CLIVAGEM DE PROTEÍNAS COM APLICAÇÃO DE GA

Ricardo Linden[1]

Margarete da Silva Ramos[2]

Carolina Viana[3]

Resumo:
Este estudo mostra a importância dos algoritmos genéticos na aplicação de problemas computacionais extremamente difíceis de resolver por possuir um número impraticavelmente grande de soluções. Os algoritmos genéticos - GA inspiram-se na natureza para gerar soluções ótimas para problemas bastante difíceis de serem resolvidos computacionalmente no qual uma população de indivíduos é criada e submetida aos operadores genéticos: seleção, cruzamento e mutação de forma a gerar um processo similar à evolução natural destes chegando a uma boa solução do problema em questão. Um problema extremamente interessante e complexo é a clivagem de proteínas, no qual se quer é encontrar regras de combinações de aminoácidos que agrupem várias sequências de proteínas. Este é um problema de muitas soluções, porque o número de combinações posição/aminoácido é proporcional à fatorial do número das posições e dos aminoácidos. Seguindo as orientações da teoria da evolução uma família de algoritmos é utilizada para resolver problemas. As estruturas são organizadas seguindo um modelo abstrato de dados e o teste é feito com uma sequência fictícia.

Palavras-chave: algoritmos genéticos, computação, proteínas.

Abstract:
This study shows the importance of genetic algorithms in the application of computational problems extremely difficult to resolve due to an impractically large number of solutions. The genetic algorithms - GA are based on nature to generate optimal solutions to difficult problems to be solved computationally in which a population of individuals is created and submitted to genetic operators: selection, crossover and mutation in order to generate a process similar to the evolution these natural reaching a satisfactory solution of the problem in question. An extremely interesting and complex problem is the cleavage of proteins, which either is to find rules that involve combinations of amino acid sequences of various proteins. This is a problem with many solutions, because the number of combinations position / amino acid is proportional to the factorial of the number of positions and amino acids. Following the guidelines of the theory of evolution is a family of algorithms used to solve problems. The structures are organized following an abstract model of data and the test is done with a sequence fictitious.

Key-words: Genetic Algorithms, computation, proteins.

1 Dr Sc Professor da FSMA e pesquisador do CEPEL.
2 Analista de Sistema e professora da FSMA.
3 Analista de Sistemas

INTRODUÇÃO

Nas últimas décadas o homem tem se preocupado mais com a qualidade de vida, essencial para a preservação da espécie, e com a busca de conhecimento sobre os mecanismos que interferem nos ciclos vitais de nosso organismo. Na busca deste objetivo têm sido desenvolvidas várias áreas de estudo, tais como a computação bioinspirada, a engenharia bioinspirada e a bioinformática. A bioinformática se constitui de pesquisa e desenvolvimento de ferramentas computacionais, matemáticas e estatísticas para a resolução de problemas da Biologia, e tornou-se um campo de pesquisa de grande importância e de muitas possibilidades.

Seu desenvolvimento envolve conhecimento integrado de diversas áreas de estudo, o que a torna extremamente interessante. Projetos de pesquisa nessa área fazem com que cada vez mais haja uma demanda de pessoas que tenham esse conhecimento multidisciplinar.

O desenvolvimento do trabalho consiste na fundamentação da ideia de aplicação de GA na clivagem de proteínas, a descrição do desenvolvimento do programa que norteia as funções para aplicação de GA na clivagem de proteínas e o resultado prático do trabalho.

1. REVISÃO CONCEITUAL

Por muitos anos, sub-áreas da biologia têm inspirado técnicas de computação, como, por exemplo, as redes neurais, os GAs, a programação genética e a vida artificial (LEON, 2003). Agora, estas técnicas estão sendo utilizadas para resolver problemas da Biologia, como é o caso de algumas aplicações em Bioinformática.

As pesquisas em Bioinformática começaram na década de 1960, quando foram mapeadas as primeiras bases de dados de sequências de aminoácidos. Entre as décadas de 1960 e 1970, pesquisadores desenvolvem algoritmos específicos para analisar esses dados. Em 1980, o *GenBank* e outras bases de dados públicas foram disponibilizadas, junto com ferramentas de análise. Na década de 1990, houve um enorme crescimento das bases de dados, tanto do *GenBank* como do PDB artificial (LEON, 2003).

O termo Bioinformática é descrito por vários autores em termos de ações que auxiliam o trabalho da biologia moderna no que tange a manipulação de substancial massa de dados gerados pelo avanço da tecnologia. A seguir algumas definições.

- Aplicação de métodos para a coleção, organização, indexação, armazenamento e análise de sequências biológicas (ADN, ARN e Proteínas), e a derivação do conhecimento relativo à localização, função e interação dos genes (genoma funcional) e das proteínas.

- Pesquisa e desenvolvimento de ferramentas computacionais, matemáticas e estatísticas para a resolução de problemas da Biologia (LEON, 2003)

- Aplicação de técnicas computacionais para compreender e organizar as informações associadas com macromoléculas biológicas (SCHATTAUER GMBH, 2001).

