.

Espécie 1	ACGACTACGTA
Espécie 2	CGCGATCGATA
Espécie 3	ACGTTACGACT
Espécie 4	TTGACACTGTA

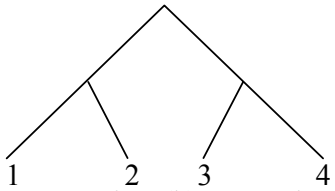


Figura 14. (a) Seqüências de bases para quatro espécies; (b) e uma das topologias possível para a árvore correspondente

Em cada nó ancestral da árvore é necessário calcular a parcimônia dos descendentes. A parcimônia pode ser obtida através de uma operação de conjunto definida como a intersecção de dois elementos, desde que esta intersecção não resulte um conjunto vazio. Neste caso, o resultado é a união entre os dois conjuntos. Para demonstrar esta operação na figura 15, será utilizada a primeira base das espécies sobre a topologia, apresentada na figura 14.

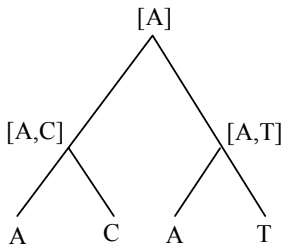


Figura 15. Executando o método de parcimônia no reconhecimento do descendente de uma base

A partir do reconhecimento da base primitiva, que está na raiz da topologia, calcula-se o valor de parcimônia em relação aos espécimes descendentes da árvore.

Pode-se definir o problema da máxima parcimônia de uma maneira matemática através da equação 11.

$$L(\tau) = \sum_{k=1}^{B} \sum_{j=1}^{N} w_j * diff(x_{k'j}, x_{k''j}) \quad (11)$$

No qual deseja-se minimizar *L* da árvore τ onde *B* é o número de ramos e *N* é o número de bases, *k'* e *k''* são nós incidentes de cada ramo, os valores de *x* correspondem à matriz de troca de bases e *diff(y,z)* é a função que define o custo desta troca. O coeficiente w_j apenas associa um peso a cada caractere e foi apresentado por SWOFFORD (1996).

3.2.3 Máxima Verossimilhança

Os métodos baseados no princípio da máxima verossimilhança são os que têm maior custo computacional, porém tendem a apresentar árvores com melhor qualidade. Isto se deve ao fato deste método utilizar modelos de evolução sobre as seqüências disponíveis para avaliar a construção de uma árvore modelo.

O método da máxima verossimilhança se baseia em modelos probabilísticos da evolução, utilizando-se para isto a probabilidade de substituição de bases em cada posição.

A implementação consiste em supor uma topologia para a árvore e calcular o comprimento dos ramos de forma a maximizar a probabilidade de ocorrência daquela árvore na evolução das diversas espécies. Comparam-se todas topologias com o intuito de obter a árvore e o tamanho de ramos que maximize o valor de probabilidade encontrado sendo considerada esta a árvore mais verossímil (NEI e KUMAR, 2000).

O método tem um alto custo computacional por avaliar a possibilidade de mutação de todas as bases contidas na seqüência e a substituição dos diversos nós internos.

Como exemplo do método, vamos analisar a árvore apresenta na figura 16. Os nós externos são apresentados com as bases G, C e T, porém não existe ainda um modelo para os nós internos da árvore. Desta forma, o método da máxima verossimilhança analisará a hipótese de substituição por cada uma das bases, utilizando para isso um modelo de substituição de base.

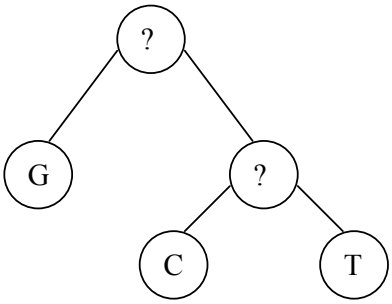


Figura 16. Árvore a ser analisada pelo método da máxima verossimilhança

Dada a árvore acima, se deve atribuir a probabilidade de cada evento, ou seja, deve-se levar em consideração a probabilidade da base A sofrer mutação para as bases C, T ou G ou continuar a mesma e assim por diante, como apresentado na figura 17.

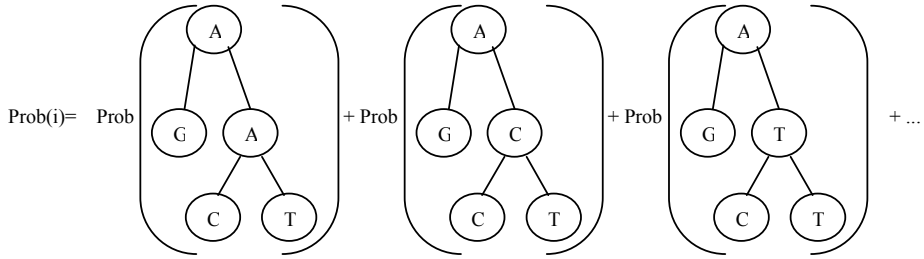


Figura 17. Cálculo da probabilidade no método da máxima verossimilhança

Desta forma, o cálculo da probabilidade da árvore é apresentado na equação 12.

$$P(i) = \sum_{k=1}^{4}\sum_{l=1}^{4}\sum_{z=1}^{4}\pi_k\, p_{kl}(v_i)\, p_{lj}(v_j)..p_{zn}(v_n) \qquad (12)$$

Onde,

P(i): é a verossimilhança da árvore;

π_k: é o fator relativo à freqüência das bases;

$p_{ij}(t)$: é a probabilidade da base *i* mudar para a base *j* no tempo *t*;

p_{ii}: é a probabilidade da base não sofrer alteração;

p_{ij}: é a probabilidade da base i sofrer alteração para base j;

p_i: é o comprimento do ramo.

Como mostrado no exemplo anterior, é necessário um modelo de substituição de bases para cálculo das probabilidades. Estes modelos são gerados levando-se em consideração diversos fatores que se supõe influenciam o processo de divergência entre seqüências.

Existem diversos modelos de substituição, porém cinco são os mais utilizados: JUKES-CANTOR (1969), figura 18a, KIMURA (1980), figura 18b, FELSENSTEIN (1981), figura 18c, HASEGAWA, KISINO E YANO (1985) e TAMURA E NEI (1993).

$$\begin{bmatrix} -3\alpha & \alpha & \alpha & \alpha \\ \alpha & -3\alpha & \alpha & \alpha \\ \alpha & \alpha & -3\alpha & \alpha \\ \alpha & \alpha & \alpha & -3\alpha \end{bmatrix} \qquad \begin{bmatrix} -\alpha-2\beta & \beta & \alpha & \beta \\ \beta & -\alpha-2\beta & \beta & \alpha \\ \alpha & \beta & -\alpha-2\beta & \beta \\ \beta & \alpha & \beta & -\alpha-2\beta \end{bmatrix}$$

(a) (b)

$$\begin{bmatrix} - & \mu\pi_C & \mu\pi_G(1+K/\pi_R) & \mu\pi_T \\ \mu\pi_A & - & \mu\pi_G & \mu\pi_T(1+K/\pi_Y) \\ \mu\pi_A(1+K/\pi_R) & \pi_C & - & \mu\pi_T \\ \mu\pi_A & \mu\pi_C(1+K/\pi_Y) & \mu\pi_G & - \end{bmatrix}$$

(c)

Figura 18. Matrizes de substituição de bases: (a) Jukes-Cantor; (b) Kimura e (c) Felsenstein

Esses modelos diferem pelo número de tipos de substituição e se todas as bases terão ou não a mesma freqüência. Além disto, estão aninhados em uma hierarquia de complexidade, o que significa que de um modelo mais complexo é possível obter os modelos mais simples através do resumo de alguns parâmetros.

CAPÍTULO 4

PROTEÍNAS

As proteínas são estruturas vitais para os seres vivos, desempenhando diversas funções no organismo. Quimicamente, uma proteína é um polímero composto de aminoácidos (STANSFIELD, 1985), isto é, uma molécula composta de um grande número de subunidades comuns entre si.

Os aminoácidos que podem compor uma proteína são conhecidos como aminoácidos primários. Esta distinção se faz necessária por existirem aminoácidos que não pertencem a nenhuma proteína conhecida. Basicamente, são apenas 20 aminoácidos essenciais que compõem as proteínas conhecidas.

Os aminoácidos se caracterizam pela presença de um carbono central, conhecido como carbono alfa (Cα), ao qual estão ligados um hidrogênio, um grupo amina (NH_2), um grupo carboxila (COOH) e um quarto composto conhecido como cadeia lateral. Este composto é o que diferencia os diversos aminoácidos, podendo variar na sua estrutura, tamanho, carga elétrica e solubilidade em água (LEHNINGER, 1991). Na figura 19 é apresentada a estrutura básica de um aminoácido.

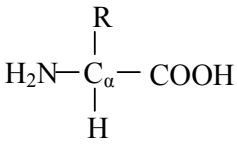


Figura 19. Estrutura de um aminoácido, onde 'R' representa a cadeia lateral

Os aminoácidos se unem em longas cadeias através de ligações peptídicas. Nestas ligações o grupo carboxila do primeiro aminoácido perde a hidroxila (OH) e o grupamento amino do segundo perde um hidrogênio (H). O resultado desta união é uma molécula de água e um dipeptídeo. Na figura 20, é apresentada esta ligação química.

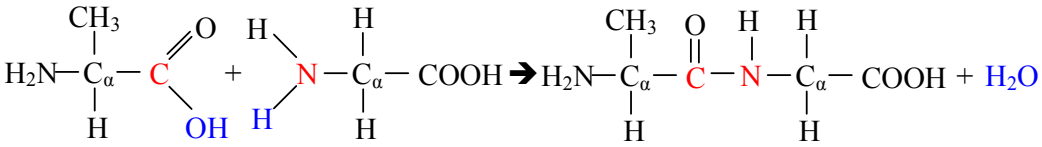


Figura 20. Ligação peptídica entre dois aminoácidos

Normalmente, uma proteína é formada por um grande número de ligações peptídicas. Desta forma ela é conhecida como uma cadeia polipeptídica (AMABIS e MARTHO, 1990).

Como a estrutura central dos aminoácidos se mantém nestas ligações, pode-se verificar que uma proteína é composta por uma seqüência que se repete regularmente, chamada de cadeia principal, e uma seqüência que depende dos aminoácidos que estão ligados, chamada de cadeia lateral.

4.1 Estrutura de proteínas

Como já mencionado, as proteínas exercem atividades vitais para os seres vivos. Porém, uma proteína apenas consegue realizar a função a qual é destina se ela estiver em uma forma tridimensional correta. Esta forma é dependente da seqüência de aminoácidos que a compõem e das suas interações entre si e com o meio no momento de sua formação.

Os aminoácidos são sintetizados a partir de alguns pedaços da seqüência de DNA, conhecidos como genes. Para cada três bases sucessivas do DNA, o ribossomo transcreve um aminoácido diferente. Porém como temos apenas 20 aminoácidos essenciais e como com as quatro bases do DNA agrupadas em conjunto de três em três é possível se gerar 64 combinações, existem combinações que geram a mesma proteína, por esse motivo diz-se que o código genético é degenerado. Porém, esse processo visa diminuir a probabilidade de um aminoácido ser codificado diferente pela alteração de apenas uma base do DNA.

Desde o momento de sua transcrição pelo ribossomo até a sua ativação, a estrutura de uma proteína passa por uma série de transformações.

A estrutura de uma proteína é definida em quatro níveis: estrutura primária, secundária, terciária e quaternária, como mostrado na figura 21.

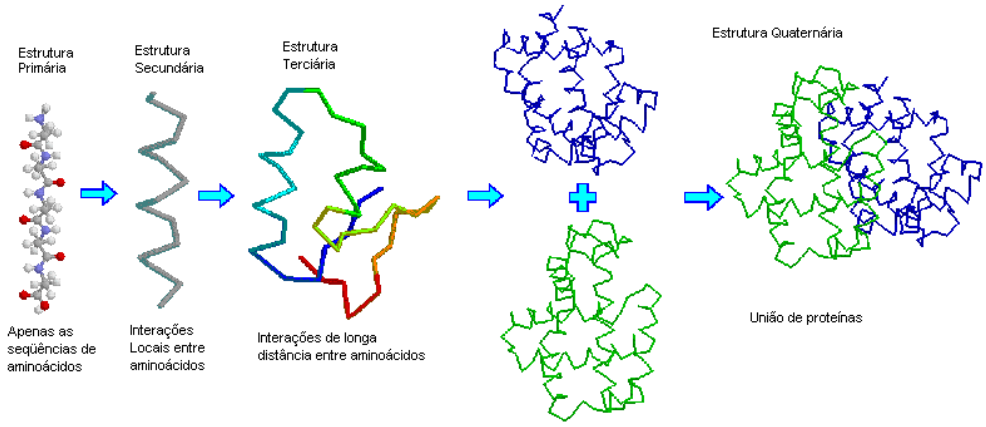


Figura 21. Os quatro níveis de estruturas de uma proteína

A estrutura primária corresponde à seqüência linear dos aminoácidos, estas seqüências podem variar de 40 aminoácidos até milhares de aminoácidos. Este nível é o mais simples e o mais importante, pois as interações dos aminoácidos dependerão de sua composição. Neste nível a estrutura de duas proteínas pode diferir em três características: tamanho da seqüência, tipo dos aminoácidos e a seqüência dos aminoácidos que as compõem.

A estrutura secundária é composta pelo arranjo espacial de pequenos segmentos locais da cadeia polipeptídica. Este é o primeiro nível de estrutura a iniciar realmente o processo de dobramento. Este arranjo ocorre devido à possibilidade de rotação das ligações do Cα com os grupos amina e carboxila dos resíduos e das atrações que ocorrem entre as moléculas. Estes arranjos se repetem ao longo de segmentos da molécula e podem ser classificados em dois grupos principais: α-hélices e folhas-β. Na figura 22 são apresentadas as formas destas duas estruturas.

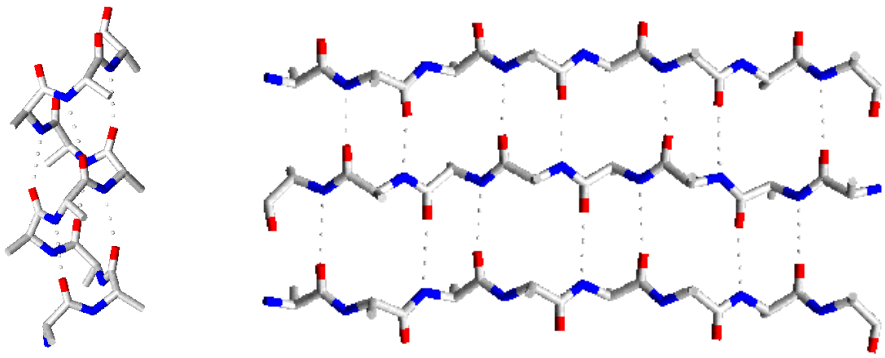


Figura 22. Exemplos de α-hélices e folhas-β

As α-hélices são as formas mais comuns de estruturas secundárias regulares. São enrolamentos do esqueleto polipeptídico em torno do eixo de uma hélice imaginária. A estrutura é estabilizada por pontes de hidrogênio entre os grupamentos NH e CO da cadeia principal. O grupamento CO de cada aminoácido forma uma ponte de hidrogênio com o grupamento NH do aminoácido que está situado a quatro unidades adiante na seqüência linear, sendo que todos os grupamentos NH e CO formam pontes de hidrogênio (STANSFIELD, 1985).

As folhas-β envolvem dois ou mais segmentos polipeptídicos da mesma molécula, ou de moléculas diferentes, arranjados em paralelo (quando as cadeias adjacentes se estenderem em um mesmo sentido) ou antiparalelo (quando se estenderem em sentidos opostos). São estabilizadas por pontes de hidrogênio entre grupamentos NH e CO em fitas peptídicas diferentes, ao contrário da α-hélice cujas pontes de hidrogênio estão entre grupamentos do mesmo filamento. As cadeias laterais se alternam para cima e para baixo ao longo do esqueleto estendido.

Apesar das α-hélices e folhas-β serem bastante comuns, apenas 50% da proteína assume uma destas duas formas, sendo o restante constituído de seqüências não repetitivas e bastante irregulares comparadas com as duas citadas.

A estrutura terciária está relacionada com a forma estrutural completa da proteína, esta forma é dada pelo dobramento global de toda a estrutura avaliando-se as interações de todos os aminoácidos que compõem a cadeia.

O que diferencia a estrutura terciária da secundária é a avaliação global das interações, o que faz com que cadeias hidrofóbicas longas perturbem a estrutura secundária e as curvem para dentro da proteína procurando se proteger de interações com o meio aquoso, enquanto que as partes hidrofílicas ficam expostas ao meio.

Por último, a estrutura quaternária é composta pelo agrupamento de duas ou mais proteínas para formar a sua estrutura total, visto que apenas uma cadeia polipeptídica não contém a estrutura completa da proteína.

Como apresentado no início desta seção, acredita-se que a função de uma proteína é intimamente correlacionada com a sua estrutura tridimensional. Assim, o conhecimento do processo pelo qual passa uma proteína desde sua estrutura primária até a sua estrutura quaternária faz com que se descubra como as proteínas funcionam. Além disto, é o primeiro passo para corrigir eventuais anomalias neste processo ou até mesmo criar novas moléculas a partir de funções que se deseja que ela execute no organismo.

4.2 Dobramento de proteínas

As proteínas ao serem transcritas no ribossomo se encontram em um estado de alta energia livre, sendo facilmente atraídas por outras moléculas da própria fita. Assim, surge o processo de *folding* (dobramento) que ao final fará com que a proteína possa exercer sua função específica. O dobramento de proteínas é o processo pelo qual a informação linear contida na seqüência de aminoácidos de um polipeptídio dá origem à conformação tridimensional bem definida da proteína funcional (HARTL, 1996).

Quando a proteína executa seu dobramento em condições fisiológicas normais ela encontra sempre uma estrutura tridimensional única, conhecida como estrutura nativa. Neste estado, ela desempenha suas funções biológicas normais. As proteínas tendem espontaneamente para esta organização através das estruturas apresentadas anteriormente procurando o mais alto grau de organização e eficiência na utilização de energia (HENEINE, 1984).

No processo de dobramento, as proteínas obedecem as mais fundamentais leis da Física, especialmente a lei da termodinâmica que diz que uma biomolécula passa a maior parte de sua vida em um estado de menor energia livre (PEDERSEN, 2000). Acredita-se que o estado nativo de uma proteína é justamente o estado no qual ela tem o menor coeficiente de energia livre. Esta hipótese conhecida como Hipótese da Termodinâmica, foi apresentada em PEDERSEN (2000).

Christian Anfisen demonstrou, em 1961, que uma proteína pode ser desdobrada até a sua estrutura primária através de uma alteração no meio no qual está inserida. Ao retornar gradativamente ao estado normal do meio a proteína retornará ao seu estado nativo. Porém, isto só é válido para proteínas pequenas que tenham domínio-único (COUNSELL, 2004). As proteínas normais têm um comprimento grande e múltiplos domínios.

DOBSON *et al.* (1994) e CSERMELY *et al.* (2003) mostram que no redobramento espontâneo e sem assistência as proteínas de domínio-único parecem seguir a seguinte ordem de tarefas:

- **Formação da estrutura secundária**: Este estado parcialmente dobrado caracteriza-se por possuir sua estrutura secundária bem desenvolvida. Sua estrutura pode ser parecida com a da proteína nativa apesar da persistência das interações ser ainda bem limitada. Os resíduos

hidrofóbicos já iniciam o processo de localização próximo ao interior da molécula;

- **Estabilização da estrutura secundária:** As estruturas secundárias iniciam as interações entre si para mútua estabilização. Algumas estruturas terciárias já começam a surgir, porém com locais de melhoria posterior;
- **Múltiplos caminhos**: Nem sempre as interações entre moléculas ocorrem da mesma forma. Não existe razão para pensar em uma forma de dobramento única. Quando redobradas mais rapidamente as moléculas tendem a parecer mais com a forma original enquanto que com dobramentos lentos tendem a aumentar a incidência de erros;
- **Estrutura nativa**: Este passo é o mais lento do dobramento. Nele as cadeias laterais são compactadas, diminuindo a forma da molécula final. Quanto mais próximo da estrutura nativa da proteína, mais lento é o seu dobramento, por não haver mais tantos graus de liberdade nas moléculas e por estar em um estado de baixa energia.

A figura 23 ilustra o processo de redobramento.

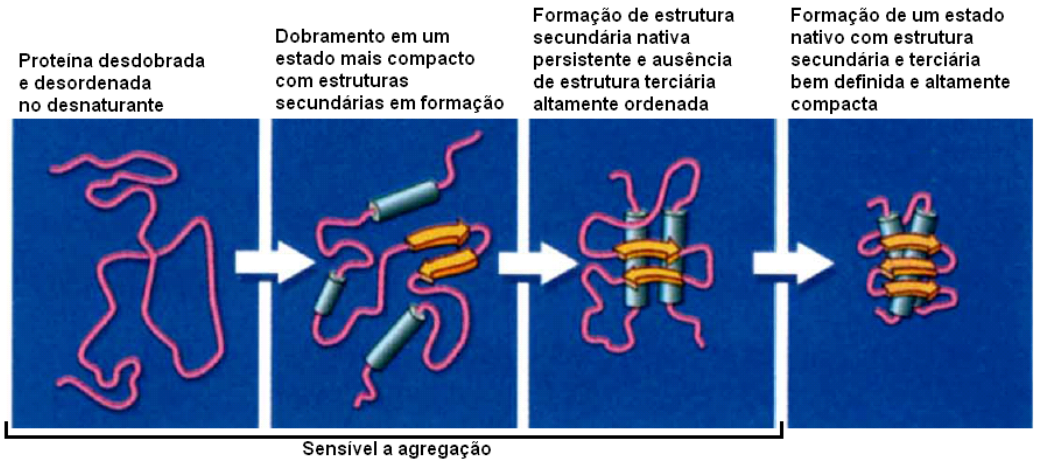


Figura 23. Processo de redobramento de uma proteína. Adaptada de (HARTL, 1996).

Porém, o ambiente intracelular é diferente do ambiente de laboratório onde esses experimentos foram realizados, sendo que o primeiro contém problemas que não ocorrem em um ambiente controlado como um tubo de ensaio. Outro detalhe é que o início da fita transcrita já se dobraria espontaneamente a partir do momento de sua

síntese no citoplasma. Este fato poderia ocasionar um pré-dobramento incorreto, outros problemas seriam a interação desta fita com o meio ou a agregação com cadeias próximas a elas.

Considerando proteínas com mais de um domínio, estas teriam seus domínios dobrados independentemente no início e após ter-se-ia a interação interdomínios, conforme a figura 24.

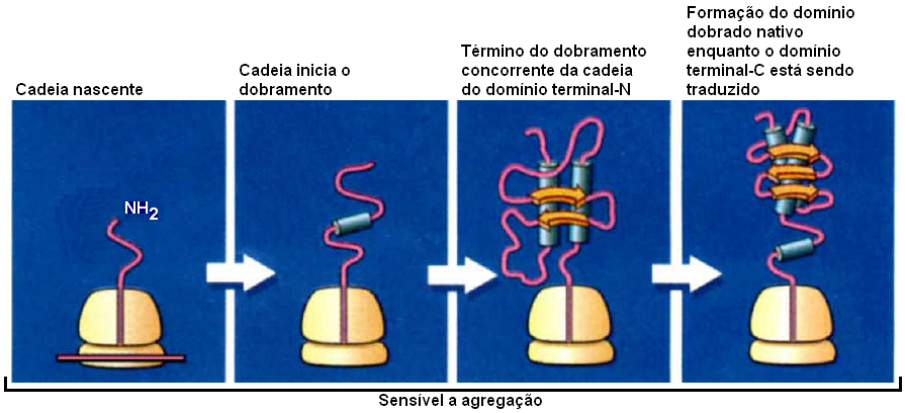


Figura 24. Exemplo de dobramento de uma proteína com dois domínios. Adaptada de (HARTL, 1996).

A consideração não leva em conta que o ambiente celular é composto por diversas moléculas, o que poderia ocasionar que, ao invés da proteína que está sendo sintetizada se dobre, ela se agregue a outras moléculas formando estruturas não funcionais (ELLIS e HARTL, 1999).

Para prevenir os problemas apresentados o organismo desenvolveu mecanismos que retardam o dobramento da proteína, através da proteção das estruturas hidrofóbicas expostas, até que a cadeia tenha um tamanho razoável para o dobramento. Estes mecanismos são conhecidos como chaperonas moleculares ou chaperoninas.

As chaperoninas são grandes complexos cilíndricos, como apresentado na figura 25, que promovem o dobramento de proteínas no ambiente de sua cavidade central. Elas são proteínas que se acoplam à proteína com estrutura ainda instável e buscam estabilizá-la através de um processo controlado, não fazendo parte da estrutura funcional final da proteína.

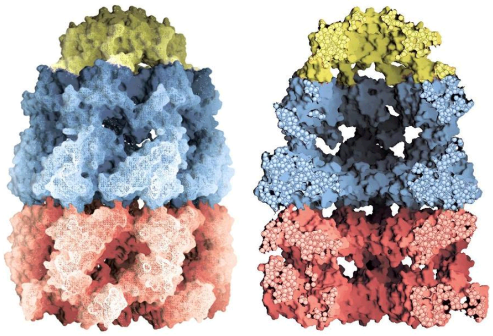


Figura 25. Estrutura de uma chaperonina

As chaperoninas apenas previnem que interações incorretas, tanto internas (entre resíduos) quanto externas (com outras proteínas), ocorram enquanto a proteína não alcançou sua conformação nativa, evitando que a mesma atinja uma conformação não-nativa (HARTL, 1996).

As chaperoninas também ajudam o sistema proteolítico intracelular na tarefa de destruir proteínas mal dobradas.

Quando há um alto nível de geração de proteínas mal-dobradas, a capacidade do sistema proteolítico e das chaperoninas pode se exceder, gerando mal funcionamento celular. Proteínas mal-dobradas podem adquirir uma função totalmente diferente ou ainda prejudicar as células ao seu redor (THOMASSON, 2001; MOGK e BUKAU, 2004).

Cada vez mais são descobertas doenças que estão relacionadas, direta ou indiretamente com proteínas mal dobradas, dentre elas o Mal de Alzheimer, Mal de Parkinson, Diabetes tipo II, Vaca-Louca, vários tipos de câncer e uma quantidade de outras doenças. Estas doenças podem ser esporádicas, herdadas ou, até mesmo, doenças infecciosas, e freqüentemente se manifestam somente em um estágio avançado da vida.

Por estes motivos, torna-se interessante algum método de predição do dobramento dessas estruturas.

4.3 O problema do dobramento

Como mencionado, as proteínas desempenham um papel fundamental no funcionamento dos seres vivos. Também foi apresentado que a funcionalidade de uma proteína está diretamente correlacionada com a sua forma física. Devido a estes fatores,

esforços têm sido dedicados à descoberta de como o intricado processo de dobramento ocorre na célula. Estes estudos visam determinar uma maneira de encontrar a forma nativa de uma seqüência protéica e com isto predizer sua função no organismo.

Um fator importante a ser analisado quando do estudo do dobramento é o "Paradoxo de Levinthal" o qual diz que uma proteína dobra-se em um espaço de tempo relativamente curto, no máximo poucos segundos, porém o espaço de busca de todas as soluções possíveis é imenso, não permitindo o uso de um método que avalie todas as possíveis conformações para uma dada seqüência para a busca da conformação ideal, semelhante a um método de força bruta.

Atualmente existem dois métodos que permitem determinar corretamente a estrutura tridimensional de uma proteína: a cristalografia por Raios-X e a Ressonância Magnética Nuclear. Porém, ambos os processos são muitos custosos, do ponto de vista financeiro e de tempo para sua realização e confirmação. Desta forma, existem poucas proteínas com sua forma tridimensional nativa conhecida através destes métodos.

Atualmente, vários estudos visam prover uma abordagem teórico/prática que permita através da estrutura primária de uma proteína descobrir o dobramento correto até a sua conformação nativa (GUEX *et al*., 1999; LYNGSØ e PEDERSEN, 2000).

Para prever computacionalmente a estrutura de uma proteína é preciso desenvolver um modelo que abstraia as informações necessárias para o dobramento em um nível de detalhamento que não seja demasiadamente custoso computacionalmente. Além disto, o modelo deve ser factível do ponto de vista físico e químico.

Como dito na seção anterior, verificou-se que o dobramento de proteínas é baseado nas leis da termodinâmica, mais precisamente na busca da conformação nativa através da minimização da energia livre. Segundo PEDERSEN (2000), um modelo baseado neste princípio, conhecido como modelo de energia livre, deve incluir os seguintes itens:

- Uma abstração das diversas moléculas e suas ligações;
- Um conjunto de regras que definem quais conformações as moléculas podem formar;
- Uma função de energia livre que permita definir quão boa é a conformação obtida.

Além disto, um modelo para ser considerado relevante deve obter conformações que tenham uma equivalência visual com as conformações reais tanto na sua estrutura

terciária quanto em suas estruturas secundárias. Outro fato, é que o modelo tenha uma equivalência de ações com o sistema real modelado.

4.3.1 Modelos de energia livre

Os modelos de energia livre atuais podem ser divididos em dois grupos distintos: analíticos e discretos. Esses dois modelos serão descritos na seqüência.

A figura 26 ilustra alguns modelos de energia livre com diferentes níveis de abstração.

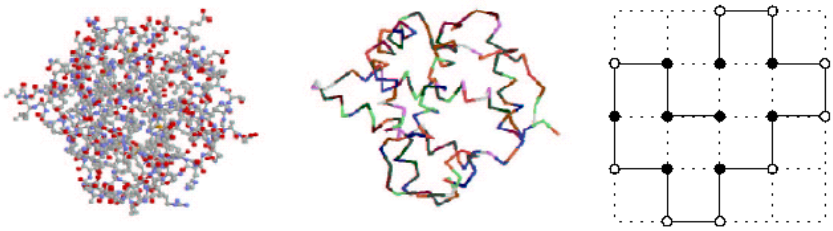


Figura 26. Diferentes níveis de detalhes de uma proteína (PEDERSEN, 2000).

4.3.3.1 Modelos Analíticos

Os modelos analíticos utilizam uma descrição detalhada das características e estados de cada um dos átomos que compõem as diversas moléculas pertencentes a uma proteína. Este modelo foi inicialmente proposto por PEDERSEN (2000).

O modelo visa descrever a proteína através das características elétricas, ângulo e torção de ligação de cada átomo. Desta forma, é necessário especificar um valor de ângulo, um valor de comprimento e um valor de torção para cada ligação atômica da estrutura, obtendo, assim, uma configuração espacial perfeita das proteínas.

Porém, este modelo tem um custo computacional enorme, mesmo para seqüências muito pequenas. Uma forma de diminuir a complexidade é agrupar átomos que tenham posicionamento comum ou que não façam grandes alterações na cadeia final. No entanto, estas reduções diminuem o nível de detalhes do modelo, acarretando uma diminuição na equivalência visual entre a conformação do modelo e a conformação nativa.

A avaliação da função de energia livre em um modelo analítico normalmente é feita somando-se as contribuições relativas das ligações de cada átomo e dos átomos

não ligados. Para as ligações atômicas, a função de energia livre tem como parâmetros os valores de ângulo, comprimento e torção da ligação. Para os átomos não ligados são utilizados os princípios físicos da eletro-atração e eletro-repulsão (forças de Coulomb, van der Walls, etc) ou informações estatísticas inferidas de estruturas conhecidas.

Como se pode verificar, no modelo baseado na descrição detalhada da estrutura da proteína e os muitos parâmetros da função de energia livre, a solução para o problema de predição de estrutura utilizando um modelo como este é computacionalmente muito custosa.

Um exemplo de modelo analítico foi proposto por NGO e MARKS (1992) e apresentado mais detalhadamente em NGO *et al*., (1994) e CHANDRU *et al.* (2003).

4.3.3.2 Modelos Discretos

A utilização de modelos analíticos para obtenção de estruturas protéicas é quase impossível, tendo em vista que o número de conformações possíveis é imenso e por necessitarem de um nível de detalhamento muito alto.

Desta forma, foram propostos modelos mais simples, diminuindo o detalhamento e aumentando o número de conformações analisadas, obtendo, assim, resultados encorajadores (DINNER *et al*., 2000; DUAN e KOLLMAN, 2001; CHANDRU *et al*., 2003). Estes modelos simplificados, chamados discretos, apresentam uma equivalência visual ainda mais baixa do que o modelo analítico simplificado, porém ainda mantêm a equivalência comportamental.

Outro problema comum a todos os modelos é o tamanho das cadeias polipeptídicas tratadas, que são muito pequenas, contrastando com as proteínas reais com grande número de moléculas.

Os modelos discretos são comumente chamados de modelos treliça por basearem a posição dos aminoácidos em uma grade do tipo treliça. Os modelos treliça se tornaram populares por permitirem uma simulação simples da conformação das proteínas, além de permitirem um estudo do seu funcionamento comportamental. Simulações em modelos analíticos envolvem muitos parâmetros e aproximações, tornando sua validade tão duvidosa quanto para modelos discretos (CHANDRU *et al*., 2003).

Apesar de todas estas simplificações que compõem os modelos treliça, UNGER e MOULT (1993), CRESCENZI *et al*. (1998) e NAYAK *et al*. (1998), entre outros,

provaram que até mesmo neste modelo o dobramento de proteínas leva a um problema NP-completo.

4.3.3.3 Modelo HP (*Hydrophobic – Polar*)

O modelo HP proposto por DILL (1985) é o modelo discreto mais simples e mais estudado, sendo o modelo mais popular de grade quadrada. Ele se baseia na hipótese de que a maior contribuição para a função de energia livre vem das ligações entre aminoácidos hidrofóbicos quando estes tendem a buscar o centro da conformação devido a sua repulsão a ligações com moléculas de água abundantes na célula humana.

O modelo HP utiliza os 20 aminoácidos primários classificados em dois sub-grupos apenas: hidrofóbicos e hidrofílicos, sem levar em consideração o nível destes. Desta forma, uma proteína é uma seqüência composta por um alfabeto de apenas duas letras {H,P}. A seqüência é posicionada sobre uma grade do tipo treliça quadrada ou triangular, sendo esta de duas ou três dimensões, recebendo os respectivos nomes de 2D HP e 3D HP.

A função de energia livre do modelo HP se baseia no número de ligações não-locais. Uma ligação não-local é composta por um par de aminoácidos que não são adjacentes na seqüência inicial, porém estão em posições adjacentes na grade.

A função de energia livre, sugerida por LI *et al*. (1996), é representada na equação 13:

$$E = \sum_{i<j} e_{v_i v_j} \Delta(r_i - r_j) \tag{13}$$

onde $\Delta(r_i - r_j) = 1$ se os resíduos r_i e r_j formam uma ligação não-local e $\Delta(r_t - r_j) = 0$ em caso contrário. Dependendo dos tipos de contatos entre os resíduos, a energia $e_{v_i v_j}$ será e_{HH}, e_{HP} ou e_{PP}, correspondendo a contatos H–H, H–P ou P–P, respectivamente. Os três tipos de ligações estão apresentados na figura 27.

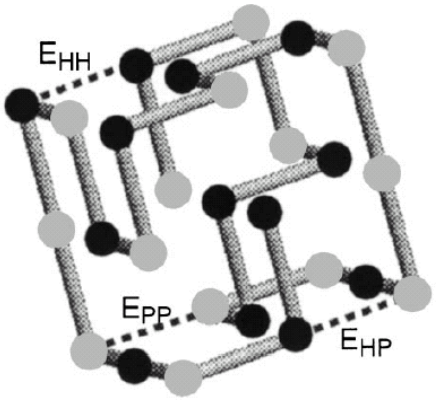


Figura 27. Modelo HP com os três tipos de ligações especificados

De acordo com LI *et al*. (1996), o modelo de energia proposto satisfaz as seguintes limitações físicas:

a) Formas compactas possuem energias menores que qualquer forma não-compacta;
b) Aminoácidos H ficam localizados no interior o máximo possível, expressado pela relação $e_{PP} > e_{HP} > e_{HH}$, que diminui a energia de configurações nas quais os Hs estão escondidos da água;
c) Tipos diferentes de aminoácidos tendem a ter menos ligações entre si do que tipos iguais, expressado pela relação $2e_{HP} > e_{PP} + e_{HH}$.

Mesmo com uma simplificação tão extensa comparado com o modelo analítico, este modelo também leva a um problema NP-Completo. Isto foi demonstrado por CRESCENZI *et al.* (1998) no modelo 2D e BERGER e LEIGHTON (1998) no modelo 3D.

Diversas generalizações já foram efetuadas neste modelo, como por exemplo: alteração dos valores de energia, alteração da função de energia, forma e tamanho da grade. Por exemplo, em AGARWALA *et al*. (1997) utilizou-se uma grade triangular e valores de energia para cada aminoácido baseados em sua hidrofobicidade.

4.4 Abordagens para o dobramento

Mesmo com o modelo mais simples, 2D HP, o problema do dobramento de proteínas ainda é NP-Completo. Posto isto, é necessário o desenvolvimento de técnicas computacionais eficientes para a busca de conformações ótimas de proteínas, pois uma

busca exaustiva tem tempo de processamento inviável à medida que aumenta o tamanho das seqüências. Diversos métodos computacionais já foram utilizados: algoritmos de aproximação, *build-up*, algoritmos de computação evolucionária e algoritmos de otimização por colônia de formigas.

Na seqüência, serão abordados alguns métodos computacionais para o estudo do dobramento de proteínas.

4.4.1 Dinâmica molecular

Este método tem como idéia básica simular os movimentos de cada átomo da proteína e das moléculas de água que a rodeia como uma função do tempo. É fornecida a energia térmica inicial ao sistema e os átomos podem se mover de acordo com regras da mecânica clássica, sendo consideradas todas as forças exercidas.

Para tornar o movimento realista, é necessário que o tempo contínuo seja discretizado no menor espaço possível, sendo considerados intervalos de 10^{-15} segundos. Isto faz com que o algoritmo só simule os 10 nanossegundos iniciais de um dobramento. Normalmente são utilizados algoritmos do tipo Monte Carlo para este tipo de simulação.

Os resultados atuais apresentam alta semelhança com os estudos reais. Porém, não existem estudos que comprovem que o sistema chegue até a conformação nativa.

4.4.2 *Build-up*

Este procedimento inicia com um segmento curto e dobra iterativamente os outros segmentos sobre este.

Este processo é bastante robusto do ponto de vista computacional, visto que o número de decisões a serem tomadas pelo algoritmo é pequeno e seu espaço de busca também. Porém, os resultados obtidos com este algoritmo não foram animadores, pois uma conformação ótima local não necessariamente leva a uma conformação ótima global e, como o algoritmo tem um espaço de busca bastante restrito, não é possível a correção de caminhos como estes.

Um exemplo de utilização deste tipo de procedimento foi apresentado por SRINIVASAN e ROSE (2002).

4.4.3 Algoritmos de aproximação

Um algoritmo de aproximação é um algoritmo que encontra soluções próximas da solução ótima. Os primeiros algoritmos de aproximação para o problema da predição de estruturas de proteínas foram formulados no modelo HP por HART e ISTRAIL (1996). Por isto, as considerações feitas aqui se referem a ele.

Num algoritmo de aproximação, procura-se encontrar um dobramento de uma seqüência S cujo escore seja uma fração do escore do dobramento ótimo. Então, torna-se necessário determinar um limite máximo permitido para o escore ótimo, OPT(S), da seqüência S. Para estabelecer um limite superior são necessárias duas observações. A primeira é que um aminoácido hidrofóbico pode formar, no máximo, duas ligações não-locais em uma grade quadrada 2D, exceto para os aminoácidos inicial e final que podem formar três ligações não-locais. A segunda observação é que dois aminoácidos hidrofóbicos, $S[i]$ e $S[j]$, podem ocupar pontos adjacentes na grade quadrada 2D se, e somente se, i é par e j é ímpar ou vice-versa. Definindo-se EVEN(S) e ODD(S), respectivamente, como o conjunto de posições pares e ímpares em S contendo um aminoácido hidrofóbico, tem-se o limite superior conforme a equação 14

$$\mathrm{OPT}(S) \le 2 \cdot \min\left\{\left|\mathrm{EVEN}(S)\right|, \left|\mathrm{ODD}(S)\right|\right\} + 2\,, \tag{14}$$

generalizando, obtém-se a equação 15

$$\mathrm{OPT}(S) \le \left(2d - 2\right) \cdot \min\left\{\left|\mathrm{EVEN}(S)\right|, \left|\mathrm{ODD}(S)\right|\right\} + 2 \tag{15}$$

onde d é a dimensão da grade sendo considerada.

CAPÍTULO 5

TRABALHOS CORRELATOS

Os algoritmos baseados na heurística de colônia de formigas são recentes e têm sido apresentados na solução de diversos problemas. Na seqüência, serão descritos os principais trabalhos conhecidos de utilização desta heurística para a reconstrução de árvores filogenéticas e para o problema de dobramento de proteínas.

5.1 Modelo de reconstrução através do caixeiro viajante

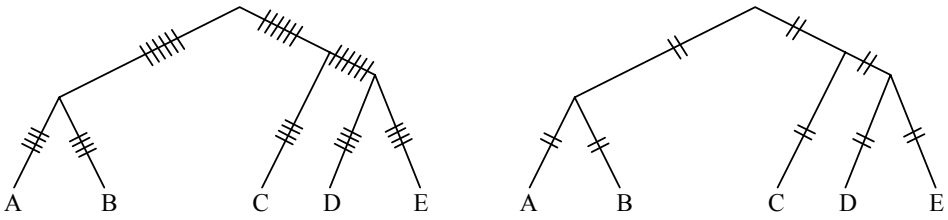
A primeira aplicação da heurística de colônia de formigas apresentada por DORIGO (1991) foi na solução do problema do caixeiro viajante, no qual a analogia entre a busca de recursos pelas formigas e a do menor caminho entre cidades são similares.

No trabalho de KOROSTENSKY e GONNET (2000) não é utilizado o ACO propriamente, porém apresenta-se uma conversão do problema de reconstrução de árvores para uma instância específica do problema do caixeiro viajante, conhecida como o problema do caixeiro viajante circular.

Para isto, uma árvore filogenética é apresentada como um grafo binário acíclico, $T = (V,E)$, onde V são os vértices (nós) e E são os pontos do grafo.

Na execução de um método normal de avaliação de escores direta entre dois nós, a distância referente aos nós internos, ancestrais, é contada mais vezes do que os vértices mais próximos aos descendentes, como apresentado na figura 28a. Não existe justificativa teórica para este fato que faz com que a distância entre ancestrais tenha maior peso sobre a distância entre descendentes. Se considerarmos que os dois nós extremos da árvore são interligados o número de vezes que se necessita percorrer os nós ancestrais para, por exemplo, avaliar a distância entre as espécies “b” e “e” se torna equivalente para todas as espécies da árvore como apresentado na figura 28b.

Com esta consideração, definiu-se que a melhor forma de interpretar esta ligação entre as duas espécies extremas do caminho seria uma forma circular, isto é, deve-se considerar que um caminho inicia-se em um nó qualquer e depois de passado por todos os nós no caminho deve-se retornar ao mesmo nó do início.



(a) (b)

Figura 28. Se avaliarmos um somatório da distância entre espécies, teremos um maior peso na distância entre ancestrais (a). Porém se considerarmos, um caminho circular todos os ramos são percorridos o mesmo número de vezes (b).

O algoritmo retornará uma seqüência que define a ordem de agrupamento das espécies que obterá a menor distância. A partir da seqüência de espécies que produzem a menor distância, é necessário construir uma árvore com ancestrais aleatórios entre as espécies e verificar se a árvore produzida obtém o mesmo escore que a seqüência obtida pelo algoritmo de resolução do caixeiro viajante.

Os resultados obtidos mostraram que o método tem grande consistência, produzindo árvores equivalentes aos principais métodos empregados para a reconstrução de árvores filogenéticas. No entanto, como é necessário um segundo passo de busca da melhor árvore, o método apresentado obteve um tempo de processamento maior que implementações do método de máxima verossimilhança, considerado o mais custoso em tempo computacional.

5.2 Modelo de reconstrução através do problema de Steiner

O problema de Steiner consiste em achar a menor rede interconectada para um grupo finito de pontos em um espaço finito, sendo um dos problemas geométricos combinatoriais mais famosos. Entretanto, o problema de Steiner é extremamente difícil em termos de estrutura combinatorial e complexidade computacional.

No trabalho de KUMNORKAEW, KU e RUENGLERTPANYAKUL (2004), aplica-se uma heurística de colônia de formigas para a resolução de um grafo baseado no problema de Steiner.

Para converter o problema da construção de árvores em um problema de Steiner, foi executado um pré-processamento através de um algoritmo para busca de ancestrais

comuns entre duas espécies arbitrárias. Este passo é executado até que o número de pontos de união seja igual a *n*-2, que é o número de nós internos, ancestrais. Todos os pontos de união são armazenados em vetores.

Na seqüência, é construída uma matriz de conexões entre os elementos e os pontos de união, como apresentado na tabela 1. Esta matriz definirá por quais pontos de união uma formiga deverá passar quando se movimentar de um nó ao outro.

Tabela 1. Rota de reconstrução dos caminhos seguindo os pontos de união

XX	A1'	B1'	1'2'	C2'	2'3'	D3'	E3'
$d_{a,b}$	1	1	0	0	0	0	0
$d_{a,c}$	1	0	1	1	0	0	0
$d_{a,d}$	1	0	1	0	1	1	0
$d_{a,e}$	1	0	1	0	1	0	1
$d_{b,c}$	0	1	1	1	0	0	0
$d_{b,d}$	0	1	1	0	1	1	0
$d_{b,e}$	0	1	1	0	1	0	1
$d_{c,d}$	0	0	0	1	1	1	0
$d_{c,e}$	0	0	0	1	1	0	1
$d_{d,e}$	0	0	0	0	0	1	1

Na tabela 1, é obtido um grafo com os pontos iniciais (nós folhas) interligados aos pontos criados pelo método descrito na figura 29, que serão as espécies ancestrais. As formigas são restritas a visitar apenas um nó em cada zona. Uma zona é criada quando define-se uma ligação entre duas espécies (nós iniciais) e um ancestral (nó criado), evitando desta forma que um ancestral contenha diversas espécies. O número de zonas disponíveis para visitar é igual a (q(q-1)/2) onde q é o número de espécies no início e é decrementado a cada movimento da formiga. Para cada conexão entre as zonas, é armazenada a trilha de feromônios que indicará os melhores caminhos.

O método proposto foi comparado com os métodos de Neighbor-Joining e com o método com TSP apresentado por KOROTENSKY e GONNET (2000), com dois conjuntos de dados diferentes: hemoglobina alpha-I e citocroma C. Estes conjuntos de dados continham 14 espécies.

O método proposto apresentou resultados melhores que o método de Neighbor-Joining e equivalente ao método com TSP. Além disso, o tempo de processamento foi menor que o método com TSP. Porém, maior que o apresentado pelo Neighbor-Joining por necessitar recriar uma série de nós ancestrais, como apresentado na figura 28.

Este trabalho apresentou a viabilidade de utilizar o algoritmo de ACO para a reconstrução de árvores filogenéticas, porém o modelo aqui apresentado não foi utilizado na metodologia desenvolvida.

5.3 Modelo 2D HP para dobramento de proteínas

O primeiro modelo do algoritmo ACO para o problema de dobramento de proteínas foi proposto por SHMYGELSKA, HERNÁNDEZ e HOOS (2000).

No modelo proposto é utilizada como base a esturtura 2D HP no qual o alfabeto de 20 aminoácidos essenciais é reduzido para um alfabeto com apenas duas letras definidas pela classificação dos aminoácidos em duas classes distintas. Assim como na definição original do modelo 2D HP, o posicionamento dos aminoácidos é realizado sobre uma treliça quadrada. Os movimentos sobre a treliça são armazenados em relação ao aminoácido anterior. Desta forma, foi definido três posições possíveis para um aminoácido em relação ao seu antecessor: adiante, esquerda ou direita (KRASNOGOR, PELTA, LOPEZ *et al*, 1998).

Para a movimentação das formigas é necessário definir primeiramente qual o ponto inicial que a formiga terá. Para isso, foi selecionada uma posição aleatória, escolhida em uma distribuição normal, entre 1 e *n-1*, onde *n* é o número de aminoácidos da seqüência. Deste ponto inicial a seqüência é dobrada nas duas direções ao mesmo tempo avaliando a posição de um aminoácido por vez.

A direção relativa na qual o próximo aminoácido será posicionado é selecionada por uma função probabilística baseada em uma informação heurística que será detalhada na seqüência. Também nesta função probabilística é utilizada a informação da trilha de feromônios. Para efetuar uma curva, seleção de direção relativa esquerda ou direita, são avaliados três aminoácidos, o primeiro é o último aminoácido já posicionado (s_{i-1}), o segundo é o aminoácido atual(s_i) e o terceiro é o próximo aminoácido(s_{i+1}), sendo que são posicionados neste movimento os dois últimos, um exemplo é apresentado na figura 29.

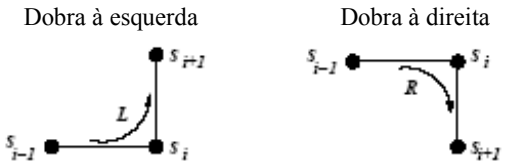


Figura 29. Exemplo de dobramento efetuado no trabalho de SHMYGELSKA, HERNÁNDEZ e HOOS (2000)

Uma vez que o algoritmo utiliza um ponto de início aleatório no centro da seqüência e movimentações relativas em relação ao aminoácido anterior, é necessário avaliar que uma dobra à esquerda, quanto se está avançando em direção ao final da seqüência, é equivalente a uma dobra à direita sobre os mesmos aminoácidos quando se esta indo em direção ao início da seqüência. Isto é demonstrado na figura 30. Por este motivo, foi definido uma relação de simetria sobre a matriz de feromônios dada por $\tau'_{i,L} = \tau_{i,D}$ e $\tau'_{i,D} = \tau_{i,L}$, sendo que τ' é a trilha de feromônios quando a direção da formiga é o início da seqüência e τ é a trilha de feromônios quando a formiga se movimenta em direção ao final da seqüência.

(a) (b)

Figura 30. Exemplo de a direção do movimento é dependente de em qual direção da seqüência se esta indo: (a) do início da seqüência para o final; (b) do final da seqüência para o início

O valor heurístico incorporado à função de probabilidade do movimento é dada pelo somatório de ligações H-H que serão realizadas se o movimento for efetuado. Para os aminoácidos polares é definido um valor heurístico constante de 1, sendo que desta forma seu posicionamento será definido pela trilha de feromônios.

Devido ao processo de construção do dobramento efetuar uma busca local para avaliar a posição do próximo aminoácido da seqüência, é possível ocorrer que todas as próximas posições que serão avaliadas para o movimento já estejam ocupadas por outros aminoácidos. Quando esta situação ocorrer o algoritmo proposto deveria retornar

à metade da seqüência já dobrada e reiniciar o processo de dobramento a partir desta posição.

Como recurso auxiliar para melhorar o desempenho do ACO foram implementados dois métodos de busca local propostos em KRASNOGOR, PELTA, LOPEZ et al (1998).

- O primeiro conhecido como "*macro mutation neighborhood"* altera a conformação em uma subseqüência definida;
- o segundo é uma mutação de apenas 1 ponto, em que apenas um movimento relativo é alterado na seqüência.

Para cada uma das seqüências obtidas pelas formigas, os métodos de busca local são executados como se segue. Inicialmente, o método de *macro mutation* é efetuado selecionando um grupo de aminoácidos com tamanho aleatório na seqüência, neste grupo todos os movimentos são alterados de tal forma que a conformação obtida ainda seja possível. Esta nova conformação é avaliada e se obteve um valor de energia melhor do que o valor da seqüência original esta última é substituída pela nova seqüência. O segundo passo consiste em alterar todos os movimentos da seqüência um de cada vez e a cada alteração a nova seqüência é avaliada.

Estes métodos de busca local foram aperfeiçoados visando simular o efeito que cada alteração de movimento produziriam em toda seqüência. O novo método de busca local foi apresentados em um segundo trabalho (SHMYGELSKA e HOOS, 2003).

Uma comparação entre o método apresentado e o desenvolvido neste trabalho é apresentada na seção 4.2.

CAPÍTULO 6

METODOLOGIA

Segundo DORIGO e STUTZLE (2004), a construção de um modelo através da heurística do ACO passa por três fases distintas:

> 1ª - Modelar o sistema na forma de uma estrutura computacional que permita considerar a movimentação das formigas em busca de uma solução a partir de uma situação inicial pré-existente;
> 2ª - Definir uma função de probabilidade para movimentação das formigas na busca pelo resultado. Esta função de probabilidade deve considerar um valor heurístico baseado em alguma particularidade do problema e em um valor probabilístico obtido durante a execução;
> 3ª - Definir uma função para atualização da matriz de feromônios considerando acréscimos a partir das soluções obtidas e decréscimos baseados no tempo.

Na seqüência, será detalhado o desenvolvimento de dois modelos. Um aplicado ao problema de reconstrução de árvores filogenéticas e outro ao problema do dobramento de proteínas.

6.1 Modelo de ACO aplicado à reconstrução de árvores filogenéticas

6.1.1 Dados de entrada

Atualmente, existem três tipos diferentes de dados de entrada na construção de árvores filogenéticas.

Os métodos baseados em matriz de distância, como o próprio nome diz, utilizam uma matriz quadrada *nxn* onde a célula da linha *i* coluna *j* contém a distância evolutiva entre duas espécies *i* e *j*. Um exemplo desta matriz é apresentado na tabela 2.