Hoje em dia, dados biológicos podem ser gerados com fenomenal rapidez: estruturas gênicas, protéicas e as interações entre elas gerando um enorme volume de dados. Esses dados somente podem ser manipulados através de computadores, o que sem estes seria impossível se conceber tanto desenvolvimento do conhecimento biológico. Como resultado deste potencial, a informática tornou-se indispensável para a pesquisa biológica (SCHATTAUER GMBH, 2001).

O uso da bioinformática é imprescindível para atender aos desafios científicos oriundos do gigantesco volume de dados produzidos pelos atuais projetos na área biológica. Nesta era pós-genômica, o desafio é determinar a função e o papel biológico de cada uma das sequências de

proteínas que forem determinadas. Se os genes são os portadores das instruções que permitem a "construção" de um determinado organismo, as proteínas são as responsáveis por sua estrutura e funcionamento. No homem, são essas moléculas que formam os ossos, músculos e os demais tecidos e comandam o metabolismo. Defeitos ou mau funcionamento delas podem causar várias doenças (LEON, 2003).

A bioinformática traz em suas pesquisas benefícios para várias áreas, tais como a Medicina, a Farmácia e a Agricultura. Na Medicina permite uma melhora no diagnóstico de doenças, facilitando a detecção para predisposições genéticas a doenças, possibilita a criação de medicamentos baseados em informações moleculares, permite a utilização de terapias genéticas como remédios e torna possível o desenvolvimento de “drogas personalizadas”, baseadas no perfil gênico individual.

1.1. DEFINIÇÕES BIOLÓGICAS

1.1.1 DOGMA CENTRAL

O Dogma Central da Biologia Molecular foi descrito em 1958 por *Francis Crick* na tentativa de relacionar o DNA, o RNA e as proteínas. O DNA pode se replicar e dar origem a novas moléculas de DNA, pode ainda ser transcrito em RNA, e este por sua vez traduz o código genético em proteínas (Figura 01).

Figura 1 - Dogma Central

A informação genética contida na sequência de nucleotídeos da molécula de RNA origina uma fita de RNA que, por sua vez, vai dar origem à sequência de uma determinada proteína. A relação entre as bases nitrogenadas do DNA e os aminoácidos é chamada de código genético, que constitui a base da vida e da evolução das espécies em nosso planeta (POIAN, 2004).

Essa foi uma outra maneira de dizer que os organismos são ligados à sua formação genética, negando que o meio-ambiente tenha o mínimo de influência na estrutura e na função dos genes.

1.1.2. GENES

São as estruturas que carregam informação para produzir as proteínas requeridas por todos os organismos. São subsequências de DNA localizadas no cromossomo. Servem como “molde” para a produção de proteínas. Encaixadas entre os genes estão segmentos chamados de regiões não codificadoras.

1.1.3. PROTEÍNAS

As Proteínas desempenham funções diversas como a de catalisadores das reações orgânicas (as enzimas), de esqueleto, de suporte das células e dos organismos, como a queratina, de

reguladores da expressão dos genes ou ainda de transportadores de oxigênio, como a hemoglobina. As proteínas actina, miosina e troponina formam a fibra muscular, atuam na contração dos músculos e em todos os nossos movimentos. Os anticorpos também são proteínas. Elas também têm papel importante na fotossíntese dos seres autrotóficos, pois esse processo ocorre devido à presença de pigmentos que ficam ligados às proteínas.

As proteínas são bio-polímeros formadas essencialmente por aminoácidos ligados entre si, em sequência linear específica, através de ligações peptídicas (Figura 02). Como biopolímeros pode-se entender moléculas que fazem parte dos seres vivos (bio=vida), que possuem em sua composição o carbono como elemento principal e que são formados pela polimerização de diversas unidades monoméricas, e por isso, são considerados macromoléculas (POIAN, 2004).

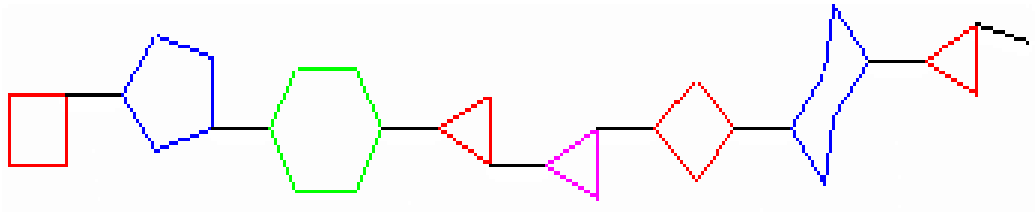


Figura 2 - Modelo Estrutural de uma Proteína

Os aminoácidos que intervêm na composição das proteínas são constituídos de um grupamento amina, uma carboxila, um átomo de hidrogênio e um radical R diferenciado, ligado a um átomo de carbono, que é chamado de carbono alfa, por ser o adjacente ao grupamento carboxila (Figura 03).