Tabela 2. Exemplo de matriz de distâncias

Espécies	**A**	**B**	**C**	**D**
A	0,000	0,199	0,837	0,830
B	0,199	0,000	0,893	0,886
C	0,837	0,893	0,000	0,009
D	0,830	0,886	0,009	0,000

Os métodos de parcimônia podem ser desenvolvidos utilizando-se dois tipos distintos de dados de entrada. O primeiro utilizado na taxonomia clássica, é uma lista de característica e um alfabeto binário informando se aquela espécie tem ou não aquela característica. O segundo método utilizado na biologia molecular leva em consideração as seqüências genômicas alinhadas e utiliza matrizes de substituição para avaliar a troca de bases. Por último, os métodos de máxima verossimilhança têm como entrada a seqüência de bases alinhadas e uma árvore modelo, sendo que este método procura utilizar as matrizes de substituição para avaliar qual seria o tamanho dos ramos no modelo especificado que produziria a árvore com a maior probabilidade de existir, isto é, a árvore mais verossímil.

Os métodos baseados em matrizes de distância são os mais rápidos, e têm uma grande similaridade com um grafo totalmente interconectado, isto pode ser visto no exemplo do método UPGMA apresentado na figura 11, sendo este tipo de entrada escolhido para o algoritmo.

Porém, o maior problema deste tipo de método é como calcular a distância entre espécies de maneira correta. O procedimento mais comum para cálculo de distância é a avaliação direta entre pares de bases das seqüências alinhadas, realizando um somatório das trocas de bases a partir dos valores de uma matriz de substituição. Este método de cálculo tem uma implementação no pacote de programas PHYLIP conhecido como DNADIST. O maior problema deste método é a necessidade de se utilizar seqüências alinhadas o que torna os valores de distância pouco confiáveis quando da utilização de espécies muito distintas ou seqüências de tamanhos diversos.

6.1.2 Cálculo de distância evolutiva

Visando implementar um método que pudesse realizar a reconstrução dos mais diversos tipos de árvores e verificando que não existe um consenso atual para calcular a distância entre seqüências não alinhadas decidiu-se pela implementação de dois métodos distintos para cálculo destas distâncias. Esses métodos foram selecionados por produzirem distâncias simétricas entre espécies, isto é, a distância da espécie *x* para a espécie *y* é igual a distância de *y* para *x*. Diversos outros métodos foram propostos para cálculo de distâncias entre seqüências não-alinhadas, porém a maior parte produzia distância que não eram simétricas.

Primeiramente, uma medida de distância deve satisfazer os axiomas necessários para uma métrica, conforme a equação 16:

$$\begin{cases} d(x,y) > 0 \ \ \forall\, x \neq y & \text{valor positivo} \\ d(x,y) = 0 \Leftrightarrow x = y & \text{anti - reflexividade} \\ d(x,y) = d(y,x) & \text{simetria} \\ d(x,y) \leq d(x,z) + d(z,y) & \text{desigualdade triangular} \end{cases} \tag{16}$$

O primeiro método para cálculo da distância entre espécies, proposto por LI *et al*. (2001), se baseia no fato de que regiões codificantes de espécies próximas têm suas informações gênicas mais correlacionadas.

Seguindo este pressuposto e, dadas duas seqüências *x* e *y*, a distância entre elas pode ser calculada pela equação 17:

$$d(x,y) = 1 - \frac{K(x) - K(x \mid y)}{K(xy)} \tag{17}$$

onde *K(x|y)* é a complexidade condicional de Kolmogorov, apresentada no anexo 3, de *x* dado *y*, isto é, o tamanho do menor programa que gere *x* quando a entrada for *y*; *K(x)* é definido como *K(x|ε)* onde ε é uma seqüência vazia, e *K(xy)* é a seqüência formada pela concatenação das seqüências *x* e *y* (LI *et al.*; 2001).

O segundo método, nomeado similaridade, proposto por CAMPBELL, MRÁZEK e KARLIN (1999), se baseia no fato de que cada genoma tem uma "assinatura" característica definida pela razão entre a freqüência de dinucleotídeos

observada e a freqüência esperada, para um determinado intervalo. Esta medida é conhecida como abundância relativa do dinucleotídeo.

Existem evidências seguras de que este perfil de assinatura genômica é notavelmente estável para o DNA de um determinado organismo (RUSSEL, WALKER, ELTON, 1976). Amostras de DNA (de tamanho maior do que 50 kbases) de diferentes regiões cromossômicas do mesmo genoma têm aproximadamente a mesma assinatura e, ainda, espécies próximas na escala biológica têm assinaturas genômicas mais similares do que espécies mais distantes entre si. Deste ponto de vista, a assinatura genômica de fato é útil para discriminar entre seqüências de diferentes organismos, o que lhe dá uma grande importância em inúmeros problemas de Biologia Molecular e, em especial, em filogenia, permitindo calcular um grau de similaridade entre espécies.

Para calcular a abundância relativa de um dinucleotídeo é utilizada a equação 18:

$$\rho_{XY} = \frac{f_{XY}}{f_X \cdot f_Y} \tag{18}$$

onde: f_{XY} é a freqüência do dinucleotídeo XY no intervalo, f_X e f_Y são as freqüências dos mononucleotídeos X e Y no mesmo intervalo e $X, Y \in \{A,C,G,T\}$. Para realizar este cálculo deve-se concatenar a seqüência com o seu complemento invertido (CAMPBELL, MRÁZEK, KARLIN, 1999).

A distância entre espécies é obtida da média absoluta da diferença entre os 17 possíveis dinucleotídeos das duas espécies σ_1 e σ_2, como apresentado na equação 19.

$$\delta(\sigma_1, \sigma_2) = \frac{1}{16} \cdot \sum_{\substack{X,Y \in \\ [A,C,G,T]}} \left| \rho_{XY}(\sigma_1) - \rho_{XY}(\sigma_2) \right| \tag{19}$$

6.1.3 Modelo do problema

Um problema modelado com ACO normalmente utiliza um grafo no qual as formigas procuram um caminho entre uma fonte de recursos e o ninho e, a partir de sucessivas iterações, um caminho otimizado surge em meio aos múltiplos caminhos

feitos. Assim, é necessário desenvolver um modelo em que os dados de entrada sejam agrupados de forma que possam ser percorridos como um caminho. Uma matriz de distância como a apresentada na tabela 2 pode ser convertida em um grafo totalmente interconectado como apresentado na figura 31 (PERRETTO e LOPES, 2004).

Um grafo como este é a forma mais simples de modelo do problema do caixeiro viajante, sendo conhecidas diversas implementações que resolvem este problema.

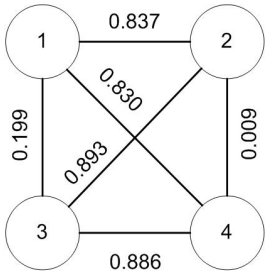


Figura 31. Grafo totalmente interconectado baseado na tabela 2

A partir deste grafo inicial, o algoritmo é modelado como em sua forma original, na qual uma formiga inicia em um nó aleatório no grafo, considerado o ninho, e percorre os demais nós sendo o ponto final (nó fonte) o último nó a ser interconectado.

No modelo do ACO, as formigas selecionam o próximo nó do caminho através de uma função de transição baseada em duas informações. A primeira é a distância entre o nó atual e os demais nós, sendo desconsiderado qualquer nó que já tenha sido percorrido por esta formiga. A segunda informação é a suscetibilidade de se escolher uma trilha que já foi percorrida por outras formigas, isto é quanto mais uma trilha é percorrida pelas formigas maior será a chance das próximas formigas escolherem a mesma trilha. Esta informação é baseada na matriz de feromônios. Na equação 20 é apresentada a função de transição utilizada:

$$P_k(i,j) = \frac{[\tau(i,j)]^{\alpha}.[1/d(i,j)]^{-\beta}}{\sum_{u \in J_i^k} \{ [\tau(i,j)]^{\alpha}.[1/d(i,j)]^{-\beta} \}} \qquad (20)$$

onde: $P_k(i,j)$ é a probabilidade da k-ésima formiga estando no nó *i* seguir para o nó *j*, sendo J_i^k o conjunto de nós conectados ao nó i que não foram visitados pela formiga; $d(i,j)$ é a distância evolucionária entre as espécies representadas pelos nós *i* e *j*; e $\tau(i,j)$ é

a trilha de feromônios representada por uma matriz (como a matriz de distâncias), porém com seus valores sendo alterados de forma dinâmica conforme os caminhos selecionados pelas formigas no grafo. Além disto, *τ(i,j)* representa a atratividade do nó *j* quando a formiga está no nó *i*. α e β são valores arbitrários que definem qual dos dois parâmetros terá maior peso na função de transição.

Como dito anteriormente, este modelo e a função de transição são baseados no modelo de ACO para o problema do caixeiro viajante primeiramente apresentado por COLORNI, DORIGO e MANIEZZO (1991).

Porém, ao contrário do modelo original em que a formiga se dirigia ao próximo nó selecionado e repetia o processo de cálculo das probabilidades de transição neste novo nó, no modelo apresentado aqui um nó intermediário n é criado entre o nó atual e o escolhido para a transição. Este nó intermediário seria a espécie ancestral entre os dois nós i e j e não será adicionada ao conjunto de espécies avaliadas. O objetivo deste nó é ajustar as distâncias entre as duas espécies selecionadas e as demais espécies pertencentes ao grafo. Para isso, as distâncias entre espécies são recalculadas como apresentado na equação 21, onde η é um parâmetro relacionado à proximidade das espécies com o seu ancestral.

$$d_{nu}(i,j) = \begin{cases} d(i,u) + [d(i,u) - d(j,u)].\eta, \\ \qquad \text{se} \quad d(j,u) > d(i,u) \\ d(j,u) + [d(j,u) - d(i,u)].\eta, \\ \qquad se \quad d(i,u) > d(j,u) \end{cases} \tag{21}$$

onde: $d_{nu}(i,j)$ é a distância entre o nó intermediário *n* e um outro nó *u* do grafo; *d(i,u)* é a distância entre os nós *i* e *u*; *d(j,u)* é a distância entre o nós *j* e *u*; e η é um parâmetro de correção ajustável pelo usuário para definir se o nó intermediário está mais próximo do primeiro ou do segundo descendente.

Na figura 32, é apresentado um exemplo da construção do nó intermediário. Na figura 32a há um nó intermediário n entre os nós 1 e 2 e o parâmetro η é 0,5 pois a distância para os dois nós é igual. Na figura 32b, a posição do nó intermediário está mais próximo da origem pois o parâmetro η é menor do que 0,5. Por fim, na figura 32c

o parâmetro η é maior do que 0,5 sendo que o nó intermediário está mais próximo do nó destino, 2.

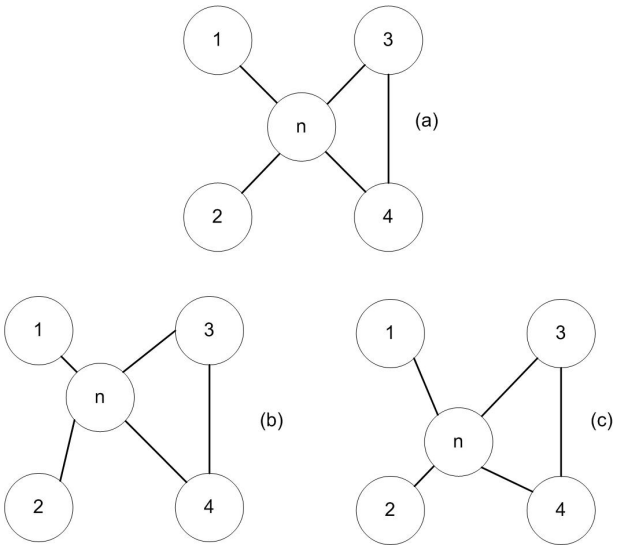


Figura 32. Exemplos de posicionamento para o nó intermediário baseados no parâmetro η: (a) η = 0,5; (b) η < 0,5; (c) η > 0,5

Como no ACO tradicional, o procedimento de seleção de nó para a transição e o cálculo de distâncias é repetido até que todos os nós sejam percorridos e um caminho seja definido. O peso deste caminho é dado pelo somatório das probabilidades de transição dos nós adjacentes do caminho. Quando todas as formigas percorreram o grafo, é considerado o final do ciclo.

Ao final de cada ciclo, é necessário recalcular os valores da matriz de feromônio, este processo é descrito na seqüência.

6.1.4 Escore da Árvore

Cada árvore produzida tem um valor de escore relativo a quão boa é sua solução. Este escore é baseado no somatório das probabilidades de transição escolhidas durante o caminho, como apresentado na equação 22:

$$S_{c(t)} = \sum_{i=0}^{n} \sum_{j=0}^{n} \begin{cases} P(i,j), se\ i,j \in c(t) \\ 0, caso\ contrário \end{cases} \tag{22}$$

Onde: $S_{c(t)}$ é o escore do caminho *c* percorrido no tempo *t*, *n* é o número de nós; *P(i,j)* é a probabilidade de transição entre os nós *i* e *j*; e *c(t)* é o caminho percorrido no tempo *t*.

O escore do melhor caminho obtido durante a execução é armazenado para utilização na atualização de feromônios, sendo conhecido como S_{best}.

Foi utilizado o somatório da função de probabilidade por dois motivos. O primeiro é devido ao fato do modelo utilizar apenas as distâncias evolutivas entre espécies o que não permitiria o uso de matrizes de substituição para avaliar a troca de bases entre as seqüências das espécies. O segundo motivo foi a suposição de que como as probabilidades são baseadas nas distâncias evolutivas entre as espécies, o somatório permitiria a minimização das distâncias no caminho obtido.

6.1.5 Atualização de feromônios

Ao contrário do modelo original de ACO para o TSP, foi definido que o incremento de feromônios seria realizado para todos os nós que pertencessem a pelo menos um caminho realizado pelas formigas no último ciclo. Este modelo foi utilizado para prevenir uma convergência muito rápida para um máximo local e não apresentou nenhuma ressalva. Na equação 23, é apresentada a equação para atualização de feromônios. Nesta equação, se considera tanto o incremento para os nós visitados no último ciclo quanto o decremento causado pela evaporação do feromônio das trilhas menos usadas. O decremento é um recurso utilizado para prevenir a convergência para máximos locais como definido no ACO original.

$$\tau(i, j) = \rho \cdot \tau(i, j) + (1 - \rho) \cdot \Delta\tau(i, j) \qquad (23)$$

Esta equação é similar à apresentada no ACO tradicional. Sendo ρ a taxa de evaporação do feromônio a cada ciclo, o que reduz a persistência do ambiente para as formigas. $\Delta\tau(i,j)$, é a taxa de incremento de feromônio. No modelo apresentado a taxa foi modificada, em relação ao ACO tradicional, para permitir um incremento proporcional entre os caminhos obtidos e o melhor caminho encontrado até o momento.

A taxa de incremento de feromônio é dada pela equação 24.

$$\Delta\tau(i,j) = \begin{cases} S_{c(t)} \cdot \left(S_{best}\right)^{-1} & \text{, se } (i,j) \in c(t) \\ 0 & \text{, caso contrário} \end{cases} \quad (24)$$

onde: c(t) é o caminho realizado por uma formiga no ciclo t, $S_{c(t)}$ é o escore do caminho c(t), e S_{best} é o melhor escore obtido até o momento.

Utilizando este procedimento para a atualização de feromônios, após um número de ciclos pré-definidos, é possível obter uma seqüência das espécies que formam o melhor caminho. Porém, é necessário reconstruir a árvore filogenética que tem a mesma estrutura seqüencial de espécies, mas com as ligações entre os ancestrais.

6.1.6 Reconstrução da árvore filogenética

A meta-heurística do ACO retorna uma seqüência das espécies que devem formar a melhor árvore. Porém é necessário descobrir de que forma estas se relacionam formando os ancestrais das espécies.

Para executar esta tarefa, é possível reutilizar os dados contidos na matriz de feromônios que conterá uma medida de proximidade entre as espécies. Isto é, espécies que são mais próximas entre si terão um valor maior na matriz de feromônios, enquanto que espécies mais distantes entre si terão valores menores. Desta forma, um algoritmo bastante simples foi implementado para encontrar o agrupamento que produzirá a árvore correta.

O algoritmo para reconstrução da árvore inicia procurando o par de espécies s1 e s2 contidos na seqüência do melhor caminho e que contenha o maior valor na matriz de feromônios. Este par é agrupado formando a espécie ancestral dessas duas. O procedimento se repete até que todas as espécies estejam agrupadas. A idéia principal de realizar esta operação é que espécies mais próximas no grafo serão mais visitadas pelas formigas, e desta forma o caminho entre essas espécies terá um maior nível de feromônio.

Detalhando melhor o algoritmo proposto, ele pode ser dividido nos passos apresentados na figura 33.

```
enquanto (todas as espécies não forem agrupadas)
      procura o par de espécies (i,j) onde o valor M(i,j) é máximo
      se (espécie i já foi agrupada) então
         substitui a espécie i pelo ancestral mais antigo dela
       fim se
      se (espécie j já foi agrupada) então
         substitui a espécie j pelo ancestral mais antigo dela
       fim se
       agrupa as duas espécies
       armazena que a nova espécie é a ancestral das duas
       recalcula as distâncias entre a nova espécie e as anteriores
       apaga o valor da matriz de feromônios para o par (i,j)
fim
```

Figura 33. Pseudocódigo detalhando o algoritmo de reconstrução de árvores

Com isto, foram apresentados os quatro passos básicos para desenvolver o modelo do problema de reconstrução de árvores filogenéticas utilizando a meta-heurística do ACO. Na seqüência, será apresentado o modelo para o problema do dobramento de proteínas.

6.2 Modelo de ACO aplicado ao problema de dobramento de proteínas

O modelo de ACO para o problema de dobramento de proteínas foi baseado no modelo 2D HP proposto por DILL (1985). Este modelo foi utilizado por permitir uma fácil correlação com o problema de otimização realizado através do ACO e por ser o mais simples modelo no caso hidrofóbico-polar.

6.2.1 Dados de entrada

Os dados de entrada neste modelo podem ser de duas formas distintas: no primeiro caso, os dados de entrada são uma seqüência de valores constituídos por um alfabeto de apenas duas letras {H,P}. Neste caso, cada aminoácido deve ser previamente definido como pertencente a um dos dois grupos principais: hidrofóbicos ou hidrofílicos (polar). O segundo tipo de dado de entrada é a seqüência de aminoácidos que compõem a estrutura primária da proteína, esta informação será convertida no alfabeto binário do primeiro caso utilizando para isto a classificação apresentada no anexo 1.

6.2.2 Modelo do problema

Para a movimentação das formigas utiliza-se a grade treliça quadrada bidimensional definida no modelo HP padrão. Pode-se considerar este como sendo o grafo padrão para a movimentação das formigas. Neste grafo, em cada movimento realizado por uma formiga é definida a posição do próximo aminoácido da seqüência de entrada.

Na meta-heurística original do ACO, são definidos pontos iniciais e finais sobre o grafo onde ocorre a movimentação das formigas. Porém, no caso do modelo HP só é possível definir o ponto inicial do dobramento da proteína sendo considerado o ponto final o local do posicionamento do último aminoácido, esta maneira de modelar o ACO foi proposta por SHMYGELSKA, HERNÁNDEZ e HOOS (2000).

Em uma grade quadrada, considerando-se um ponto qualquer, é possível se realizar apenas quatro movimentos diferentes, apresentados na figura 34: seguir adiante, virar à esquerda ou virar à direita e voltar atrás. Porém, o movimento de retorno deve ser considerado inválido por retornar a um nó já visitado.

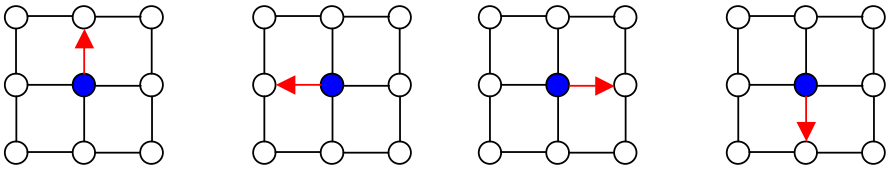


Figura 34. Movimentos possíveis de serem realizados em uma grade treliça

Como os movimentos não levam em consideração o posicionamento relativo à movimentação anterior da formiga, é necessário saber em qual direção está sendo realizado o movimento. Desta forma, é possível definir quatro direções diferentes: acima, abaixo, esquerda e direita.

A movimentação das formigas sobre a treliça é realizada através da função de transição (equação 25) que se baseia nas duas informações padrão do ACO: matriz de feromônios e informação local.

$$P_k(i,j) = \frac{[\tau(i,j)]^{\alpha}.[1/d(i,j)]^{-\beta}}{\sum_{u \in J_i^k} \{ [\tau(i,j)]^{\alpha}.[1/d(i,j)]^{-\beta} \}} \qquad (25)$$

Na equação 25 temos: $P_k(i,j)$ é a probabilidade da k-ésima formiga estando no nó *i* da grade treliça seguir para o nó *j*, sendo J_i^k o conjunto de nós conectados ao nó i que não foram visitados pela formiga; $d(i,j)$ é o número de ligações de aminoácidos hidrofóbicos não-locais, e $\tau(i,j)$ é a trilha de feromônios representada por uma matriz que define a atratividade de cada movimento para a formiga.

Caso a posição atual da formiga não permita nenhum movimento para a localização do próximo aminoácido, como apresentado na figura 35a, é desfeito um passo no caminho realizado e o movimento realizado é considerado inválido, como apresentado na figura 35b.

Figura 35. Exemplo de movimento não permitido, as bolinhas hachuradas representam aminoácidos, as bolinhas brancas são posições da grade: a) O aminoácido quadriculado é o último da seqüência que foi colocado, porém não existe nenhum movimento válido; b) O último movimento é desfeito e o caminho que resultou no movimento inválido é marcado como não permitido

A matriz de feromônios armazena a atratividade de cada movimento para as formigas. Desta forma, uma seqüência de movimentos que resulta em um dobramento com baixa energia livre será considerada mais atraente e terá maior chance de ocorrer novamente.

Pode-se deduzir também que a matriz de feromônios será uma matriz *i* x 3, onde *i* é o número de aminoácidos da seqüência de entrada e três são os movimentos possíveis de serem realizados que foram apresentados anteriormente.

6.2.3 Cálculo do Escore

Como mencionado na seção anterior, dobramentos com valores menores de energia livre devem ser preferidos em relação aos dobramentos que produziram valores maiores. Para que isto ocorra deve-se avaliar cada seqüência de movimentos que as formigas realizaram durante um ciclo.

No modelo 2D HP original, o escore é baseado nas ligações não locais dos aminoácidos com valores distintos para ligações H-H, H-P e P-P. Porém, no modelo aqui desenvolvido foi definido que apenas as ligações não locais entre aminoácidos hidrofóbicos (ligações H-H) seriam consideradas. Uma ligação não local entre aminoácidos ocorre quando dois aminoácidos estão em posições adjacentes da treliça e não são subseqüentes na seqüência.

Desta forma, para cada seqüência de movimentos obtida pelas formigas é necessário obter as posições cartesianas correspondentes. Para isto, é definido que o aminoácido inicial da seqüência terá sua posição na origem cartesiana (*x=0* e *y=0*) e, para cada movimento da seqüência levando em conta a direção atual, é posicionado o próximo aminoácido. Com estas informações, é construída uma lista contendo o tipo do aminoácido e sua posição nos eixos x e y, como apresentado na figura 36a, sendo que o dobramento correspondente pode ser visto na figura 36b.

H	0	0
P	1	0
P	1	1
H	0	1
P	-1	1
H	-1	0

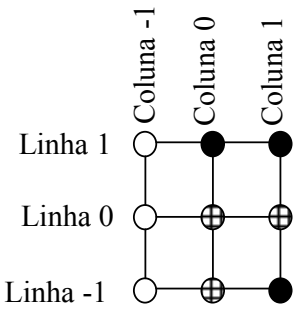


Figura 36. Exemplo de obtenção das posições, bolinha achuradas são aminoácidos hidrofóbicos, bolinhas pretas são aminoácidos polares: a) Lista com os tipos dos resíduos e suas posições cartesianas; b) Conformação da lista apresentada em (a)

Para a obtenção do escore, percorre-se a lista procurando o primeiro resíduo hidrofóbico (H). Encontrado este, procura-se nos resíduos subseqüentes da lista por outro hidrofóbico que o seja adjacente, isto é, que tenha alteração unitária em apenas uma das coordenadas. Para cada resíduo encontrado que seja adjacente soma-se -1 no escore do caminho. Se toda a lista foi percorrida em busca dos resíduos adjacentes repete-se o processo de busca do primeiro resíduo desconsiderando o que foi selecionado da última vez, este processo é detalhado no algoritmo apresentado na figura 37.

```
Escore = 0
x=0
Enquanto (x < número de resíduos)
        Se resíduo[x] é 'H' então
                Y = x+1
                Enquanto (y <número de aminoácidos)
                        Se (resíduo[y] é 'H' E  resíduo[y] adjacente resíduo[x]) então
                                Escore = Escore -1
                        Fim se
                Fim
         Fim se
         x = x + 1
fim
```

Figura 37. Pseudocódigo do cálculo de escore para o dobramento de proteínas

6.2.4 Atualização do Feromônio

Assim como no modelo apresentado para reconstrução de árvores filogenéticas, a atualização de feromônios é realizada para todas as seqüências de movimentos que foram produzidas. O objetivo aqui também foi diminuir a convergência rápida para um dobramento único.

A equação 26 apresenta a forma padrão de atualização de feromônios. Nesta fórmula, o par *ij* representa os movimentos possíveis para cada aminoácido da seqüência, ρ é a taxa de evaporação do feromônio que será responsável pela realimentação negativa do sistema, $\Delta\tau$ representa o incremento que será dado na matriz de feromônio para o movimento *j* quando da presença do *i*-ésimo aminoácido da seqüência.

$$\tau(i, j) = \rho \cdot \tau(i, j) + (1 - \rho) \cdot \Delta\tau(i, j) \tag{26}$$

Como no caso da reconstrução de árvores filogenéticas, o incremento a ser realizado para um dado dobramento é relativo ao escore do melhor dobramento realizado até o momento, como apresentado na equação 27. Este método de incremento da matriz de feromônio permite favorecer caminhos com escores mais altos que o atualmente encontrado e penalizar os caminhos que obtiveram valores menores.

$$\Delta\tau(i,j) = \begin{cases} S_{c(t)} \cdot (S_{best})^{-1} & \text{, se } (i,j) \in c(t) \\ 0 & \text{, caso contrário} \end{cases} \quad (27)$$

6.2.5 Formigas especiais

Para permitir uma busca mais eficaz do melhor dobramento e tornar o algoritmo mais robusto para obter bons resultados em problemas maiores, foi definida a criação de alguns tipos de recursos denominados de formigas especiais.

Estes recursos com funções incrementadas ou diferentes das formigas normais visam produzir melhores resultados através de um processamento maior das informações. No caso do ACO é proposta a criação de três tipos de formigas especiais, que serão descritos na seqüência.

6.2.5.1 Formiga Exploradora

Este recurso visa aumentar o espaço local que será analisado para realizar um movimento sobre a treliça. Este aumento se dá através da análise de múltiplos aminoácidos na seqüência para a realização de apenas um movimento. Isto é, se for definido que a formiga exploradora terá uma visão de três níveis, para realizar o próximo movimento sobre a treliça e definir a posição do próximo aminoácido, serão analisados todos os dobramentos possíveis para os três próximos aminoácidos da seqüência. Será definido como melhor movimento o que obtiver maior escore para a seqüência.

O número de combinações a serem analisadas será sempre o número de movimentos possíveis elevado ao número de níveis que a formiga terá de visão. No caso de uma formiga normal, onde se analisa apenas o próximo aminoácido, tem-se 3 combinações distintas de movimentos a serem analisados. Porém, no caso da formiga exploradora será necessário analisar 27 combinações diferentes. Disto pode-se intuir que apesar do operador produzir bons resultados, deve ser utilizado com parcimônia.

6.2.5.2 Formiga de dobramento em U

Um dobramento em “U” é realizado quando se tem uma seqüência com dois aminoácidos hidrofílicos entre dois aminoácidos hidrofóbicos (figura 38a, aminoácidos hidrofóbicos representados por bolinhas preenchidas com linhas na diagonal). Nesta situação particular os dois aminoácidos centrais normalmente se movimentarão para fora, aproximando os dois aminoácidos externos, como apresentado na figura 38b.

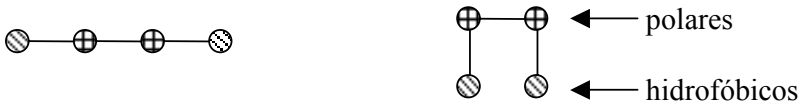


Figura 38. Exemplo de um dobramento em “U”: a) seqüência que gera um dobramento em “U”; b) dobramento em “U” realizado sobre a seqüência (a)

A formiga de dobramento em “U” busca sobre o melhor caminho a localização de uma seqüência que permita este dobramento e o realiza, fazendo um acréscimo na matriz de feromônios proporcional ao aumento do valor de energia livre.

No caso do operador percorrer toda a seqüência sem encontrar nenhum local para fazer o dobramento, este operador é desativado sendo novamente ativado quando da localização de um novo dobramento com escore maior do que o atual.

6.2.5.3 Formiga de dobramento em C

Um dobramento em “C” é a expansão do dobramento em “U” e tenta simular a criação de folhas-β. O primeiro passo para realizar um dobramento em “C” é procurar uma subseqüência dentro da seqüência original que se repita ao contrário a partir dos dois resíduos centrais. Isto significa que o tipo do primeiro resíduo da subseqüência deve ser igual ao tipo do último resíduo desta, e assim por diante, até os dois resíduos centrais que devem ser do mesmo tipo. Este caso é o ideal para o dobramento em “C”. Porém, em alguns casos a diferença de apenas 1 ou 2 resíduos na sub-seqüência não descaracteriza o dobramento em “C” que pode vir a ocorrer nestes casos. Entretanto, dobramento com alguns resíduos incorretos não foi implementado, pois é necessário um maior estudo das forças envolvidas no dobramento.

Na figura 39a é apresentada uma seqüência ideal para o dobramento em "C", pode-se ver que os tipos dos resíduos da primeira metade da seqüência são iguais aos tipos da segunda metade da seqüência vista do final para o início. Na figura 39b, mostra-se como ocorre o dobramento em "C" tornando as duas subseqüências paralelas.

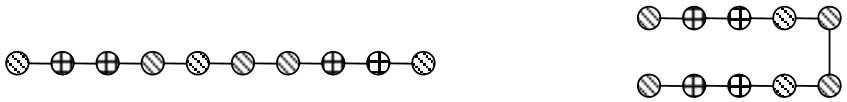


Figura 39. Exemplo de dobramento em "C": a) seqüência que gera um dobramento em "C"; b) dobramento em "C" realizado sobre a seqüência apresentada em (a)

Assim como no caso do dobramento em "U", este operador será realizado sobre o melhor caminho encontrado até o momento.

O operador de dobramento em "C" contém um índice interno que define o número de aminoácidos que devem ser analisados para que o dobramento ocorra. Este índice inicia-se com um valor fixo de dez, sendo que a cada vez que uma formiga de dobramento em "C" encontra um dobramento possível, este índice é incrementado em duas unidades. Quando a formiga não encontra um dobramento possível o índice é decrementado em duas unidades. Finalmente, quando o índice chega ao valor quatro, equivalente ao dobramento em "U", o operador é desativado sendo reativado quando da obtenção de um caminho melhor do que o atual.

CAPÍTULO 7

RESULTADOS

Na seqüência, serão apresentados os testes realizados e os resultados obtidos para os modelos de ACO aplicados à reconstrução de árvores filogenéticas e ao dobramento de proteínas.

Os resultados estão divididos da seguinte maneira: inicialmente, é apresentado o conjunto de dados utilizado para fazer a análise dos algoritmos implementados. Na seqüência, são apresentados os testes realizados para busca no conjunto de parâmetros que obtém o melhor resultado para o conjunto de dados. Em seguida, os valores de parâmetros tidos como ideais são utilizados para outros conjuntos de dados. Por último, são realizados testes para verificar o tempo de processamento para a obtenção dos resultados.

7.1 Reconstrução de árvores filogenéticas

Para validar o método desenvolvido para reconstrução de árvores filogenéticas foi utilizado um conjunto de dados com seqüenciais mitocondriais de vinte espécies de mamíferos, sendo os seguintes: rato (Rattus norvegicus), camundongo (Mus musculus), foca cinzenta (Halichoerus grypus), foca (Phoca vitulina), gato (Felis catus), rinoceronte branco (Ceratotherium simum), cavalo (Equus caballus), baleia da espécie finback (Balaenoptera physalus), baleia azul (Balaenoptera musculus), vaca (Bos taurus), gibão (Hylobates lar), gorila (Gorilla gorilla), homem (Homo sapiens), chimpanzé (Pan troglodytes), chimpanzé pigmeu (Pan paniscus), orangotango (Pongo pygmaeus), orangotango de sumatra (Pongo pygmaeus abelii), gambá (Didelphis virginiana), canguru (Macropus robustus) e ornitorrinco (Ornithorhynchus anatinus).

Foram utilizadas seqüências mitocondriais por estas serem herdadas apenas do organismo materno e, desta forma, acredita-se que estas seqüências mantenham-se mais integras e permitam uma melhor análise das características evolutivas.

Este conjunto de dados foi selecionado por haver um amplo estudo das ligações evolucionárias entre tais espécies, realizado por CAO *et al.* (1998), sendo a árvore apresentada no final o consenso atual da evolução destas espécies.

Além disso, diversos outros estudos (LI *et al.*, 2001; KOROTENSKY, GONNET, 2000) também utilizaram estes dados para avaliação.

As árvores obtidas com o método aqui proposto, assim como as apresentadas na literatura, não contêm distâncias evolutivas entre os ramos. Devido a este fator e para que fosse possível a comparação entre as árvores produzidas pelo método proposto e as apresentadas na literatura, foi utilizado um método de cálculo de distância topológica entre árvores. Este método foi desenvolvido por ROBINSON e FOULDS (1981) e leva o nome de seus autores, este método de cálculo de distância é apresentado no anexo 4.

7.1.1 Análise de parâmetros

Com a definição do conjunto de dados a ser utilizado inicialmente, e uma forma de verificar a qualidade das soluções obtida em relação a árvores filogenéticas reais, faz-se necessária a verificação das soluções obtidas com a alteração dos parâmetros de configuração do algoritmo. Esta seção visa verificar o comportamento do algoritmo em função da variação de seus parâmetros básicos.

Para isto, o ideal é testar todos os valores possíveis para todos os parâmetros. Porém, esta solução é inviável. Considerando os 6 parâmetros de entrada: α, β, ρ, d, k e c; e um intervalo de valores para cada um de 10 unidades, temos que o número de árvores a analisar seria da ordem de 10^6 ou 100.000 árvores produzidas. Se, além disso, for considerado um tempo médio de 30 segundos para reconstrução de uma árvore e cálculo de sua distância topológica, o tempo computacional para realização deste conjunto de teste seria de 30×10^6 segundos ou 35 dias.

Devido ao custo computacional, os parâmetros foram avaliados em grupos dependendo da correlação entre eles. Por exemplo, os parâmetros α e β são apenas relacionados com a probabilidade de uma formiga escolher um dado nó em detrimento dos demais. Desta forma, é possível supor que a alteração em um destes dois parâmetros terá um resultado independente aos outros parâmetros. Com isto, apenas as combinações destes dois parâmetros são avaliadas, enquanto os demais parâmetros são mantidos constantes.

Na seqüência, os parâmetros de taxa de evaporação (ρ) e número de formigas (k) são avaliados agrupados da mesma forma anterior. Pois, segundo a literatura, estes dois fatores contribuem para o processo de estigmergia e obtenção da solução. Isto é, se tiver um número muito baixo de formigas ou uma taxa de evaporação muito alta o sistema

não convergirá para um padrão. Da mesma forma, se o número de formigas for muito alto ou houver uma taxa de evaporação baixa, o sistema tenderá a um máximo local rapidamente.

Por último, os parâmetros de distância entre espécies e ciclos foram avaliados separadamente dos outros parâmetros por não terem correlação com os demais parâmetros.

No caso da avaliação dos parâmetros α e β, o primeiro parâmetro controla o nível de exploração no espaço de busca, pois através deste é possível definir a importância dos resultados anteriores no movimento das formigas. O parâmetro β define a importância da distância evolutiva para o movimento da formiga.

Durante os testes, verificou-se que valores altos do primeiro parâmetro faziam o algoritmo convergir rapidamente para um máximo local, sendo que a árvore produzida tinha uma distância topológica maior do que a da melhor árvore obtida com o ACO. Foi verificado que o parâmetro β deveria ter um valor um pouco superior ao parâmetro α, sendo que valores menores produzem árvores sub-ótimas e valores maiores produzem árvores com as espécies agrupadas seqüencialmente, como apresentado na figura 40.

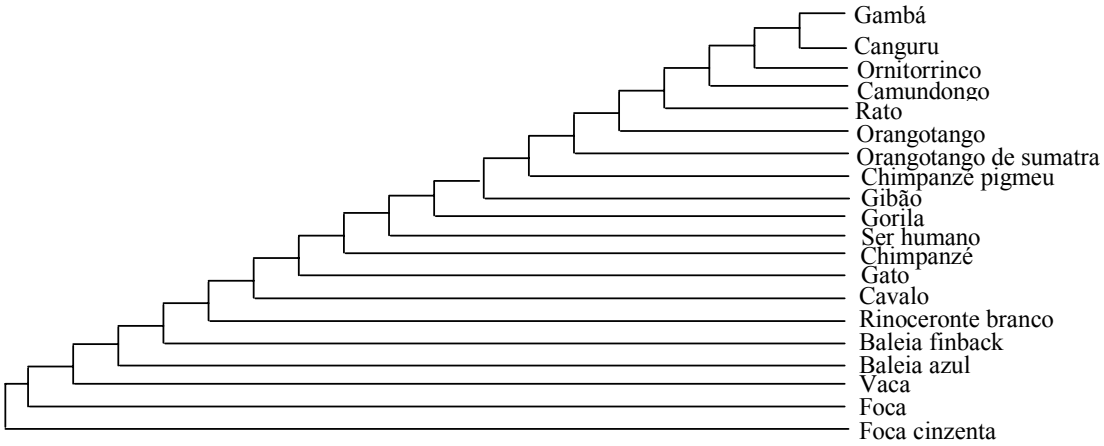


Figura 40. Árvore reconstruída com valores muito altos de β

Na figura 41a é apresentada a melhor árvore produzida com o método ACO. Na figura 41b é reproduzida a árvore apresentada por Cao como sendo a árvore que melhor expressa a relação evolutiva das espécies.

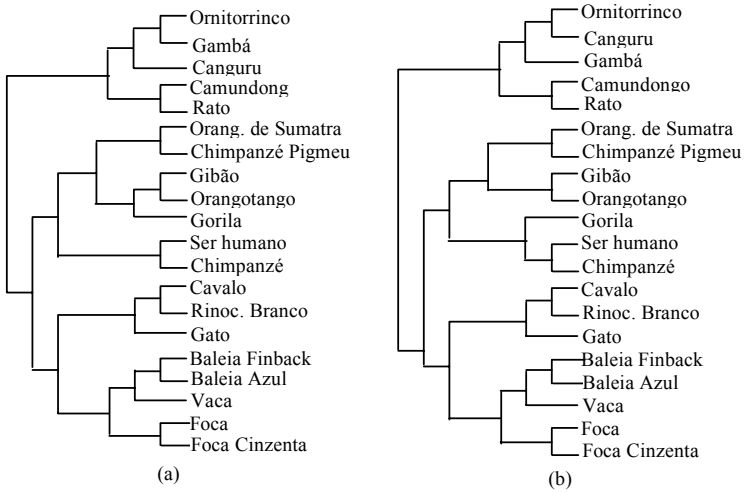


Figura 41. Duas árvores distintas: a) através do método ACO; b) árvore consenso.

Nas árvores apresentadas na figura 41, existem apenas duas diferenças topológicas mínimas, que são: a ordem de ancestrais entre o ornitorrinco, o canguru e o gambá. Na árvore (a) o gambá e o ornitorrinco descendem do mesmo ancestral que, por sua vez descende do mesmo ancestral do canguru. Para a árvore (b), o gambá e o canguru trocam de ancestrais entre si. A segunda alteração diz respeito à localização do gorila junto aos outros primatas. Na árvore obtida pelo método desenvolvido, o gorila aproxima-se mais do gibão e do orangotango, enquanto na árvore (b) o gorila está mais próximo evolutivamente dos seres humano e do chimpanzé.

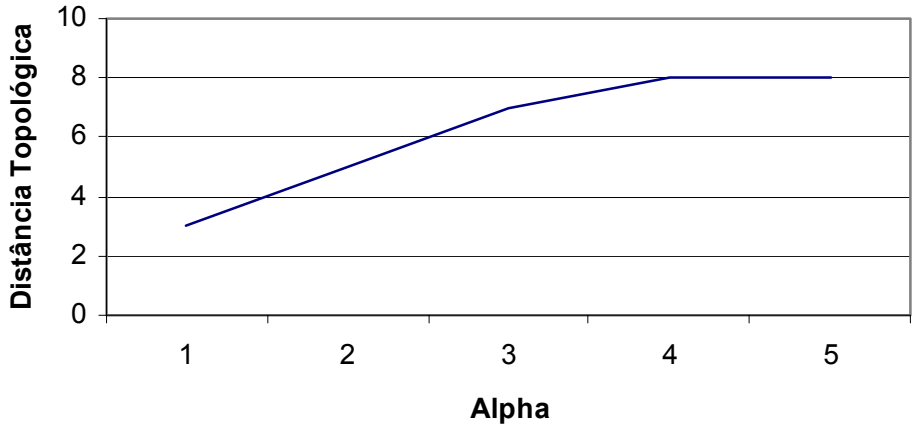
Na tabela 3, são apresentadas as distâncias topológicas de acordo com a variação dos cinco parâmetros. A primeira linha contém o conjunto de parâmetros que obteve o melhor resultado e na seqüência são apresentados os outros casos testados. Nesta tabela, foram agrupados os parâmetros conforme as características das árvores apresentadas. Entretanto, os parâmetros foram avaliados nos seguintes intervalos: α e β com valores inteiros entre 1 e 5; ρ entre 0,1 e 0,9, variando-se em passos de 0,1; k entre 50 e 500, em passos de 50; e, por último, c entre 5 e 50, com variação de 5.

Tabela 3. Distâncias topológicas obtidas com a variação dos parâmetros de entrada

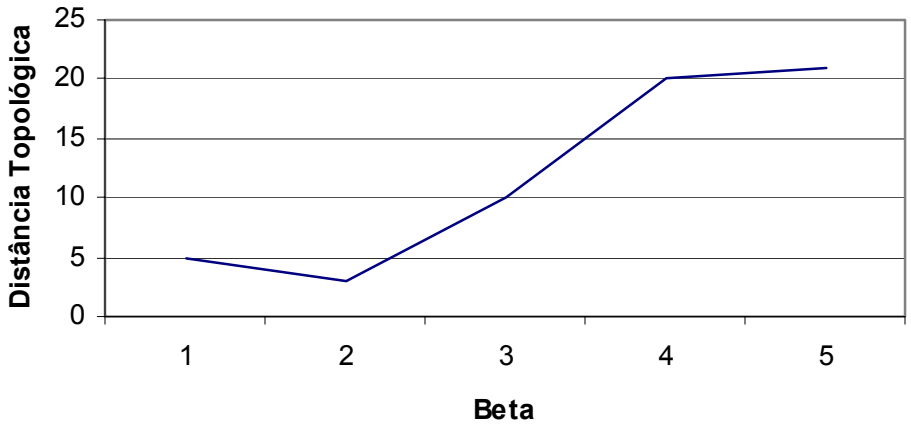
Parâmetros de Entrada						Distância RF
α	**β**	**ρ**	**d**	**k**	**c**	
1	2	0,9	0,5	500	50	3
3	2	0,9	0,5	500	50	7-8
1	>3	0,9	0,5	500	50	20-22
1	2	<0,5	0,5	500	50	6-8
1	2	0,9	0,5	<150	50	6-15 (aleatório)
1	2	0,9	0,5	500	<40	5-15
1	2	0,9	<0,5	500	50	5
1	2	0,9	>0,5	500	50	8

Pode-se verificar que conforme os valores inseridos nos parâmetros de entrada é possível obter um grande número de árvores candidatas à árvore filogenética para as espécies de entrada. Também é possível verificar que há uma boa variabilidade da distância topológica obtida em rodadas diferentes com os mesmos parâmetros de entrada.

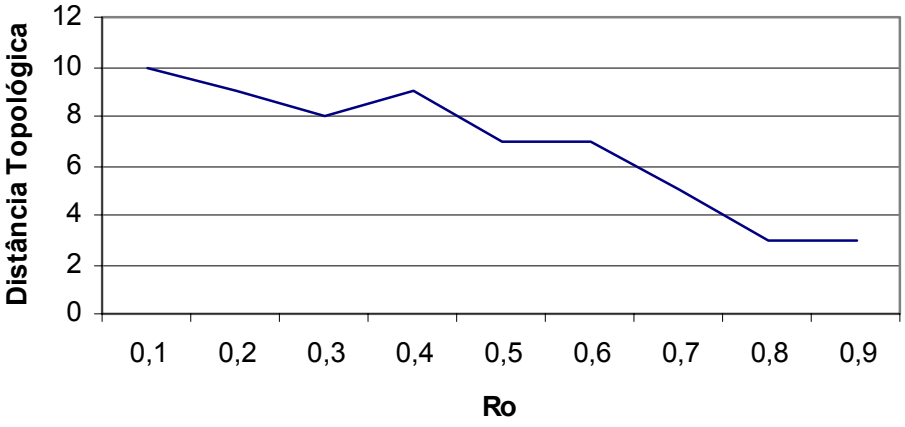
Nos gráficos apresentados na figura 42, pode ser visto a variação da distância topológica quando da alteração de cada um dos parâmetros separadamente. Estes testes foram realizados fixando os valores dos parâmetros no conjunto de melhor resposta e variando apenas um parâmetro por vez.



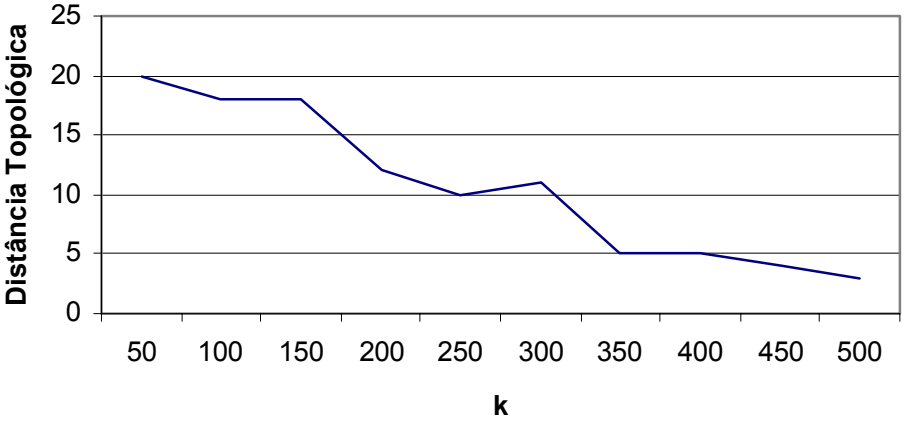
(a)



(b)



(c)



(d)

Figura 42. Distâncias topológicas obtidas em relação à árvore consenso, com a variação dos parâmetros do ACO: a) variando-se α; b) em relação à variação de β; c) conforme o acréscimo da taxa de evaporação; d) em relação ao aumento do número de formigas

7.1.2 Tempo de processamento

Como mencionado na revisão de literatura, o maior problema da reconstrução de árvores filogenéticas é o crescimento exponencial do espaço de busca em relação ao número de espécies a serem analisadas. Este é o principal motivo pelo qual torna-se interessante a utilização de algoritmos heurísticos.

Desta forma, qualquer nova metodologia proposta deve conter uma análise do tempo de processamento. Para efetuar esta avaliação, foi utilizado um segundo conjunto de dados, composto de 470 seqüências completas de DNA mitocondrial de diversas espécies.

Este conjunto de dados foi utilizado apenas com o intuito de verificar o tempo de processamento, não sendo verificadas as árvores produzidas. Sendo que também não se procurou avaliar o melhor conjunto de parâmetros para este conjunto de dados.

No teste de desempenho foi utilizado um computador PC Celeron 1.1GHz com 512MB de RAM rodando Windows 2000 Professional.

O teste foi realizado da seguinte forma: foi construída a matriz quadrada com as distâncias evolutivas entre as 470 espécies com o método da similaridade (CAMPBELL, MRÁZEK, KARLIN; 1999), apresentado na seção 3.1.1. O algoritmo iniciava com um valor inicial de cinco espécies para serem analisadas, ao final da construção da árvore incluía-se uma nova espécie e o processo era reiniciado.

O ACO rodava com os valores ideais para os parâmetros de entrada. Foram considerados parâmetros ideais os que obtiveram a menor distância topológica no teste de análise dos parâmetros. O critério de parada selecionado foi o número de ciclos.

Na figura 43, é apresentado o gráfico da evolução do tempo de processamento em relação ao número de espécies de entrada.

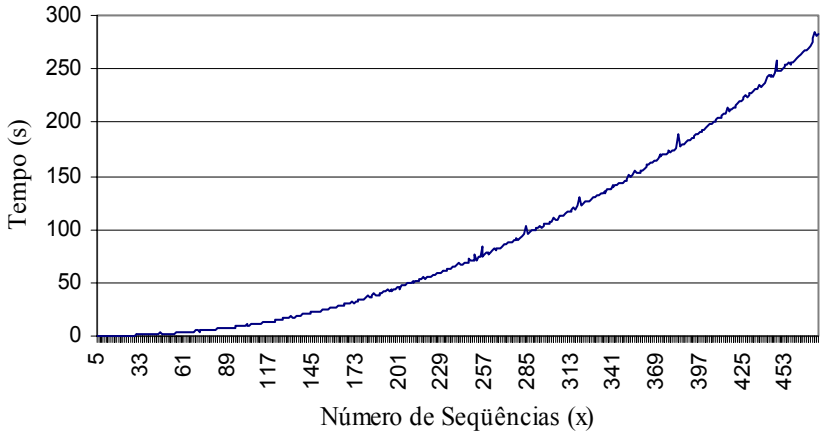


Figura 43. Tempo de processamento em segundos com relação ao número de espécies a serem analisadas

Para avaliar o crescimento do tempo de processamento em relação ao número de espécies, foi inserida uma linha de tendência polinomial em relação à curva obtida na figura 43. A equação polinomial obtida foi $y = 0{,}0013x^2 - 0{,}0359x + 1{,}0019$, onde y é o tempo de processamento esperado em segundos com x espécies de entrada. A curva apresentada tem um $R^2 = 0{,}9997$, o que demonstra que a curva ajustada é praticamente igual à curva do tempo de processamento, dentro da faixa de valores observados.

7.1.3 Comparação com outros métodos

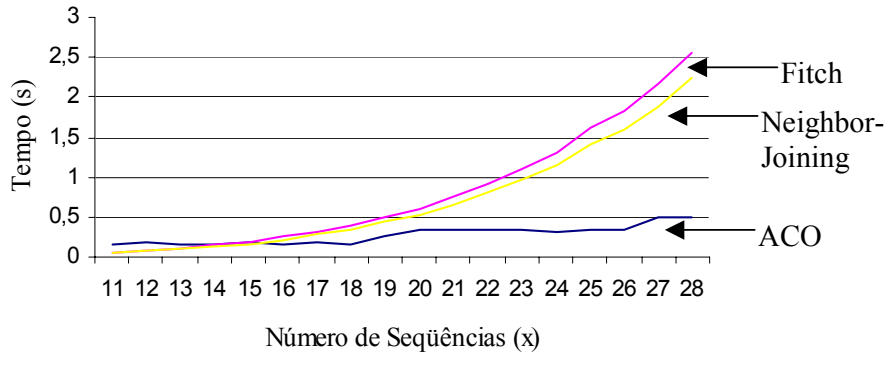
Para avaliar o desempenho do método proposto através de ACO foram realizados testes utilizando os mesmos dados de entrada com os métodos Fitch e Neighbor-Joining. Para isso foi utilizado o pacote de programas PHYLIP, que contém implementações destes métodos. Também foi utilizada na comparação uma quarta árvore apresentada por LI *et al.* (2001) onde é utilizado o método Hypercleaning sobre diversas árvores obtidas pelo método Neighbor-Joining.

Da mesma forma que na avaliação do ACO, foi calculada a distância Robinson-Foulds das árvores geradas pelos dois métodos com a árvore considerada como consenso atual. Na tabela 4 são apresentadas as distâncias topológicas obtidas pelos quatro métodos em relação a árvore consenso.

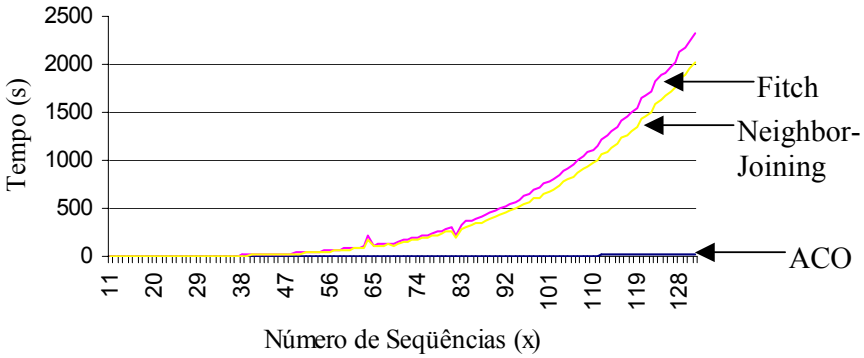
Tabela 4. Distâncias RF obtidas pelos 4 métodos em relação a árvore consenso

Método	**Distância RF**
ACO	5
Fitch	15
Neighbor-Joining	17
Hypercleaning	4

No teste de tempo de processamento, foi comparado o ACO com os algoritmo Fitch e Neighbor-Joining. Isto se deve ao fato de o Hypercleaning não ter uma implementação fácil, sendo comparada apenas a árvore obtida na literatura. Nas figuras 44a e 44b são apresentados o tempo de processamento do ACO, do Fitch e do Neighbor-Joining para instâncias com um número de até 27 espécies e para um segundo conjunto de seqüências, com 131 seqüências, respectivamente. Não foram realizados testes acima desta quantidade de espécie.



(a)



(b)

Figura 44. Comparação entre o método desenvolvido e o programa Fitch: a) para pequenas instâncias; b) para grandes instâncias

7.2 Dobramento de proteínas

Assim como no caso do modelo anterior, foi desenvolvida uma rotina para validação do modelo desenvolvido para o dobramento de proteínas. A rotina de teste consiste na seleção de uma proteína com 85 resíduos, e que seja conhecido o valor ótimo para o nível de energia livre. Para esta proteína, este valor é -51. A proteína escolhida foi selecionada apenas por conter um número de aminoácidos considerado razoável. Não foi considerada a estrutura nativa da proteína para esta seleção.

Com esta seqüência definida, são realizados testes variando-se os parâmetros de entrada para verificar o conjunto de valores que apresentam a melhor resposta. Na seqüência, os testes foram realizados utilizando outras seqüências que também tenham seus valores ótimos de energia conhecidos. Por último, são apresentados os resultados dos testes de tempo de processamento do algoritmo.

7.2.1 Análise de parâmetros

Assim como no caso anterior, à análise de todas as combinações possíveis dos parâmetros de entrada torna-se inviável devido ao tempo de processamento necessário para tal análise. Da mesma forma que antes, os parâmetros foram agrupados conforme sua funcionalidade.

Para cada combinação de parâmetros, foram executadas 100 rodadas de teste com a seqüência definida.

Primeiro, foi analisada a variação dos parâmetros α e β, entre 1 e 5, cada um, sendo os demais parâmetros fixos em um valor escolhido. Na tabela 5, são apresentados os valores de melhor com o número de vezes que este valor foi obtido entre parênteses, a média mais ou menos o desvio-padrão.

Tabela 5. Resultados obtidos quando varia-se os parâmetros α e β no intervalo entre 1 e 5

Parâmetros de Entrada					Média	Melhor
α	**β**	**ρ**	**k**	**c**		**Valor**
1	1	0,9	500	50	-47,63 ± 2,06	-51(5)
2	1	0,9	500	50	-48,52 ± 1,25	-51(9)
3	1	0,9	500	50	-50,2 ± 0,91	-51(33)
4	1	0,9	500	50	-49,96 ± 1,45	-51(16)
5	1	0,9	500	50	-48,12 ± 2,06	-51(14)
1	2	0,9	500	50	-47,82 ± 2,06	-51(8)
2	2	0,9	500	50	-48,94 ± 1,25	-51(18)
3	2	0,9	500	50	-49,83 ± 0,91	-51(28)
4	2	0,9	500	50	-49,05 ± 1,45	-51(25)
5	2	0,9	500	50	-49,52 ± 1,75	-51(19)
1	3	0,9	500	50	-48,43 ± 2,06	-51(30)
2	3	0,9	500	50	-49,93 ± 1,25	-51(26)
3	3	0,9	500	50	-50,3 ± 0,91	-51(33)
4	3	0,9	500	50	-49,43 ± 1,45	-51(30)
5	3	0,9	500	50	-49,48 ± 1,75	-51(29)
1	4	0,9	500	50	-49,08 ± 2,06	-51(22)
2	4	0,9	500	50	-49,67 ± 1,25	-51(24)
3	4	0,9	500	50	-49,89 ± 0,91	-51(25)
4	4	0,9	500	50	-49,52 ± 1,45	-51(25)
5	4	0,9	500	50	-49,43 ± 1,75	-51(22)
1	5	0,9	500	50	-49,41 ± 2,06	-51(21)
2	5	0,9	500	50	-49,76 ± 1,25	-51(25)
3	5	0,9	500	50	-49,91 ± 0,91	-51(28)
4	5	0,9	500	50	-49,72 ± 1,45	-51(26)
5	5	0,9	500	50	-49,61 ± 1,75	-51(23)

Na seqüência, foram variados os parâmetros ρ, entre 0,1 e 0,9, e k, entre 50 e 500. Para este teste, foram escolhidos os dois valores de α e β que apresentaram a maior média e o menor desvio-padrão. Na tabela 6, são apresentados os resultados para α=3 e β=1 e, na tabela 7, os resultados para α=3 e β=3.

Tabela 6. Resultados obtidos variando-se ρ entre 0,1 e 0,9 e k entre 50 e 500 sendo que α=3 e β=1

Parâmetros de Entrada					Média	Melhor
α	**β**	**ρ**	**k**	**c**		**Valor**
3	1	0,1	50	50	-23,76 ± 4,37	-28(3)
3	1	0,2	50	50	-30,25 ± 3,22	-32(2)
3	1	0,3	50	50	-35,37 ± 4,13	-37(3)
3	1	0,4	50	50	-31,29 ± 5,86	-35(1)
3	1	0,5	50	50	-27,25 ± 3,39	-29(2)
3	1	0,6	50	50	-31,05 ± 8,86	-43(3)
3	1	0,7	50	50	-30,18 ± 5,43	-36(4)
3	1	0,8	50	50	-32,83 ± 7,28	-41(2)
3	1	0,9	50	50	-33,79 ± 8,59	-46(5)
3	1	0,1	250	50	-41,23 ± 1,25	-42(2)
3	1	0,2	250	50	-41,26 ± 1,21	-42(1)
3	1	0,3	250	50	-42,85 ± 1,37	-44(3)
3	1	0,4	250	50	-43,67 ± 1,21	-45(3)
3	1	0,5	250	50	-43,73 ± 1,07	-45(5)
3	1	0,6	250	50	-44,89 ± 1,08	-46(7)
3	1	0,7	250	50	-46,01 ± 1,05	-47(7)
3	1	0,8	250	50	-46,26 ± 1,26	-48(5)
3	1	0,9	250	50	-46,29 ± 1,2	-48(6)
3	1	0,1	500	50	-41,46 ± 1,08	-43(1)
3	1	0,2	500	50	-42,38 ± 0,98	-43(3)
3	1	0,3	500	50	-43,05 ± 0,9	-44(4)
3	1	0,4	500	50	-43,36 ± 0,8	-44(6)
3	1	0,5	500	50	-46,09 ± 0,96	-47(5)
3	1	0,6	500	50	-47,99 ± 0,98	-49(10)
3	1	0,7	500	50	-50,03 ± 0,95	-51(5)
3	1	0,8	500	50	-50,08 ± 0,92	-51(20)
3	1	0,9	500	50	-50,2 ± 0,91	-51(33)

Tabela 7. Resultados obtidos variando-se ρ entre 0,1 e 0,9 e k entre 50 e 500 sendo que α=3 e β=3

Parâmetros de Entrada					Média	Melhor Valor
α	β	ρ	k	c		
3	3	0,1	50	50	-24,01 ± 4,81	-29(1)
3	3	0,2	50	50	-31,02 ± 3,89	-35(3)
3	3	0,3	50	50	-34,87 ± 3,26	-37(2)
3	3	0,4	50	50	-35,74 ± 4,21	-38(3)
3	3	0,5	50	50	-32,52 ± 5,82	-39(5)
3	3	0,6	50	50	-36,05 ± 3,69	-40(3)
3	3	0,7	50	50	-28,08 ± 5,29	-37(4)
3	3	0,8	50	50	-35,46 ± 6,19	-42(5)
3	3	0,9	50	50	-31,91 ± 7,26	-40(4)
3	3	0,1	250	50	-41,25 ± 1,07	-42(6)
3	3	0,2	250	50	-41,31 ± 0,91	-42(7)
3	3	0,3	250	50	-42,99 ± 1,31	-45(5)
3	3	0,4	250	50	-43,84 ± 1,08	-45(8)
3	3	0,5	250	50	-43,92 ± 1,05	-45(5)
3	3	0,6	250	50	-44,95 ± 1,05	-46(7)
3	3	0,7	250	50	-46,09 ± 1,02	-47(9)
3	3	0,8	250	50	-46,45 ± 1,46	-48(8)
3	3	0,9	250	50	-46,51 ± 1,48	-48(12)
3	3	0,1	500	50	-41,74 ± 1,42	-43(7)
3	3	0,2	500	50	-42,78 ± 0,95	-43(8)
3	3	0,3	500	50	-43,25 ± 0,86	-44(8)
3	3	0,4	500	50	-44,16 ± 0,85	-45(9)
3	3	0,5	500	50	-46,79 ± 1,03	-49(10)
3	3	0,6	500	50	-49,34 ± 1,19	-51(8)
3	3	0,7	500	50	-50,08 ± 1,14	-51(14)
3	3	0,8	500	50	-50,16 ± 1,02	-51(25)
3	3	0,9	500	50	-50,3 ± 0,91	-51(33)

A partir dos resultados apresentados nas tabelas anteriores, optou-se pela realização dos demais testes com os seguintes parâmetros: $\alpha=\beta=3$, $\rho=0,9$, $k=500$, $c=50$. Pois este conjunto apresentou ao mesmo tempo os maiores valores de mínimo de energia e o menor valor de desvio-padrão o que indica que o algoritmo está convergindo em quase todas as rodadas para o resultado ideal.

7.2.2 Energia livre

Para avaliar melhor a metodologia, foram realizados testes utilizando outras 15 seqüências diversas encontradas em outros artigos e na Internet. As seqüências no formato H-P são apresentadas no anexo 2. Na tabela 8, são apresentados os resultados obtidos para cada uma das seqüências junto ao máximo encontrado. Nestas seqüências, os testes foram realizados de duas formas: a primeira, com as formigas especiais desativadas e, no segundo caso com elas ativas. Para cada uma das seqüências, foram executadas 10 rodadas de teste sendo definidos os parâmetros básicos do ACO definidos com os melhores valores obtidos na análise de parâmetros.

Foi verificado que não era necessário um grande número de formigas especiais para que o algoritmo começasse o processo de realimentação positiva nos melhores caminhos. Por este motivo foi definido que as formigas exploradoras seriam 5% da colônia total e as formigas especiais para dobramento seriam executadas uma vez a cada final de ciclo.

Tabela 8. Resultados obtidos para as seqüências avaliadas com os parâmetros definidos

Seq.	Tamanho	Máximo conhecido	Sem formigas especiais		Com formigas especiais	
			Máximo obtido	Média obtida	Máximo obtido	Média obtida
1	20	-9	-9(20)	-9 ± 0	-9(20)	-9 ± 0
2	24	-9	-9(20)	-9 ± 0	-9(20)	-9 ± 0
3	25	-8	-8(20)	-8 ± 0	-8(20)	-8 ± 0
4	36	-14	-14(20)	-14 ± 0	-14(20)	-14 ± 0
5	48	-23	-23(20)	-23 ± 0	-23(20)	-23 ± 0
6	50	-21	-21(20)	-21 ± 0	-21(20)	-21 ± 0
7	60	-36	-36(18)	-35,8 ± 0,5	-36(20)	-36 ± 0
8	64	-42	-42(20)	-42 ± 0	-42(20)	-42 ± 0
9	85	-53	-51(18)	-50,7 ± 0,5	-53(20)	-53 ± 0
10	100	-50	-49(9)	-47,6 ± 0,7	-50(20)	-50 ± 0
11	100	-48	-43(17)	-42,8 ±0,6	-48(18)	-47,6 ± 0,8
12	106	---	-45(14)	-44,5 ± 1,1	-49(15)	-47,9 ± 1,5
13	113	---	-53(16)	-51,8 ± 2,5	-60(18)	-59,8 ± 0,6
14	128	---	-70(10)	-67,3 ± 3,01	-73(20)	-73 ± 0
15	330	---	-146(4)	-143,7 ± 2,7	-160(20)	-160 ± 0

7.2.3 Tempo de processamento

A utilização de algoritmos de busca ou otimização, como o ACO, só é interessante a problemas que tem um espaço de busca muito amplo e que não permite a um método de busca normal percorrê-lo em um tempo razoável. Da mesma forma é necessário avaliar se o tempo de processamento necessário para que o modelo desenvolvido retorne um resultado satisfatório seja compatível com a complexidade do problema e não se torne rapidamente improdutivo.

Nos testes que foram realizados anteriormente, para verificar o nível de energia do dobramento foram armazenados os tempos de processamento, que são apresentados na tabela 9.

Tabela 9. Tempo de processamento para as seqüências avaliadas

		Sem formigas especiais		**Com formigas especiais**	
Seqüência	**Tamanho**	**Tempo Máximo (s)**	**Média de tempo (s)**	**Tempo Máximo (s)**	**Média de tempo (s)**
1	20	5,05	4,87	15,15	14,82
2	24	6,05	5,76	17,96	16,82
3	25	6,86	6,39	20,82	17,96
4	36	10,72	10,01	31,26	29,86
5	48	15,31	13,86	46,37	43,82
6	50	22,39	21,46	68,26	65,68
7	60	40,07	39,71	116,26	115,82
8	64	46,17	45,58	131,29	130,95
9	85	50,33	49,12	143,89	143,85
10	100	50,33	49,42	143,96	141,62
11	100	54,37	54,04	151,97	149,84
12	106	56,81	56,21	164,23	160,25
13	113	60,87	59,71	171,94	168,96
14	128	65,59	64,96	178,87	175,62
15	330	288,00	285,43	843,16	839,06

Para melhor visualização do tempo de processamento em relação ao número de resíduos da seqüência de entrada, na figura 45 são apresentados dois gráficos com os tempos de processamento em barras. Foi dividido em dois gráficos para facilitar a visualização, utilizando escalas diferentes.

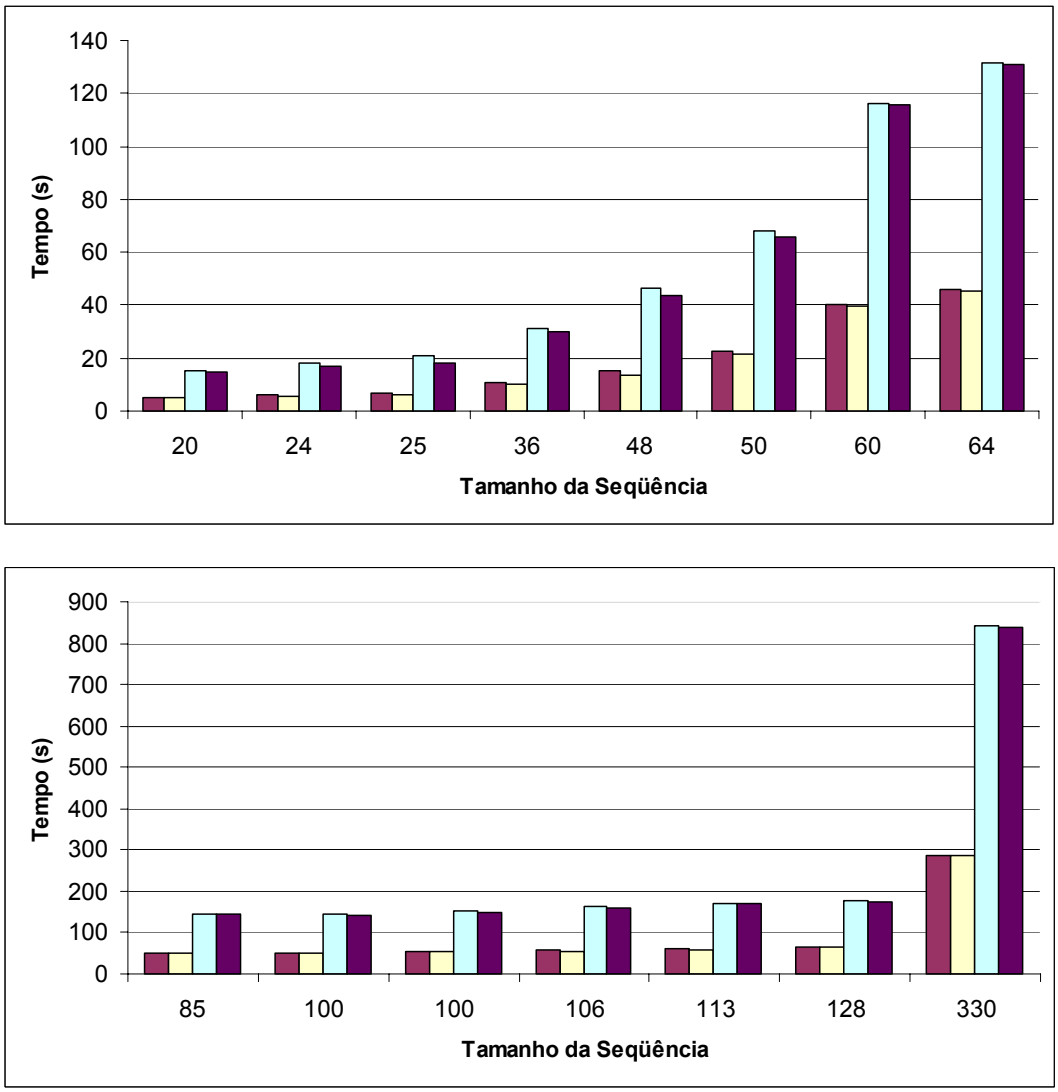


Figura 45. Gráficos com os tempos de resposta na seguinte ordem: 1ª barra – tempo máximo sem formigas especiais; 2ª barra – tempo médio sem formigas especiais; 3ª barra – tempo máximo com formigas especiais; 4ª barra – tempo médio com formigas especiais.

Assim como no caso dos tempos de processamento para a reconstrução de árvores, foi adicionada uma linha de tendência ao pior caso, tempo máximo com formigas especiais, com o intuito de obter uma equação que expressasse o tempo de processamento necessário conforme o número de resíduos na seqüência. A equação obtida foi $y = 3{,}9338x^2 - 16{,}132x + 31{,}006$ sendo que esta tem o valor de $R^2 = 0{,}9839$, apresentando uma grande similaridade com a curva obtida.

7.2.4 Comparação com outros métodos

Os resultados obtidos neste trabalho foram comparados com os resultados obtidos por dois outros modelos desenvolvidos com a meta-heurística ACO apresentados por SHMYGELSKA, HERNÁNDEZ e HOOS (2002) e SHMYGELSKA e HOOS (2003). Foram selecionados estes trabalhos por utilizarem as mesmas seqüências de entrada, o que permite uma avaliação apenas dos métodos entre si.

Na tabela 10 são apresentados os valores de energia e o tempo de processamento para a obtenção dos valores. Os valores apresentados na literatura são tempos médios em relação a 200 a 700 rodadas do primeiro método e 300 a 500 rodadas do segundo método sendo utilizado um computador PC Intel Pentium III de 1 GHz com 512 MB de memória RAM. Por outro lado, os resultados apresentados neste trabalho são relativos ao tempo médio e à média de energia livre obtidos em 20 rodadas com a metodologia proposta e foram obtidos em um computador PC Intel Pentium III de 600 MHz com 256 MB de memória RAM.

Para este teste foi utilizada a configuração de parâmetros ideal verificada na análise de parâmetros e foram ativadas as formigas especiais da mesma forma que no teste de energia livre.

Tabela 10. Valores de energia máxima e tempo de processamento para os dois métodos de dobramento com ACO da literatura e com o novo método proposto

	Seqüências		**ACO Original SHMYGELSKA *et al.* (2002)**		**ACO Aprimorado SHMYGELSKA *et al.* (2003)**		**Este trabalho**	
Nº	**Tam**	**Máximo Energia**	**Energia Obtida**	**Tempo (s)**	**Energia Obtida**	**Tempo (s)**	**Energia Obtida**	**Tempo (s)**
1	20	-9	-9	23,9	-9	3,3	-9	14,82
2	20	-9	-9	26,4	-9	2,5	-9	16,82
3	25	-8	-8	35,3	-8	10,6	-8	17,96
4	36	-14	-14	4746,1	-14	11,8	-14	29,86
5	48	-23	-23	1920,9	-23	405,7	-23	43,82
6	50	-21	-21	3000,2	-21	4952,9	-21	65,68
7	60	-36	-34	4898,7	-36	6247,1	-36	115,82
8	64	-42	-32	4736,9	-42	5844,9	-42	130,95

Como pode ser visto na tabela, o modelo desenvolvido apresentou resultados equivalentes ao segundo modelo apresentado na literatura. Porém o tempo de processamento necessário para o nosso modelo é quase 45 vezes menor que o modelo apresentado na literatura. Cogita-se que esta diferença no tempo de processamento ocorreu pelo método de retorno de caminhos inválidos desenvolvido neste modelo em relação ao método apresentado na literatura.

CAPÍTULO 8

DISCUSSÃO E CONCLUSÃO

8.1 Análise dos Resultados

Os testes realizados utilizaram conjuntos de dados na avaliação da reconstrução de árvores e um conjunto de seqüências para o dobramento de proteínas.

Quanto ao problema da reconstrução de árvores, pode-se verificar que o modelo proposto apresentou uma estrutura de árvore mais próxima do consenso atual quando comparado a outros métodos atuais que se baseiam em matrizes de distância como dados de entrada.

Apesar disto, é possível verificar uma grande suscetibilidade do algoritmo aos seus parâmetros de entrada, observável nos gráficos apresentados na figura 37, visto que apenas um conjunto de parâmetros obteve o melhor valor, observa-se que a alteração de quaisquer dos parâmetros de entrada em sua unidade mínima já altera o resultado.

Porém, mesmo com esta grande suscetibilidade o algoritmo apresentou bons resultados para a grande maioria dos conjuntos de parâmetros testados. Isto se torna claro quando comparadas às distâncias de Robinson-Foulds apresentadas na tabela 3 (que foram obtidas com a variação dos parâmetros) com aquelas apresentadas na tabela 4 (que contém as distâncias obtidas com diversos métodos).

O método desenvolvido também apresentou uma grande vantagem quando comparado o tempo de processamento computacional para instâncias com um grande número de espécies. Apesar do crescimento exponencial do tempo de processamento, apresentado no gráfico da figura 43, esta curva apresentou um crescimento pequeno quando comparado com um método bastante robusto como algoritmo de Fitch. Isto se torna evidente nos gráficos apresentados na figura 44, onde se pode verificar que o algoritmo desenvolvido se torna mais rápido a partir de instâncias com 16 espécies ou mais.

O único método que superou o modelo proposto quanto à topologia obtida foi o Hypercleaning apresentado em LI *et al.* (2001). Na verdade o algoritmo de Hypercleaning não é realmente um método para reconstrução de árvores filogenéticas, sendo um método iterativo que executa um algoritmo de reconstrução, por exemplo: o Neighbor-Joining, com apenas uma parcela dos dados de entrada. O procedimento é

repetido e são analisados quais ramos são mais produzidos e, assim sendo, têm maior probabilidade de serem corretos. Por ser baseado em um método iterativo sobre os algoritmos de reconstrução de árvores, o método de Hypercleaning torna-se inviável computacionalmente quando o número de espécies de entrada aumenta muito.

Para o problema do dobramento de proteínas, foram escolhidas 15 seqüências de proteínas, apresentadas no anexo 2. Estas proteínas foram selecionadas por permitirem comparação com trabalhos anteriores e pelas suas características.

O modelo proposto para o dobramento de proteínas parece ser mais robusto que o de reconstrução de árvores filogenéticas. Esta afirmação se baseia nos dados obtidos no ajuste de parâmetros e apresentados nas tabelas 5, 6 e 7. Nelas, é possível verificar que um maior conjunto de parâmetros consegue obter o máximo de energia da seqüência utilizada para avaliação. Pode-se verificar também que não existem saltos abruptos no espaço de busca, como no caso do aumento de β do modelo de reconstrução.

Um fato a salientar é o desvio-padrão na seqüência de ajuste de parâmetros se manter constante com α. Isto é, para um determinado valor de α não importando o valor de β o desvio-padrão terá o mesmo valor, sendo a média diferente. Pode-se considerar que, como α controla a exploração do espaço de busca a ser analisado através dos resultados anteriores, o algoritmo não convergiu em todas as rodadas, sendo o desvio-padrão maior desta forma. Isto esta de acordo com BONABEAU, DORIGO e THERAULAZ (2001) onde se afirma que apesar da estigmergia produzir um resultado para o caminho analisado um ACO não converge totalmente para este caminho.

Como os resultados obtidos no ajuste de parâmetros para o dobramento de proteínas foram próximos entre si, foram escolhidos dois valores de α e β para a análise seguinte que foi apresentada nas tabelas 6 e 7. Ao final desta série de teste, obteve-se o conjunto de parâmetros considerado ideal.

O método proposto foi avaliado com as outras seqüências que foram apresentadas nos trabalhos de SHMYGELSKA, HERNÁNDEZ e HOOS (2002) e SHMYGELSKA e HOOS (2003) e comparadas com os resultados apresentados nestes trabalhos. Com a comparação, é possível verificar que o método proposto atinge os valores máximos e tem um desempenho muito melhor do que os outros métodos baseados em ACO para instâncias com um grande número de resíduos.

Este fato é bastante significativo visto que as proteínas reais têm, em geral, um número de resíduos maior do que aquelas seqüências avaliadas, sendo necessário um

método que além de robusto tenha um tempo de processamento aceitável, como o modelo apresentado.

Pode-se considerar como um dos principais fatores para a diminuição do tempo de processamento em instâncias grandes a maneira de tratar um resíduo que não tenha mais espaço na treliça. Como visto na Metodologia, o algoritmo proposto, ao encontrar um local na grade onde não pode realizar mais nenhum movimento, retorna apenas um resíduo e informa que aquele movimento é ilegal, enquanto que nos algoritmos propostos na literatura quando este fato ocorre desfazem metade do dobramento atual. Este método não prejudica seqüências com poucos resíduos, mas provoca um grande dano no dobramento de seqüências grandes.

Na tabela 8, são apresentadas ainda outras sete seqüências encontradas. Destas 2 seqüências (9 e 10) não foram utilizadas em outros trabalhos e não é possível fazer a comparação com outros métodos. Já outras 5 seqüências não tem valores mínimos de energia conhecidos sendo que desta forma uma comparação com outros métodos não é confiável.

8.2 CONCLUSÃO

Este trabalho apresentou duas novas metodologias baseadas na otimização por colônia de formigas para dois problemas de bioinformática: a reconstrução de árvores filogenéticas e o dobramento de proteínas. Como fruto deste trabalho foram desenvolvidos dois softwares para avaliar os modelos propostos: o PhyloAnt para reconstrução de árvores e o AntFold para o dobramento de proteínas.

O modelo de ACO para o problema de reconstrução de árvores filogenéticas foi desenvolvido a partir da constatação que o menor escore de uma árvore pode ser obtida através de um TSP circular, este fato foi apresentado por KOROTENSKY e GONNET(2000), e utiliza a matriz de feromônios para a obtenção das ligações do ramo. Desta forma, o modelo apresentado teve um tempo de processamento menor que algoritmos que utilizam o mesmo tipo de dados de entrada (*Neighbor-Joining* e *Fitch*), e pressupõe-se que também obtenha melhores resultados que outros modelos baseados no ACO, pois estes têm etapas de pré ou pós-processamento que não são efetuadas no modelo desenvolvido.

O método desenvolvido apresentou tempo de processamento equivalente que outros algoritmos baseados em matriz de distâncias avaliados para instâncias com um

pequeno número de espécies e foi extraordinariamente mais rápido quando o número de espécies é grande, acima de 200 espécies.

Além disto, o modelo proposto apresentou distâncias topológicas menores em relação a árvore consenso do que os outros algoritmos avaliados com o mesmo conjunto de dados.

Os resultados mostram que o método desenvolvido só obteve resultado pior que um modelo iterativo e probabilístico, que se torna inviável quando é necessário analisar um grande número de espécies.

O modelo proposto de dobramento de proteínas foi desenvolvido com uma metodologia diferente as comumente utilizadas para a aplicação do ACO. Neste modelo. Foi proposto utilizar uma estrutura que não tem conexões fixas desde o início. Porém a estrutura proposta ainda segue a premissa básica de movimentação de formigas. Além disto, ao contrário dos modelos de ACO da literatura que retornam metade da conformação quando esta encontra um ponto onde não é mais possível realizar o dobramento, no modelo proposto apenas retorna-se o número mínimo de aminoácidos para que se possa continuar o dobramento. Esta metodologia provou ser mais interessante quando utilizada com seqüências com mais de 50 aminoácidos, onde o tempo de processamento obtido pelo modelo proposto é muito inferior ao modelo proposto na literatura.

O modelo apresentado para o dobramento de proteínas também obteve melhores resultados quando comparado com metodologias atuais que utilizam o mesmo princípio do ACO.

Além disto, foram propostos recursos especiais em relação ao modelo básico, como a utilização de uma quantidade maior de dados heurísticos para avaliar o posicionamento do próximo aminoácido na conformação. Os resultados apresentados pelo algoritmo foram melhores quando foram utilizados estes recursos especiais. Imagina-se que estes recursos permitem explorar com mais eficiência o espaço local de busca.

A principal contribuição deste trabalho é apresentar novas metodologias através da meta-heurística do ACO para problemas da bioinformática onde ainda não é muito difundida. Além disto, a construção de recursos especiais para trabalhar com cada um dos problemas específicos possibilitou uma melhora nos resultados obtidos. Por último, a disponibilização de dois softwares desenvolvidos, de fácil utilização e bastante eficientes, poderá fomentar a pesquisa em bioinformática.

8.3 Trabalhos Futuros

A criação de um conjunto de *benchmark* para avaliação de algoritmos de reconstrução de árvores através de matrizes de distância seria interessante para reavaliar a capacidade do algoritmo e seu tempo de processamento, visto que os testes realizados foram baseados em dados e árvores da literatura atual.

Quanto à reconstrução de árvores poderia ser alterado o modelo para trabalhar não somente com matrizes de distâncias, mas também com os métodos da máxima parcimônia e máxima verossimilhança, o que permitiria uma maior confiabilidade ao algoritmo.

A construção de parâmetros auto-adaptativos, nos dois sistemas, que variam conforme a evolução dos resultados obtidos teria uma grande valia evitando a tarefa de descobrir o melhor conjunto de parâmetros para cada problema em particular. Também seria interessante implementar uma forma de controle para o usuário que permita parar no meio de uma execução, alterar parâmetros ou salvar resultados intermediários e continuar posteriormente.

Quanto aos softwares desenvolvidos seria muito interessante a implementação de duas funcionalidades, que são: suporte multiplataforma visto que atualmente os sistemas só rodam em máquinas com Windows, e implementação de processamento paralelo que permitiria a análise de seqüências ainda maiores, por diminuir o tempo de processamento e assim permitir o aproveitamento do tempo ocioso de diversas máquinas de uma rede.

Por último, o trabalho mostrou que a metodologia do ACO pode obter bons resultados para problemas da bioinformática, sendo interessante à construção de modelos para outros problemas tais como o alinhamento de seqüências.

REFERÊNCIAS BIBLIOGRÁFICAS

AGARWALA, R., BATZOGLOU, S., DACÍK, V., DECATUR, S. E., FARACH, M., HANNENHALLI, S., SKIENA, S. Local rules for protein folding on a triangular lattice and generalized hydrophobicity in the HP model. **Journal of Computational Biology**, v. 4, p. 275–296, 1997.

AMABIS, J. M., MARTHO, G. R. **Fundamentos da Biologia Moderna**. 1ª ed. – São Paulo: Moderna, 1990.

BERGER, B., LEIGHTON, F. T. Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete. **Journal of Computational Biology**, v. 5, p. 27–40, 1998.

BONABEAU, E., DORIGO, M., THERAULAZ, G. **Swarm Intelligence: From Natural to Artificial Intelligence**. New York: Oxford University Press, 1999.

BRANDEN, C., TOOZE, J.. **Introduction to Protein Structure**. 2ª ed. – New York: Garland Publishing, 1999.

BULLNHEIMER, B., HARTL, R.F., STRAUSS, C. Applying the ant system to the vehicle routing problem. In: Voss, S., Martello S., Osman, I.H., Roucairol, C. (Eds.), **Meta- Heuristics: Advances and Trends in Local Search Paradigms for Optimization**, Boston: Kluwer Academic Publishers, p. 285-296, 1999.

BULLNHEIMER, B., HARTL, R.F., STRAUSS, C. A new rank-based version of the ant system: a computational study. **Central European Journal of Operations Research**, v. 7, n. 1, p. 25-38, 1999.

CAMPBELL, A. MRÁZEK, J., KARLIN, S. Genome signature comparisons among prokaryote, plasmid and mitochondrial DNA. **Proceedings of the National Academy of Sciences U.S.A**., v. 96, n. 16, p. 9184–9189, 1999.

CAO, Y., JANKE, A., WADDELL, P.J., WESTERMAN, M., TAKENAKA, O., MURATA, S., OKADA, N., PÄÄBO, S., HASEGAWA, M. Conflict among individual mitochondrial proteins in resolving the phylogeny of eutherian orders. **Journal Molecular Evolution**, v. 47, p. 307-322, 1998.

CHANDRU, V., DATTASHARMA, A., KUMAR, V. S. A. The algorithmics of folding proteins on lattices. **Discrete Applied Mathematics**, v. 127, p.145–161, 2003.

COLORNI, A., DORIGO, M., MANIEZZO, V. Distributed optimization by ant colonies. **Proceedings of ECAL'91 European Conference on Artificial Life**, Amsterdam: Elsevier Publishing, p. 134-142, 1991.

CORDON, O., VIANA, I. F., HERRERA, F., MORENO, L. A new ACO model integrating evolutionary computation concepts: the best-worst ant system. **From Ant Colonies to Artificial Ants: Second International Workshop on Ant Algorithms (ANTS'2000)**, Brussels, p. 22-29, 2000.

COUNSELL, D. Bioinformatics of Protein Evolution, Part I. MRes Biomolecular Sciences Lecture Notes. Disponível em: <http://www.hgmp.mrc.ac.uk/~dcounsel/Mres/MRes2.html>. Acesso em: 30 de Novembro de 2004.

CRESCENZI, P., GOLDMAN, D., PAPADIMITRIOU, C., PICCOLBONI, A., YANNAKAKIS, M. On the complexity of protein folding. **Journal of Computational Biology**, v. 5, p. 423–465, 1998.

CSERMELY, P., SÖTI, C., KALMAR, E., PAPP, E., PATO, B., VERMES, A., SREEDHAR, A. S. Molecular chaperones, evolution and medicine. **Journal of Molecular Structure**: THEOCHEM, v. 666-667, p. 373–380, 2003.

DILL, K. A. Theory for the folding and stability of globular proteins. **Biochemistry**, v. 24, p.1501–1509, 1985.

DINNER, A. R., SALI, A., SMITH, L. J., DOBSON, C. M., KARPLUS, M. Understanding protein folding via free-energy surfaces from theory and experiment. **Trends in Biochemical Sciences**, v. 25, p. 331–339, 2000.

DOBSON, C. M., EVANS, P. A., RADFORD, S. E. Understanding how proteins fold: the lysozyme story so far. **Trends in Biochemical Sciences**, v. 19, p. 31–37, 1994.

DORIGO, M., DI CARO, G. The ant colony optimization meta-heuristic. In Corne, D., Dorigo, M., and Glover, F. (Eds.). **New Ideas in Optimization**, New York: McGraw-Hill, p. 11-32, 1999.

DORIGO, M., GAMBARDELLA, L.M. Ant colonies for the traveling salesman problem. **Biosystems**, v. 43, n. 2, p. 73-81, 1997.

DORIGO, M., GAMBARDELLA, L.M. Ant algorithm for discrete optimization. **Technical Report 98-10**, IRIDIA, Université Libre de Bruxelles, Belgium, 1998.

DUAN, Y., KOLLMAN, P. A. Computational protein folding: from lattice to all-atom. **IBM Systems Journal**, v. 40, p. 297–309, 2001.

ELLIS, R. J., HARTL, F. U. Principles of protein folding in the cellular environment. **Current Opinion in Structural Biology**, v. 9, p. 102–110, 1999.

FELSENSTEIN, J. Evolutionary trees from DNA sequences: A maximum likelihood approach. **Journal of Molecular Evolution**, v. 17, p. 368-376, 1981.

FEO, T.A., RESENDE, M.G.C. Greedy randomized adaptive search procedures. **Journal of Global Optimization**, vol. 6, p. 109-133, 1995 .

FITCH, W. M., MARGOLIASH, E. Construction of phylogenetic trees. **Science**, v. 155, n. 760, p. 279-284, 1967.

GAMBARDELLA, L.M., DORIGO, M. Solving symmetric and asymmetric TSPs by ant colonies. **Proceedings of the IEEE Conference on Evolutionary Computation, (ICEC96)**, p. 622-627, 1996.

GAMBARDELLA, L.M., DORIGO, M. An ant colony system hybridized with a new local search for the sequential ordering problem. **INFORMS Journal on Computing,** v. 12, n. 3, p. 237-255, 2000.

GUEX, N., DIEMAND, A., PEITSCH, M. C. Protein modelling for all. **Trends in Biochemical Sciences**, v. 24, p. 364–367, 1999.

HARTL, F. U. Molecular chaperones in cellular protein folding. **Nature**, v. 381, p. 571–580, 1996.

HENEINE, I. F. **Biofísica Básica**. 1ª ed. – São Paulo: Livraria Atheneu, 1984.

JONES T., FORREST, S. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. **Proceedings of the 6th International Conference on Genetic Algorithms**. San Francisco, CA: Morgan Kaufman, p. 184–192, 1995.

JUKES, T. H., CANTOR, C. R. Evolution of protein molecules. In MUNRO, H. N. (Eds.), **Mammalian Protein Metabolism**, New York: Academic Press, p. 21-132, 1969.

KIMURA, M. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. **Journal of Molecular Evolution**, v. 16**,** p. 111-120, 1980.

KOROSTENSKY, C., GONNET, G.H. Using traveling salesman problem algorithms for evolutionary tree construction. **Bioinformatics***,* v. 16, n. 7, p. 619-627, 2000.

KRASNOGOR, N., PELTA, D., LOPEZ, P.M., MOCCIOLA, P., DE LA CANAL, E. Genetic Algorithms for the protein folding problem: a critical view. In C.F.E. Alpaydin, ed., **Proc. Engineering of Intelligent Systems**, ICSC Academic Press, 1998.

KUMNORKAEW, M., KU, K., RUENGLERTPANYAKUL, P. Application of ant colony optimization to evolutionary tree construction. **Proceedings of 15th Annual**

Meeting of the Thai Society for Biotechnology. Chiang Mai, Thailand, [s.p.], 2004.

LEHNINGER, A. L. **Princípios de Bioquímica**. 1ª ed. – São Paulo: Sarvier, 1991.

LI, H., HELLING, R., TANG, C., WINGREEN, N. Emergence of preferred structures in a simple model of protein folding. **Science**, v. 273, p. 666–669, 1996.

LI, M., BADGER, J. H., CHEN, X., KWONG, S., KEARNEY, P., ZHANG, H. An information based sequence distance and its application to whole mitochondrial genome phylogeny**. Bioinformatics**, v. 17, n. 2, p. 149-154, 2001.

LYNGSØ, R. B., PEDERSEN, C. N. S. Protein folding in the 2D HP model. In: **Proceedings of the 1st Journées Ouvertes: Biologie, Informatique et Mathématiques (JOBIM)**, [s.p.], 2000.

MANIEZZO, V. Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem, **INFORMS Journal of Computing,** v. 11, n. 4, p. 358-369, 1999.

MOGK, A., BUKAU, B. Molecular chaperones: structure of a protein disaggregase. **Current Biology**, v. 14, p. R78–R80, 2004.

NAKHLEH, L., ROSHAN, U., JOHN, K. S., SUN, J., WARNOW, T. The performance of phylogenetic methods on trees of bounded diameter**.** In: **Proceedings of the First International Workshop on Algorithms in Bioinformatics** (WABI), p. 214-226, 2001.

NAYAK, A., SINCLAIR, A., ZWICK, U. Spatial codes and the hardness of string folding problems. In: **Proceedings of the 9th Annual Symposium on Discrete Algorithms (SODA)**, p. 639–648, 1998.

NEI, M., KUMAR, S. **Moledular Evolution and Phylogenetics**. New York: Oxford University Press, 2000.

NGO, J. T., MARKS, J. Computational complexity of a problem in molecular structure prediction. **Protein Engineering**, v. 5, p. 313–321, 1992.

NGO, J. T., MARKS, J., KARPLUS, M. Computational complexity, protein structure prediction, and the Levinthal paradox. In: Merz Jr, K.; LeGrand, S., (eds.) **The Protein Folding Problem and Terciary Structure Prediction**. Boston: Birkhäuser, p. 433-506, 1994.

PARPINELLI, R.S., LOPES, H.S., FREITAS, A.A. An ant colony algorithm for classification rule discovery. In: Abbas, H. A., Sarker, R. A., Newton, C. S. (Eds.)

Data Mining: A Heuristic Approach. Hershey: Idea Group Publishing, p. 190-208, 2001.

PEDERSEN, C. N. S. **Algorithms in Computational Biology**. Dissertation (Ph.D. in Science) – Department of Computer Science. University of Aarhus, Denmark, 2000.

PERRETTO, M., LOPES, H.S. Reconstruction of phylogenetic trees using the ant colony optimization paradigm. In: **Proceedings of 3rd. Brazilian Workshop on Bioinfomatics**, Brasília (DF), [CD-ROM], 2004.

RABAUD, E. Phénomène social et sociétés animales, **Bibliothèque de Philosophie Contemporaine**. Librairie Felix Arcan, Paris, 1937.

REIJMERS, T.H., WEHRENS, R., DAEYAERT, F. D., LEWI, P. J., BUYDENS, L. M. C. Using genetic algorithms for the construction of phylogenetic trees: application to G-protein coupled receptor sequences. **Biosystems**, v. 49, n. 1, p. 31-43, 1999.

ROBINSON, D. F., FOULDS, L. R. Comparison of phylogenetic trees. **Mathematical Biosciences**, v. 53, p. 131-147, 1981.

RUSSELL, G.J., WALKER, P.M.B., ELTON, R.A. Doublet frequency analysis of fractionated vertebrate nuclear DNA. **Journal of Molecular Biology**, v. 108, n. 1, p. 1–23, 1976.

SAITOU, N., NEI, M. The neighbor-joining method: a new method for reconstructing phylogenetic trees. **Molecular Biology and Evolution**, v. 4, p. 406-425, 1987.

SHMYGELSKA, A., HERNÁNDEZ, R. A., HOOS, H. H. An ant colony optimization algorithm for the 2D HP protein folding problem. In: **Proceedings of the 3rd International Workshop (ANTS), Lecture Notes in Computer Science**, v. 2463, p. 40–55, 2002.

SHMYGELSKA, A., HOOS, H. H. An improved ant colony optimisation algorithm for the 2D HP protein folding problem. In: **Proceedings of the 16th Canadian Conference on Artificial Intelligence**, p. 400-417, 2003.

SOKAL, R.R., MICHENER, C.D. A statistical method for evaluating systematic relationships. **University of Kansas Bulletin**, v. 38, p. 1409-1438, 1958.

SRINIVASAN, R., ROSE, G. D. Ab initio prediction of protein structure using LINUS. **Proteins: Structure, Function, and Bioinformatics**, v. 47, p. 489–495, 2002.

STANSFIELD, W. D. **Genética**. 2ª ed. São Paulo: McGraw-Hill do Brasil, 1985.

STÜTZLE, T., HOOS, H Max-min ant system. **Future Generation Computer Systems**, v. 16, n. 8, p. 889-914, 2000.

SWOFFORD, D. L., OLSEN, G. J., WADDELL, P. J., HILLIS, D. M. Phylogenetic inference. In: Hillis, D. M., Moritz, C., Mable, B. (Eds.) **Molecular Systematics**, Sunderland: Sinauer Associates, p. 407-514, 1996.

TAMURA, K., NEI M. Estimation of the number of nucleotide substitutions in the control region of mitochondrial DNA in humans and chimpanzees. **Molecular Biology and Evolution,** v. 10, p. 512-526, 1993.

TATENO, Y., NEI, M., TAJIIMA, F. Accuracy of estimated phylogenetic trees from molecular data. I. Distantly related species. **Journal of Molecular Evolution**, v. 18, n. 6, p. 387-404, 1982.

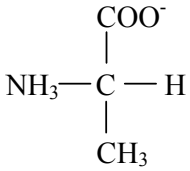
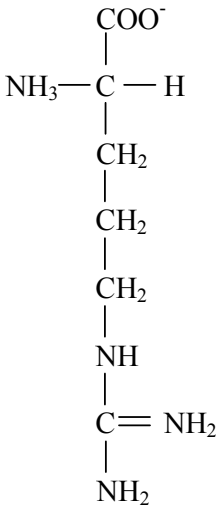
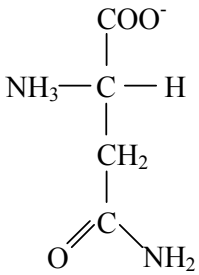
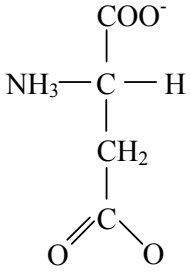
THOMASSON, W. A. B. Unraveling the mystery of protein folding: Breakthroughs in Bioscience – A series of articles for general audiences. Disponível em: <www.faseb.org/opar/protfold/protein.html >. Acesso em : 1º de Fevereiro de 2005.

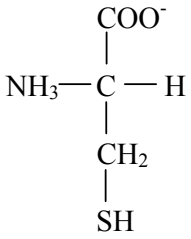
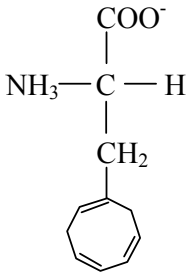
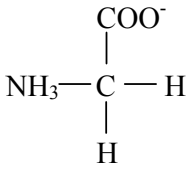
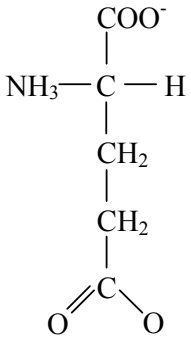
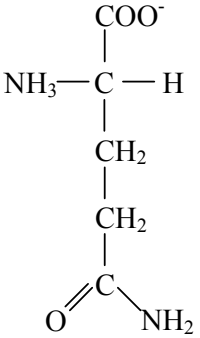
UNGER, R., MOULT, J. Finding the lowest free energy conformation of a protein is a NP-hard problem: proof and implications. **Bulletin of Mathematical Biology**, v. 55, p. 1183–1198, 1993.

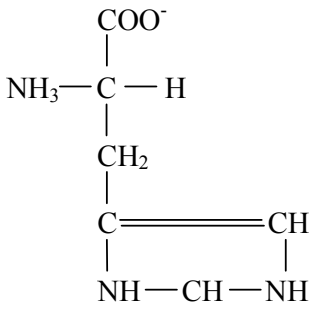
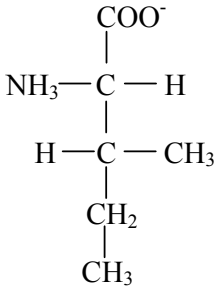
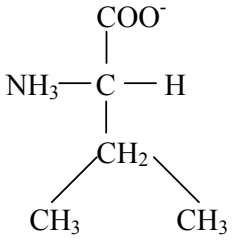
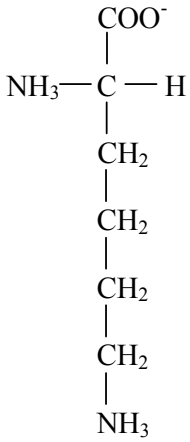
WEIR, B. S. **Genetic Data Analysis 2: Methods for Discrete Population Genetic Data**. Sunderland: Sinauer Associates, 1996.

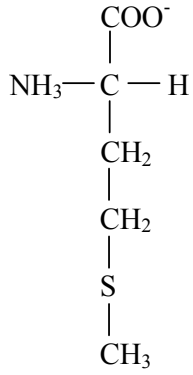
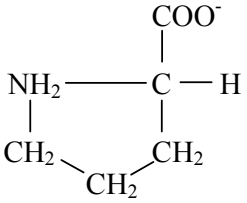
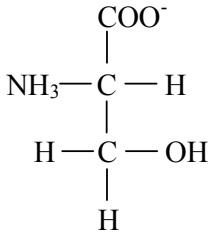
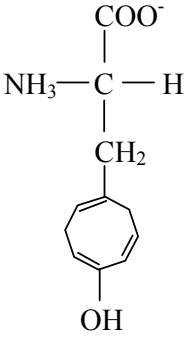
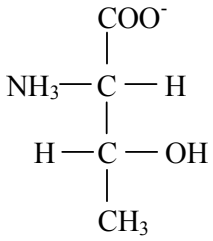
Anexo 1

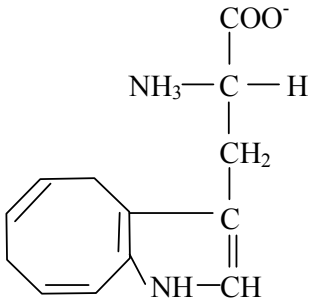
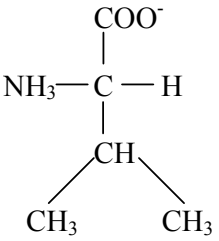
LISTA DE AMINOÁCIDOS

Aminoácido	Sigla com três letras	Sigla com uma letra	Composto químico	Hidrofóbico/ Polar
Alanina	Ala	A		H
Arginina	Arg	R		H
Asparagina	Asn	N		P
Aspartato (ácido aspártico)	Asp	D		P

Cisteína	Cis	C		P
Fenilalanina	Fen	F		H
Glicina	Gli	G		P
Glutamato (ácido glutâmico)	Glu	E		P
Glutamina (glutamida)	Gln	Q		P

Histidina	His	H		H
Isoleucina	Ile	I		H
Leucina	Leu	L		H
Lisina	Lis	K		H

Metionina	Met	M		H
Prolina	Pro	P		H
Serina	Ser	S		P
Tirosina	Tir	Y		P
Treonina	Ter	T		P

Triptofano	Trp	W		H
Valina	Val	V		H

Anexo 2

SEQÜÊNCIAS DE AMINOÁCIDOS UTILIZADAS NOS TESTES

Seqüência	Tamanho	Máximo conhecido	Seqüência HP
Teste	85	51	HHHPPPPPHHHHHHHHHHHHPPPPPPHHHHHH HHHHHHPPPHHHHHHHHHHHHPPPHHHHHHH HHHHHPPPHPPHHPPHHPPHPH
1	20	9	HPHPPHHPHPPHPHHPPHPH
2	24	9	HHPPHPPHPPHPPHPPHPPHPPHH
3	25	8	PPHPPHHPPPPHHPPPPHHPPPPHH
4	36	14	PPPHHPPHHPPPPPHHHHHHHPPHHPPPPHHPPH PP
5	48	23	PPHPPHHPPHHPPPPPHHHHHHHHHHPPPPPPH HPPHHPPHPPHHHHH
6	50	21	HHPHPHPHPHHHHPHPPPHPPPHPPPPHPPPHPP PHPHHHHPHPHPHPHH
7	60	36	PPHHHPHHHHHHHHPPPHHHHHHHHHHPHPPP HHHHHHHHHHHHHPPPPHHHHHHPHHPHP
8	64	42	HHHHHHHHHHHHPHPHPPHHPPHHPPHPPHHP PHHPPHPPHHPPHHPPHPHPHHHHHHHHHHHH
9	85	53	HHHHPPPPHHHHHHHHHHHHPPPPPPHHHHHH HHHHHHPPPHHHHHHHHHHHHPPPHHHHHHH HHHHHPPPHPPHHPPHHPPHPH
10	100	50	PPPHHPPHHHHPPHHHPHHPHHPHHHHPPPPPP PPHHHHHHPPHHHHHHPPPPPPPPPHPHHPHHH HHHHHHHHPPHHHPHHPHPPHPHHHPPPPPPH HH
11	100	48	PPPPPPHPHHPPPHHHPHHHHHPHHPPHHPPHH PHHHHHPHHHHHHHHHHHPHHPHHHHHHHPPP PPPPPPPPHHHHHHHPPHPHHHPPPPPPHPHPP
12	106	---	HPHPPPHPHHPPPHHHHPHHPPHHPHHPHHHH

			HPHHHHPHPHHHPPHHPPHPPPPPPHHPHPHHH PHHHHPHPPHHHHHPPHHHHPHPHHHPPHHHH PPHHHPHHP
13	113	---	HHPHPPPHHHPHPHPHHHHHHPPPHHHHHPHH HHHPPHPPHPHHPPHPPHHHHHHHHPPPPHHPPP PHHHHHPHHHHHPPHHHHHPHHPHPHHHHHP HHHPHHHHHPHHHPHH
14	128	---	PPPHPPPHHHHPPHHHPHHHPPHHPHHHHHHH HHPHPHPHHPPHPPHHHHHHHHPHHHHHHHP HHPPHPHHPPHHHPHPHPHHPHHHHHHHPPHH HHHHHHHHPPHHHHPHHHHHHHHHHHPPH HPP
15	330	---	PHHPHHHHPHPPHPPHHHHPHPHHHPHPHHPH PHPHHPPPHHHHPPPHHHPPHPHPHHHHPPHPH HHHHHPHHHHPHPHHPHPPPPHPHHHPHHPHH HHHHHHPHHPPHHHHPPPHHPPPHHPHHHHHH HPHHPHHPPHPPHHHHPPHHHPHPPPHHHHPH HHHHPHHHPHHHHPHHHHHHPHHHHHHPHHP HHHPHHPHHHHHHPHHHHPHPHPHHHPHHHH HHHHPHHHHPHHHHPHPHHHPPPPHHHHHHP HPHHHPPHHHHHHPHPHHHPHPHHPHHPHPHH PPPHHHHHPPPHHHHHPHHHPHHHHHHHHHH PHHHHPHHHHHPH

Anexo 3

COMPLEXIDADE DE KOLMOGOROV

A complexidade de Kolmogorov é um método, proposto por Andrei Nikolaevich Kolmogorov, sendo uma teoria algorítmica de análise da aleatoriedade. Mais especificamente ela visa descrever a seqüência através da menor estrutura algorítmica, isto é, programa possível.

Na computação a complexidade de Kolmogorov pode ser dada pela máquina de Turing *M* que a partir de uma seqüência binária de entrada *y*, tem como resultado a seqüência de saída *x*:

$$M_{p,y} = x$$

As operações típicas da máquina de Turing são:

- Ler um símbolo da fita;
- Escrever um símbolo na fita;
- Apagar um símbolo;
- Mover o cabeçote.

A complexidade condicional de Kolmogorov $C_M(x|y)$ de um número *x* com respeito a um número *y* é o tamanho da menor descrição *p* tal que $M_{p,y} = x$, sendo dada por:

$$C_M(x \mid y) = \min_{M_{p,y} = x} | p |$$

A complexidade de Kolmogorov vem sendo utilizada para a construção de compactadores para fins específicos. LI *et al*. (2001) apresentaram um compactador para seqüências de nucleotídeos baseado no teorema da complexidade de Kolmogorov. Neste trabalho, as operações da máquina de Turing foram definidas como:

- Substituição: esta operação é expressa por (R,p,char) ao qual significa substituir o caractere na posição p pelo caractere char;
- Inserção: esta operação é expressa por (I,p,char) ao qual significa inserir o caractere char após a posição p;
- Deleção: esta operação é expressa por (D,p) ao qual significa deletar o caractere na posição p;

- Cópia: esta operação é expressa por C apenas, sendo mantido o caractere na posição atual da fita.

Um exemplo simples seria a conversão da seqüência “gaccttca” em “gaccgtca” que pode ocorrer de duas maneiras distintas:

C C C C R C C C	C C C C I C D C C
g a c c g t c a	g a c c g t c a
g a c c t t c a	g a c c t t c a

No primeiro caso foi realizado apenas uma substituição, enquanto que no segundo caso foi necessário uma inserção e uma deleção. Para que seja possível realizar a transição entre as seqüências é necessário armazenar as operações realizadas sobre elas. Uma lista das operações pode ser dada por λ(u,v). Desta forma, no primeiro caso λ(“gaccttca”,“gaccgtca”)=(R,4,t). No segundo caso λ(“gaccttca”,“gaccgtca”)=(I,4,g),(D,6).

Existem diversas maneiras de codificar uma seqüência a partir de outra, sendo duas as formas principais:

- Método exato: As repetições de caracteres das duas seqüências são representadas através da quantidade em que se repete, enquanto que caracteres diferentes são inseridos na seqüência. Por exemplo: ((0,4),g,(5,3)) pode ser considerada a conversão de “gaccttca” na seqüência “gaccgtca” pois indica que os quatro primeiros caracteres devem ser repetidos, o quinto deve ser “g” e os três próximos novamente serão repetidos. Neste caso, este método necessita de dezessete bits para armazenar a segunda seqüência com base na primeira;
- Método de aproximação: Neste caso é armazenado no início das operações o tamanho da seqüência e após as operações realizadas. Desta forma, as operações seriam ((0,8),(R,4,g)). Este método precisara de quinze bits para armazenar a seqüência.

Neste segundo caso a seqüência de operações é obtida através da seqüência de bits “0 000 111 1 00 100 10”, onde R é codificado por 00, g é codificado por 10 e o 0/1 inicial indica como deve ser lido o próximo operador, em dupla de bits ou triplas, expresso no primeiro operador.

Em LI et al. (2001) é apresentado um estudo mais abrangente sobre este dois métodos e também apresenta o segundo método como o que obteve melhores resultados.

Com a definição das operações a serem realizadas e com o método de codificação de uma seqüência a partir de outra é possível definir a distância entre duas espécies através de:

$$d(x,y) = \frac{K(x) - K(x \mid y)}{K(xy)}$$

Sendo que K(x) será o tamanho da seqüência de operações que produzem x a partir de uma seqüência vazia. K(x|y) é o tamanho da seqüência de operações que produzem x com a entrada da seqüência y.

A partir disto para obter a distância entre as seqüências “gaccttca” e “gaccgtca” deve-se calcular:

K(x) = tamanho em bits de λ(“gaccttca”,“”) = 15;

K(x|y) = tamanho em bits de λ(“gaccttca”,“gaccgtca”) = 15;

K(xy) = tamanho em bits de λ(“gaccttcagaccgtca”,“”) = 7;

Com isto, temos:

d(“gaccttca”,“gaccgtca”) = (15 – 15) / 7 = 0.

Com este resultado poderia considerar que a distância entre as duas seqüências é nula, devido a pequena alteração entre elas.

Anexo 4

MÉTODO DE CÁLCULO DE DISTÂNCIA ROBINSON FOULDS

A distância obtida através do método de Robinson-Foulds é o número mínimo de operações elementares que devem ser efetuadas sobre uma das árvores para que ela se transforme na outra árvore. Como operações elementares considera-se a união de dois ramos, a separação de um ramo em dois e a troca de posições entre ramos.

Como apresentado por NAKHLEH *et al.* (2001), a distância Robinson-Foulds pode ser definida da seguinte forma: cada ramo *r* em uma árvore *A* define uma bipartição π_r na folha (induzido pela deleção do ramo *r*), e dado que A é codificado pelo conjunto C(A) = (π_r: r ε R(A)), sendo R(A) o conjunto de todos os nós internos de A. Se considerarmos A como sendo a árvore consenso, e A' a árvore constituída pelo método de reconstrução de árvores filogenéticas, então o erro topológico é calculado através da equação iv.1, e esta é considerada como a média do número de operações básicas necessárias para converter a árvore A' na árvore A.

$$D_{RF} = \frac{C(A') - C(A)}{2} \qquad \text{(iv.1)}$$

Anexo 5

MANUAL DO USUÁRIO PHYLOANT

V.1 – Introdução

O presente documento visa descrever e apresentar como utilizar o programa PhyloAnt para construção de árvores filogenéticas com um modelo desenvolvido através da otimização por colônia de formigas.

Neste documento será descrito os tipos de dados que podem ser utilizados e como deve ser configurado o sistema para seu funcionamento, acerto de parâmetros.

V.2 – Público Alvo

Pesquisadores envolvidos na área de pesquisa genética e de bioinformática que desejam utilizar um novo método para predição da estrutura de árvores filogenéticas ou que desejam conhecer este método.

V.3 – Descrição Geral

O sistema aqui apresentado visa produzir árvores filogenéticas para avaliação e validação em estudos de inferência filogenética.

As principais características do sistema são:

- Modelo de busca baseado em otimização por colônia de formigas;
- Dois tipos de dados de entrada: matriz de distância ou seqüências genômicas.
- Dois métodos para cálculos de distâncias de seqüências não-alinhadas.
- Lê e grava as seqüências de entrada e as matrizes de distância produzidas em formato FASTA, permitindo o intercâmbio de dados com outros programas;
- Permite a gravação da árvore produzida em formato próprio e a importação de árvores para comparação;
- Permite a alteração dos seus parâmetros de busca, permitindo uma maior customização ao utilizador.

V.4 – Controle de Acesso

O sistema desenvolvido não possui contas de usuário ou qualquer controle de acesso que permita a diferenciação de usuários, personalização do sistema ou proteção dos dados gerados.

V.5 – Requisitos de Desempenho

Assim como a maioria dos algoritmos de computação evolucionária a quantidade de elementos que podem ser tratados depende apenas do tempo de resposta sendo este dependente da máquina utilizada. Recomenda-se a utilização de máquinas com grande quantidade de memória RAM e processadores rápidos.

V.6 – Plataforma de Uso

Hardware

Intel Pentium III 600 MHz, 256 MB RAM, HD com 1GB de memória.

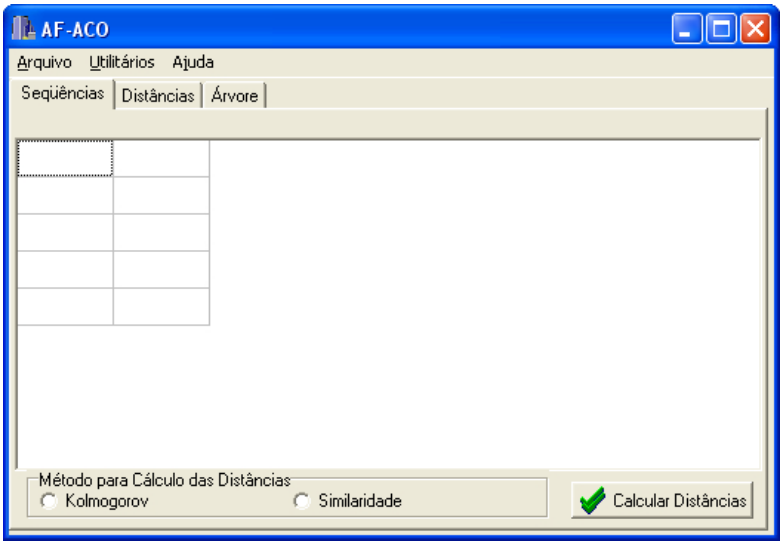
Sistema Operacional

O sistema foi desenvolvido para utilização com o sistema operacional Microsoft Windows da versão 95 em diante.

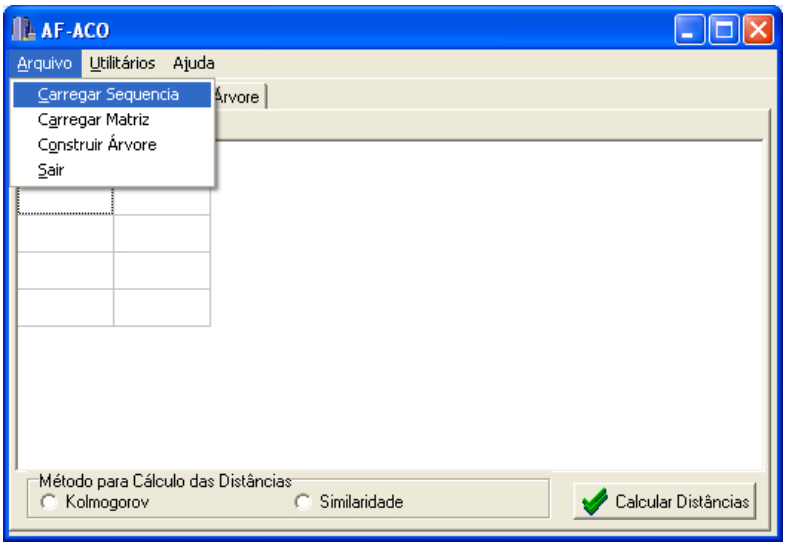
V.7 Telas do Software

V.7.1 Tela de Seqüências

Esta tela é a primeira a ser apresentada e será preenchida se o usuário selecionar um arquivo que contenha as seqüências de aminoácidos para avaliação.



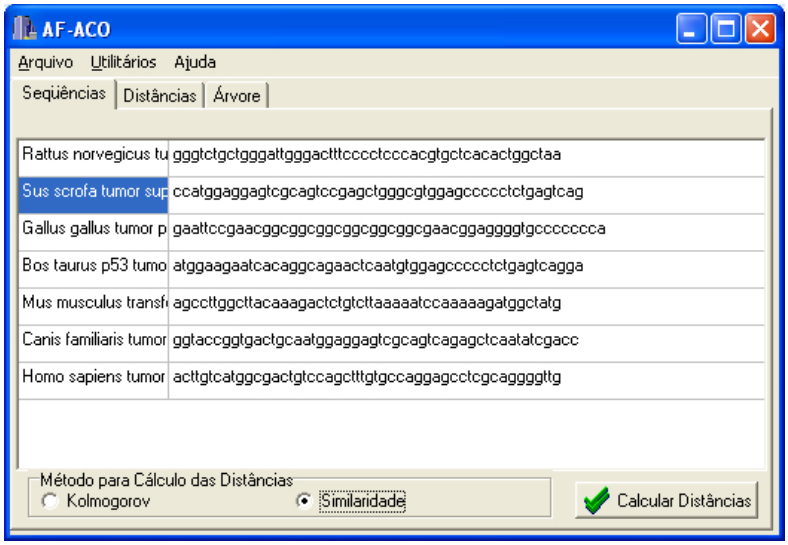
Para que o sistema leia um arquivo de entrada, seja ele de seqüências ou distâncias, deve-se selecionar a opção correspondente no menu arquivo.



Ao selecionar a opção carregar seqüência será apresentada uma tela de busca de arquivo padrão do Windows onde o usuário deverá selecionar um arquivo no formato FASTA que contenha todas as seqüências a serem analisadas.

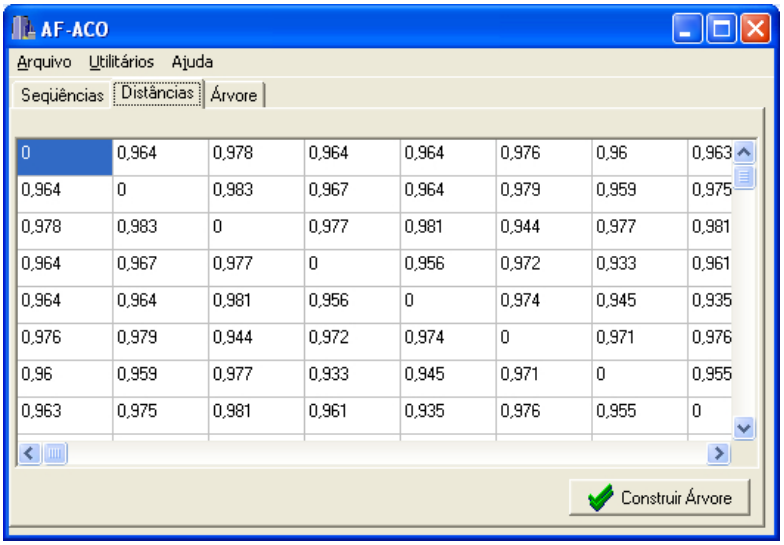
Decorrido a abertura do arquivo, será apresentada na tela uma tabela com duas colunas, na primeira estará o nome das espécies a serem analisadas e na segunda a seqüência de aminoácidos carregada do arquivo.

O usuário deverá então selecionar o método de cálculo da distância, Kolmogorov ou Similaridade, nos radio buttons apresentados na parte de baixo do programa e clicar no botão calcular distâncias.

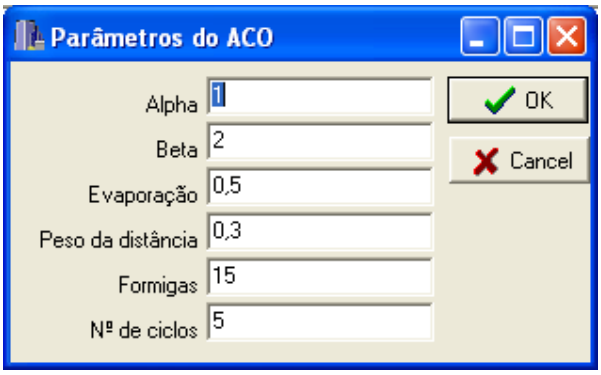


Quando terminado o cálculo de distâncias o sistema automaticamente apresentará a próxima aba, a aba de distâncias, nesta tela será apresentada a matriz de distâncias das espécies.

O usuário poderá carregar apenas a matriz de distâncias se desejar, para isso devesse selecionar a opção carregar matriz do menu arquivo. A matriz será carregada no sistema e a aba de distâncias será automaticamente selecionada.

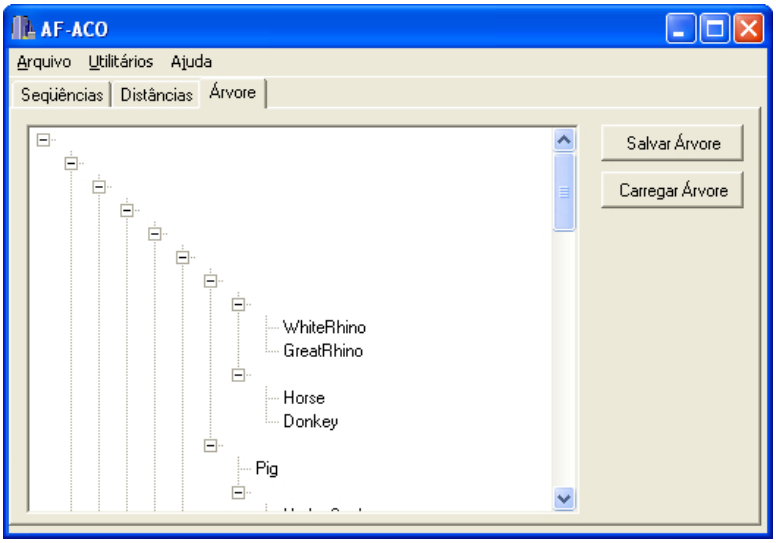


O próximo passo do usuário será a construção da árvore através do ACO para isso deve-se clicar no botão Construir Árvore da aba distâncias ou no menu arquivo. Quando clicado este botão, aparecerá a tela de definição dos parâmetros do ACO.



O usuário poderá definir novos valores para os parâmetros ou aceitar os parâmetros atualmente configurados como padrão.

Ao clicar no botão OK o software automaticamente iniciará o processo de busca da árvore. Ao final deste processo, a aba árvore será automaticamente selecionada apresentado a árvore obtida.



Nesta tela o usuário poderá salvar a árvore gerada ou apenas carregar uma árvore já calculada apenas para análise. Os arquivos gerados serão salvos com a extensão ".tree" e o software só lerá arquivos que contenham esta extensão.

Anexo 6

MANUAL DO ANTFOLDER

VI.1 – Introdução

O presente documento visa descrever e apresentar como utilizar o programa AntFolder para analisar o dobramento de proteínas com um modelo desenvolvido através da otimização por colônia de formigas.

Neste documento será descrito os tipos de dados que podem ser utilizados e como deve ser configurado o sistema para seu funcionamento, acerto de parâmetros.

VI.2 – Público Alvo

Pesquisadores envolvidos na área de pesquisa genética e de bioinformática que desejam utilizar um novo método para analisar o dobramento de proteínas buscando novas conformações que obtenham bons valores de mínimo de energia livre.

VI.3 – Descrição Geral

O sistema aqui apresentado visa realizar o dobramento de proteínas no modelo 2D HP para avaliação e validação.

As principais características do sistema são:

- Modelo de busca baseado em otimização por colônia de formigas;
- Dois tipos de dados de entrada: seqüência de resíduos ou de tipos H-P.
- Permite a gravação da seqüência de movimentos que realiza o dobramento com menor valor de energia livre;
- Permite a alteração dos seus parâmetros de busca, permitindo uma maior customização ao utilizador.

VI.4 – Controle de Acesso

O sistema desenvolvido não possui contas de usuário ou qualquer controle de acesso que permita a diferenciação de usuários, personalização do sistema ou proteção dos dados gerados.

VI.5 – Requisitos de Desempenho

Assim como a maioria dos algoritmos de computação evolucionária a quantidade de elementos que podem ser tratados depende apenas do tempo de resposta sendo este dependente da máquina utilizada. Recomenda-se a utilização de máquinas com grande quantidade de memória RAM e processadores rápidos.

VI.6 – Plataforma de Uso

Hardware

Intel Pentium III 600 MHz, 256 MB RAM, HD com 1GB de memória.

Sistema Operacional

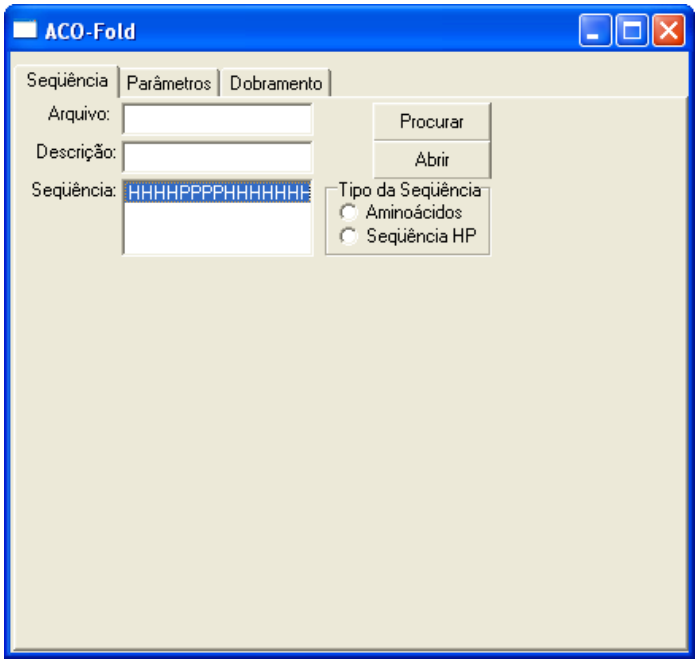
O sistema foi desenvolvido para utilização com o sistema operacional Microsoft Windows da versão 95 em diante.

VI.7 Telas do Software

V.7.1 Tela Principal

O software contém três telas: a primeira é utilizada para carregar a seqüência a ser dobrada, a segunda tem os parâmetros de configuração do ACO e a última apresenta o dobramento gerado.

Inicialmente o usuário deverá apresentar o caminho do arquivo que contém a seqüência a ser estudada. Em seguida deve-se definir se o arquivo contém uma seqüência de resíduos ou já classificada em hidrofóbicos e hidrofílicos (HP). Ao clicar em abrir a seqüência deverá aparecer na caixa seqüência, um comentário inserido no cabeçalho do arquivo antecedido pelo sinal “>” como no formato FASTA aparecerá na caixa descrição, se houver.

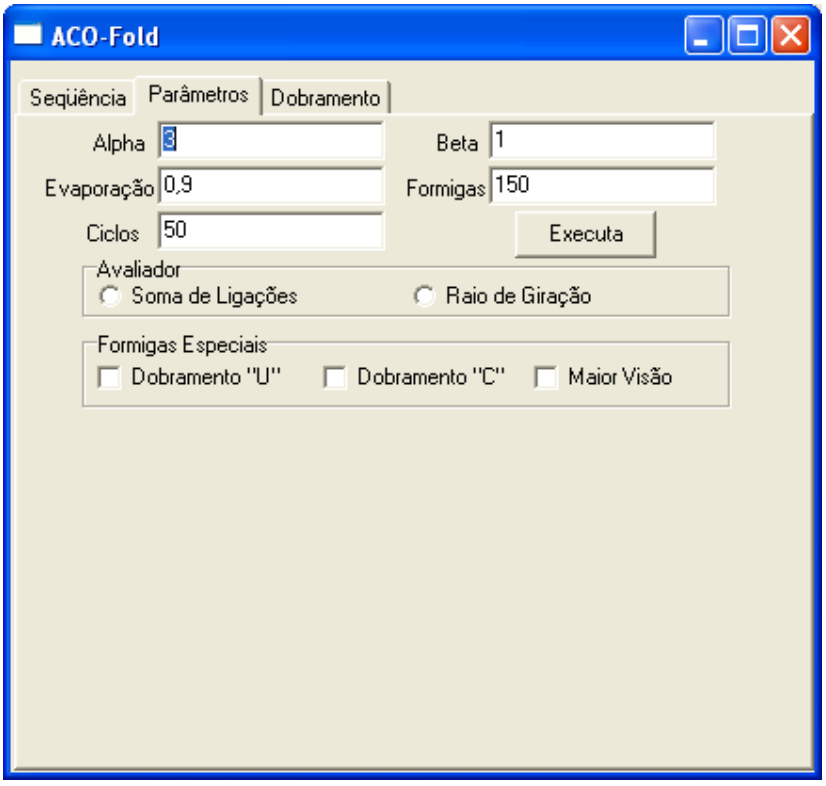


O usuário deve selecionar a próxima aba, parâmetros, onde é possível configurar os principais parâmetros do software.

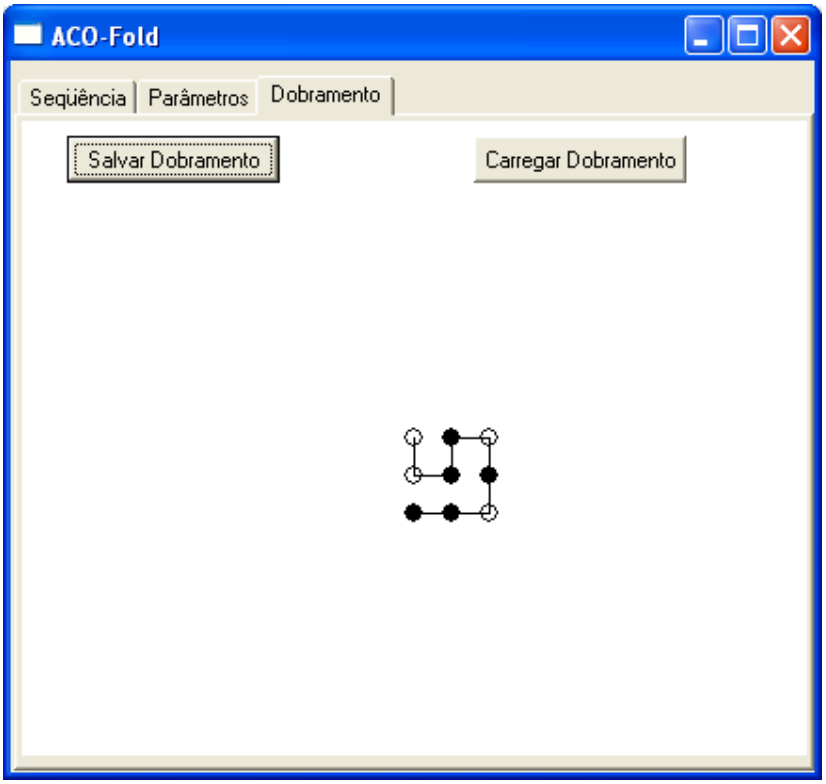
Nesta é possível configurar os parâmetros comuns ao ACO: alpha, beta, evaporação, número de formigas e de ciclos, e as alterações realizadas na melhoria do algoritmo básico. É possível definir se o dobramento gerado levará em consideração o raio de giração na hora de avaliar a qualidade da seqüência ou apenas a soma das ligações não locais.

Por último, é possível definir quais formigas especiais serão utilizadas, marcado e desmarcando as respectivas checkbox.

Ao final da configuração o usuário deverá clicar no botão “Executa” para que o algoritmo produza o dobramento.



Ao final da execução a aba “dobramento” será automaticamente selecionada apresentando o dobramento da melhor seqüência obtida.



Ao clicar no botão salvar dobramento será salva a seqüência de movimentos efetuada para obter tal dobramento, sendo um movimento por linha. Os movimentos permitidos são: adiante, direita e esquerda.

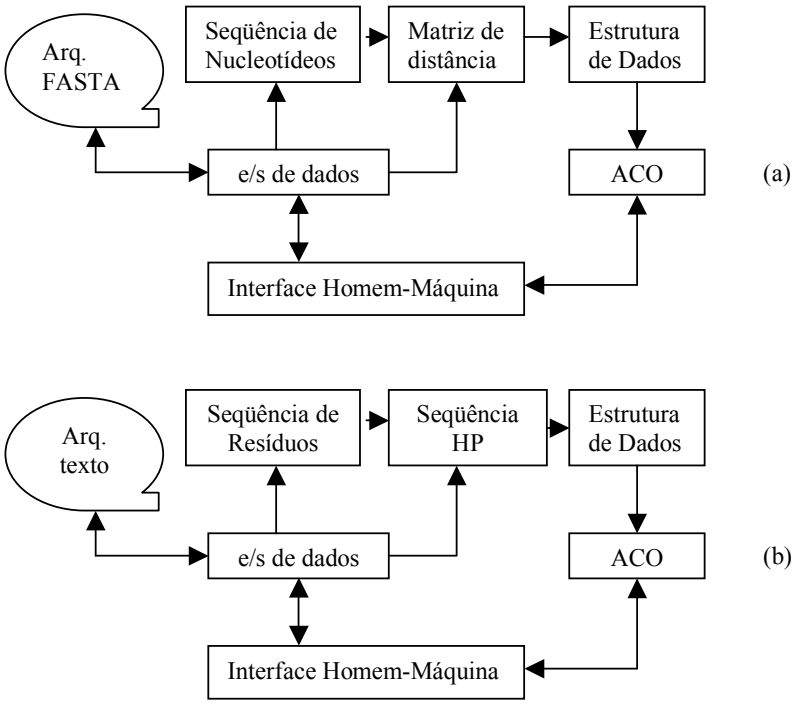
O arquivo produzido terá extensão ".fold" e deverá ser utilizada apenas arquivos com esta extensão na opção de carregar dobramento, ao selecionar esta opção o software lerá o arquivo procurando a seqüência de movimentos e a reproduzindo na tela.

ANEXO 7

IMPLEMENTAÇÃO DOS MODELOS

Para avaliar os modelos apresentados e auxiliar estudantes e pesquisadores da área de bioinformática que estejam trabalhando com a reconstrução de árvores filogenéticas ou dobramento de proteínas, foram desenvolvidos dois *softwares* sendo um para cada problema.

Na figura vii.1, são apresentados os diagramas de blocos funcionais dos



softwares de reconstrução de árvores filogenéticas e do dobramento de proteínas respectivamente.

figura vii.1 - Diagrama em blocos funcionais dos dois *softwares* desenvolvidos

VII.1 *Software* de reconstrução de árvores filogenéticas

Para o caso da reconstrução de árvores foram criadas algumas classes para tratamento dos dados de entrada e uma classe específica com a estrutura original do ACO. Na figura vii.2 é apresentada uma descrição das classes que visa esclarecer a estrutura de dados interna do *software*.

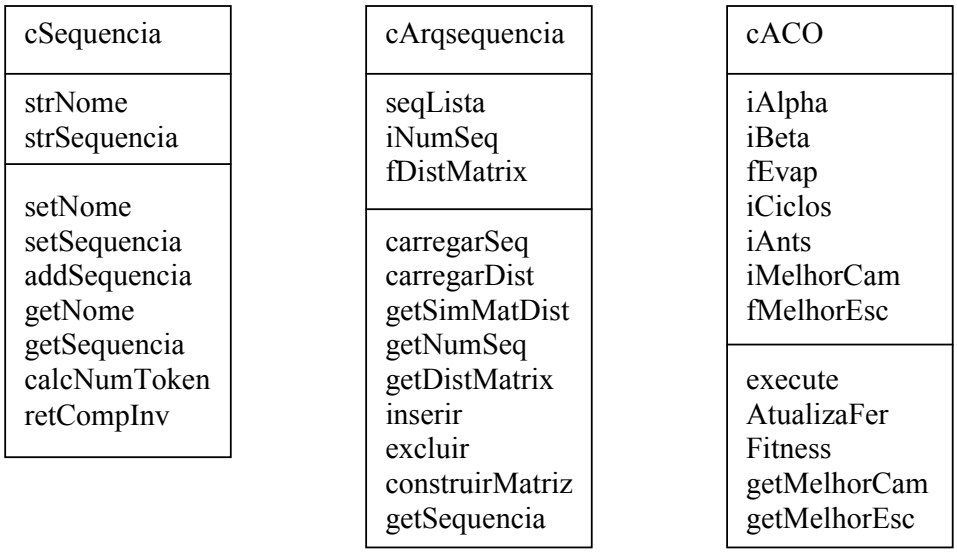


figura vii.2 - Classes desenvolvidas para tratamento interno dos dados no *software* de reconstrução de árvores filogenéticas

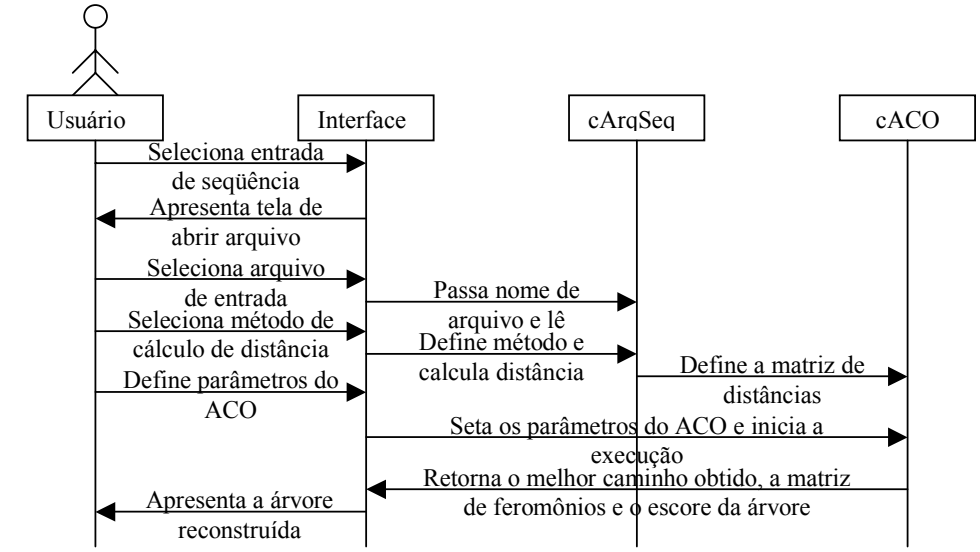
A classe cSequencia é utilizada para armazenar apenas uma seqüência de nucleotídeos quando é selecionado este método de entrada de dados. Cada instância desta classe possui dois atributos que são a descrição da seqüência, a linha começada pelo caractere “>” no formato FASTA, e a seqüência de nucleotídeos em si. Além dos métodos comuns de escrita e leitura dos atributos a classe ainda retorna o número de vezes que uma sub-seqüência ocorre na seqüência, sendo a sub-seqüência um parâmetro do método calcNumToken. A classe também retorna o complemento invertido da seqüência. Estes dois métodos foram implementados para facilitar o cálculo de distância através do método da similaridade.

A classe cArqSequencia foi desenvolvida para fazer a integração real com um arquivo FASTA. Ela contém métodos para leitura de arquivos de seqüências, carregarSeq, e arquivos de matrizes de distância, carregarMat. Esta classe contém um atributo que é um vetor de todas as seqüências que são lidas de um arquivo, tendo métodos para cálculo das distâncias evolutivas das seqüências carregadas. Por fim, ele armazena a matriz de distâncias que será utilizada no ACO.

A classe ACO, como dito anteriormente, contém a estrutura básica do ACO original. Além disto, os métodos AtualizaFer e Fitness são definidos como *friendly* e

virtual podendo ser implementados em classes derivadas ou até mesmo em métodos que estejam fora da classe permitindo a reutilização da classe ACO em outros problemas. Os métodos de getMelhorCam e getMelhorEsc retornam o melhor caminho e o maior escore respectivamente e são inerentes ao problema de reconstrução de árvores.

O *software* contém apenas um ator, o próprio usuário, e apenas três casos de uso que são: a construção da árvore a partir de arquivo de seqüências, a construção da árvore a partir de uma matriz de distâncias e a carga de uma árvore já salva para análise. Na figura vii.3, vii.4 e vii.5 são apresentados os diagramas de seqüências para os três casos de uso descritos acima.



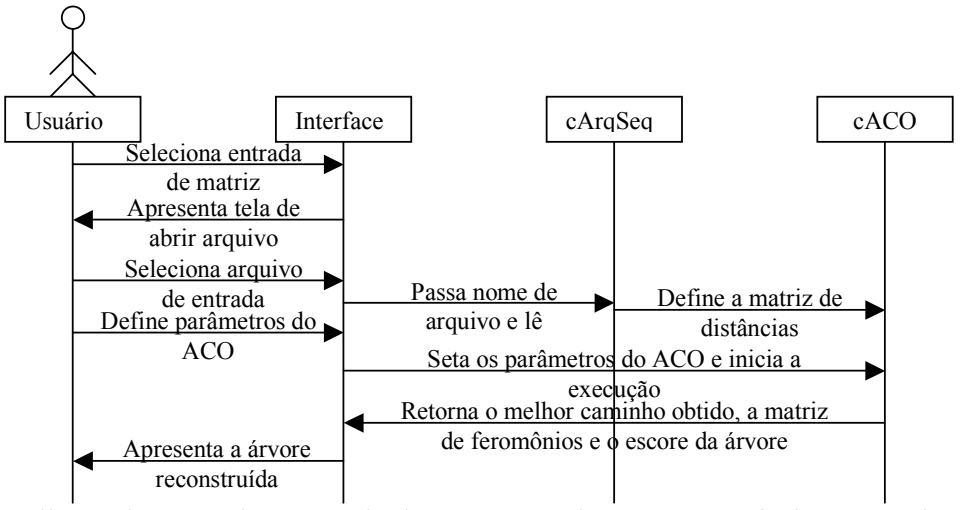


Figura vii.3 - Diagrama de seqüência da construção da árvore a partir de um arquivo de seqüências

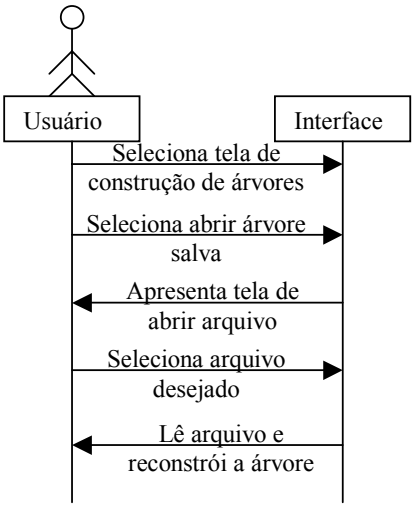


Figura vii.4 - Diagrama de seqüência da construção da árvore a partir de uma matriz de distâncias

Figura vii.5 -Diagrama de seqüência de leitura de arquivo com árvore salva

Na seqüência são apresentadas as principais telas do software disponíveis ao usuário. A primeira tela, apresentada na figura vii.6, é utilizada para apresentar as seqüências selecionadas para serem carregadas. Também é possível definir o método que será utilizado para o cálculo das distâncias evolutivas. Na figura vii.7, é apresentada a tela de matriz de distâncias, onde é possível verificar as distâncias entre as diversas espécies e definir os parâmetros de execução do ACO. Por último, na figura vii.8 é apresentada a tela de visualização das árvores obtidas.

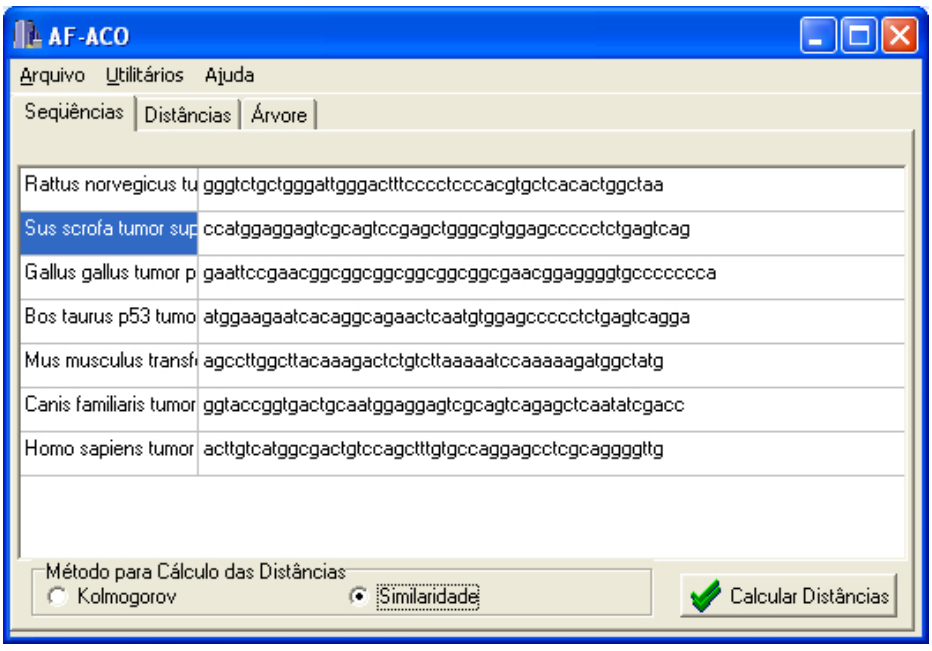


Figura vii.5 - Tela de visualização das seqüências carregadas e definição do método de cálculo de distâncias

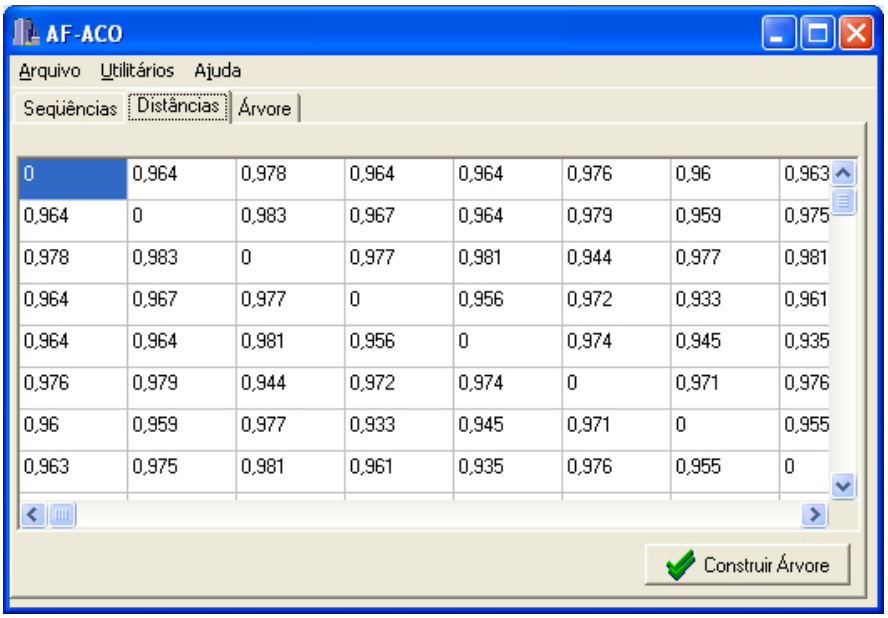


Figura vii.6 - Tela de visualização da matriz de distância, carregada ou calculada, e definição dos parâmetros de execução do ACO

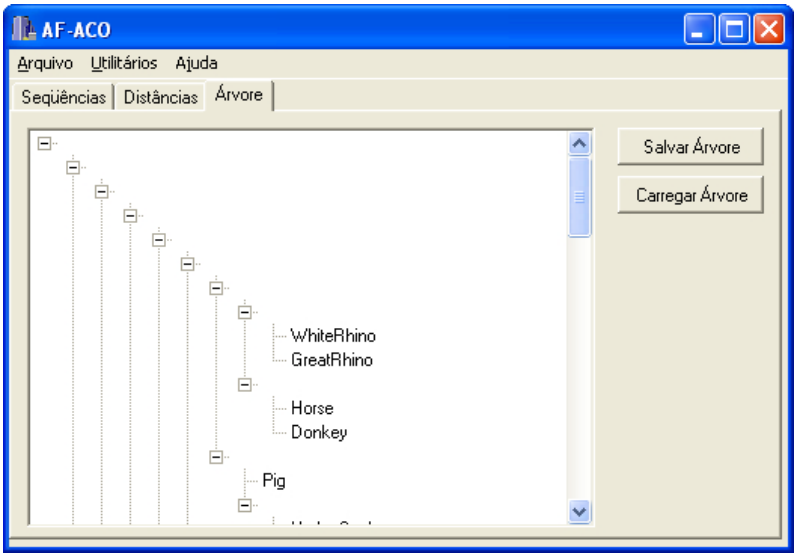


Figura vii.7 - Tela de visualização da árvore obtida com o método ou carregada a partir de um arquivo

VII.2 Software de dobramento de proteínas

O *software* desenvolvido para implementar a metodologia proposta para o dobramento de proteínas não contém classes para carregar as seqüências de dados, visto que os dados de entrada são facilmente convertidos de seqüência de resíduos para seqüência HP se este tipo de dado for selecionado, sendo utilizada apenas a classe de ACO anteriormente descrita.

A esta classe foram adicionados métodos para verificar a possibilidade de realizar um movimento. Também foi criada uma estrutura que armazena a lista de coordenadas cartesianas e o tipo de resíduo para realizar o cálculo de escore descrito na seção 3.2.3.

O *software* de dobramento de proteínas também só tem um ator, como o outro *software*. Porém, pode-se definir apenas dois casos de uso: realização de um dobramento a partir de uma seqüência e carregamento de uma seqüência de movimentos. Na figura 45 é apresentado o diagrama de seqüência para a realização de um dobramento. O carregamento de uma seqüência é similar ao carregamento de uma árvore apresentada na figura vii.8.

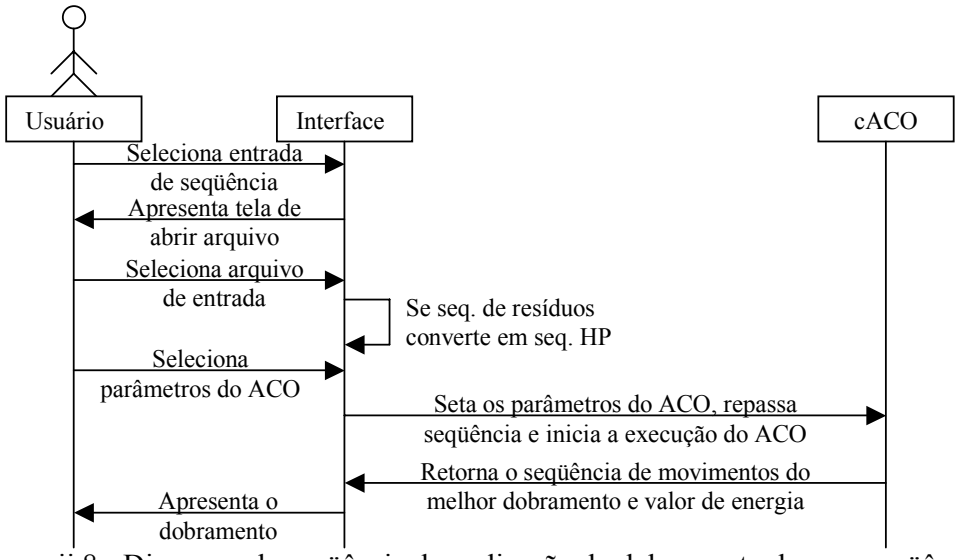


Figura vii.8 - Diagrama de seqüência de realização do dobramento de uma seqüência

Por último, são apresentadas as telas do *software*. A tela inicial, apresentada na figura vii.9, é utilizada para o carregamento da seqüência de entrada a ser analisada. Quando o usuário define uma seqüência o *software* automaticamente passa à tela da figura vii.10. Nesta tela o usuário deve selecionar os parâmetros do ACO e a opção de

utilizar os recursos das formigas especiais. Ao final da execução do ACO o dobramento é apresentado ao usuário, como mostrado na figura vii.11.

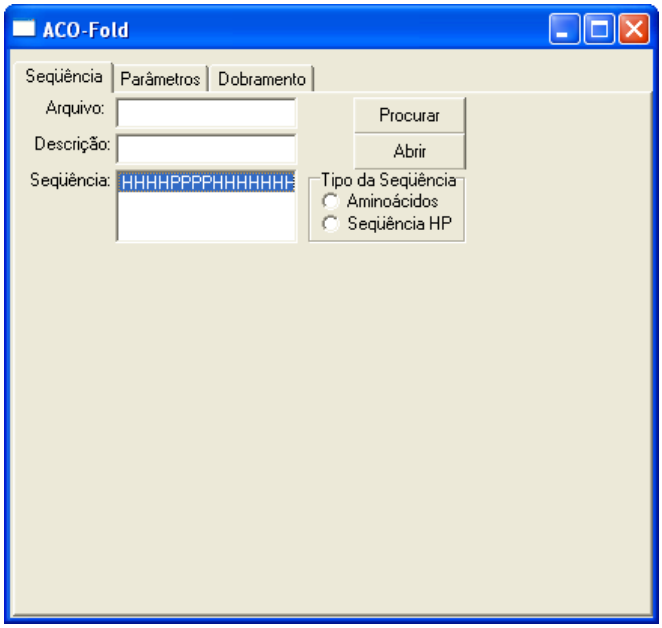


Figura vii.9 - Tela inicial do *software* de dobramento onde o usuário deve definir a seqüência a ser analisada

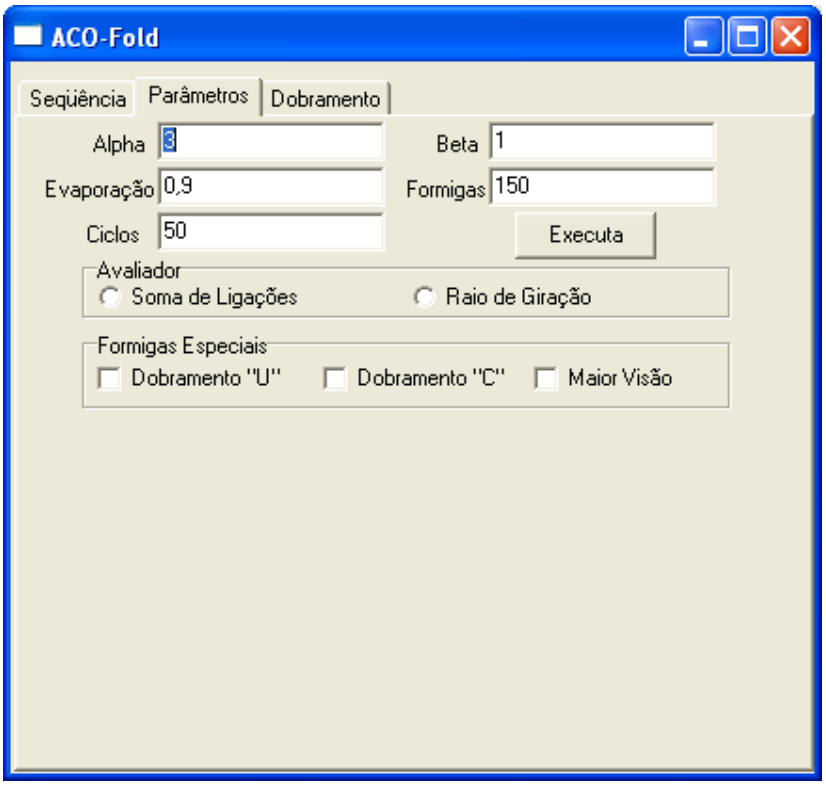


Figura vii.10 - Tela de definição dos parâmetros do ACO e seleção das formigas especiais

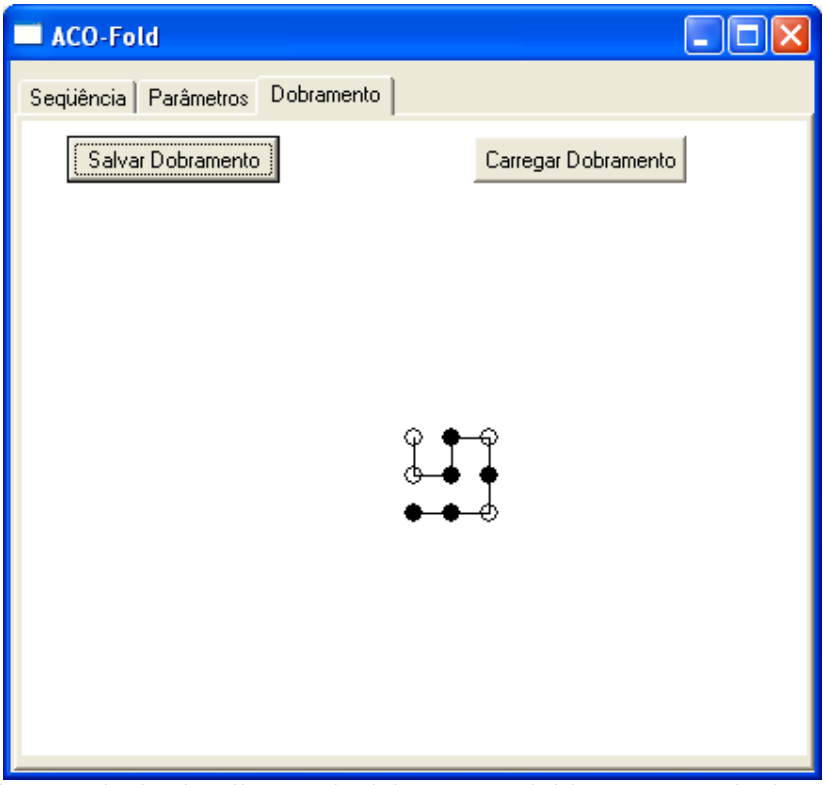


Figura vii.11 - Tela de visualização do dobramento obtido ou carregado de um arquivo

RESUMO:

O ser humano tem uma grande estima pelo processo de raciocínio que desenvolveu durante a sua evolução. Uma das áreas da computação foi desenvolvida com o objetivo inicial de simular a inteligência humana dentro de programas computacionais. Esta área ficou conhecida como inteligência artificial. Nas últimas décadas a inteligência artificial tem se baseado nas mais diversas formas de organização que tenham padrões. Um desses métodos é o algoritmo de otimização por colônias de formigas, apresentado no início da década de 90, e que apresentou bons resultados para vários problemas que tiveram modelos implementados.

A biologia molecular visa analisar as estruturas moleculares contidas nos seres vivos, dentre elas as seqüências de DNA, RNA e os aminoácidos das proteínas. Devido o grande número de informações envolvidas nessa análise torna-se inviável em termos de tempo de processamento uma busca em todo o espaço de soluções possíveis, o que torna interessante o uso de algoritmos que percorram este espaço de busca de forma eficiente.

Um dos problemas da biologia molecular é a reconstrução de árvores filogenéticas. Ele visa relacionar de forma hereditária as diversas espécies através das informações contidas em suas seqüências. Desta forma é possível saber quais espécies são mais próximas em termos evolutivos..

Outro problema é o dobramento de proteínas. Uma proteína é um polímero que pode desempenhar as mais diversas funções em um ser vivo. A função que uma proteína desempenha esta diretamente relaciona a sua forma tridimensional. Uma proteína é codificada no DNA, e sintetizada no ribossomo de uma forma linear, a partir desta forma ela se dobra sobre a sua estrutura obtendo a sua forma final. Com a compreensão deste processo, seria possível a identificação de proteínas mal formadas e até mesmo o desenvolvimento de novas proteínas com funções específicas.

O presente trabalho visa descrever dos modelos, baseados na otimização por colônia de formigas, desenvolvidos para os problemas. Além disso, foram desenvolvidos recursos especiais que permitem percorrer o espaço de busca de forma mais efetiva obtendo melhores soluções.

Os resultados obtidos com as metodologias propostas apresentaram resultados similares ou até melhores que métodos já conhecidos que utilizaram o algoritmo de otimização por colônia de formigas para os mesmos problemas.

PALAVRAS-CHAVE

Bioinformática; Otimização por Colônia de Formigas; Árvores Filogenéticas; Dobramento de Proteínas.

ÁREA/SUB-ÁREA DE CONHECIMENTO

1.03.03.04-9 Sistemas de Informação

2.08.04.00-8 Biologia Molecular

2005

N°: 360