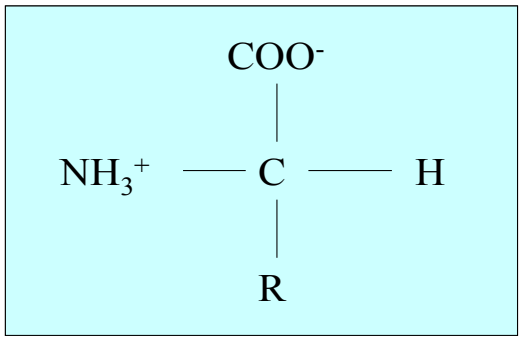


Figura 3 – Modelo Estrutural de um Aminoácido

Existem vinte aminoácidos que podem ser encontrados em todas as proteínas, seja ela de uma bactéria, planta ou mamífero. Entretanto podemos encontrar outros aminoácidos presentes em grupos restritos de proteínas. Eles são modificados por enzimas específicas após estarem incorporados às proteínas correspondentes para auxiliarem em processos metabólicos importantes dos ciclos vitais.

Os 20 aminoácidos encontrados nas proteínas diferem entre si através de suas cadeias laterais ou grupos R, os quais variam em estrutura, tamanho e carga elétrica, e influenciam a solubilidade do aminoácido em água (Quadro 01).

QUADRO 1 - Aminoácidos presentes nas proteínas dos seres vivos

Símbolo	Abreviação	Nome	Símbolo	Abreviação	Nome
A	Ala	Alanina	L	Leu	Leucina
B	Asx	Asparagina ou Aspartato	M	Met	Metionina
C	Cis ouCys	Cisteana	N	Asn	Asparagina
D	Asp	Aspartato (Ácido aspartico)	P	Pro	Prolina
E	Glu	Glutamato (Ácido glutamico)	Q	Gln	Glutamina (Glutamida)
F	Fen ou Phe	Fenilalanina	R	Arg	Arginina
G	Gli ou Gly	Glicina	S	Ser	Serina
H	His	Histidina	T	Tre ou Thr	Treonina
I	Ile	Isoleucina	V	Val	Valina
K	Lis ou Lys	Lisina	W	Trp	Triptofano (Triptofana)
Y	Tir ou Tyr	Tirosina	Z	Glx	Glutamina ou Glutamato

Fonte:POIAN – Apostila de bioquímica – Fundação CECIERJ

A sequência de aminoácidos é o elo entre a mensagem genética no DNA e a estrutura tridimensional que executa a função biológica de uma proteína. Ela revela muito de sua história evolutiva. As proteínas só se assemelham umas às outras, em sequência de aminoácidos, se tiverem um ancestral comum. Conseqüentemente, a paleontologia molecular é uma área florescente de pesquisa.

As Proteínas são estabilizadas por ligações e interações que se estabelecem, não só entre elementos da própria cadeia, como ainda com o meio circundante, de onde dependem as propriedades físico-químicas e a atividade biológica. Elas também possuem uma estrutura primária, secundária, terciária e quaternária. Os níveis de organização podem ser descritos resumidamente (POIAN, 2004).

- Estrutura primária: é a sequência linear dos aminoácidos da proteína. Considerando a simplicidade, este trabalho utiliza esta estrutura na modelagem do problema.
- Estrutura secundária: é a maneira como esses aminoácidos se organizam no espaço. Os elementos de estruturas secundárias mais comuns são as hélices, fita e voltas;
- Estrutura terciária: é a maneira como a proteína se organiza no espaço tridimensional, isto é, o movimento, a organização dos hélices, fitas e voltas no espaço tridimensional;
- Estrutura quaternária: é quando a proteína tem mais de uma subunidade, isto é, formas dímeros, trímeros, tretámeros e oligômeros.

As proteínas podem ser diferenciadas da seguintes maneiras:

- Pelo número de aminoácidos: uma proteína A é formada por 610 aminoácidos de determinados tipos e ordenados numa certa sequência. Uma proteína B é formada pelos

mesmos tipos de aminoácidos, na mesma sequência, mas em número de 611. A proteína B será diferente da A apenas por conter uma unidade a mais.

- Pelo tipo de aminoácidos: uma proteína C apresenta, num certo trecho de sua molécula, aminoácidos tais corno valina, glicina, leucina, triptofano, treonina, alanina e arginina. Uma proteína D, formada pelo mesmo número de aminoácidos e na mesma sequência que a proteína C, apresenta nesse trecho os aminoácidos valina, glicina, isoleucina, triptofano, treonina, alanina e arginina. Apenas pelo fato de na proteína C haver leucina no trecho de molécula considerado, as proteínas C o D são diferentes.
- Pela sequência dos aminoácidos; uma proteína é formada, em determinado trecho de sua molécula, pelos aminoácidos cisteína, serina, metionina, leucina, histidina e lisina. Uma proteína F é formada pelos mesmos aminoácidos, mas, no trecho em exame, há uma inversão na posição de dois deles; cisteína, metionina, serina, leucina, hístidina e lisina. Por causa disso, as proteínas E e F são diferentes.

Conclui-se, então, que podendo repetir-se à vontade os 20 tipos de aminoácidos e, ainda, combinando-se de várias formas a partir das diferenças que acabamos de examinar, uma célula pode produzir um número praticamente infinito de proteínas diferentes.

2 COMPUTAÇÃO EVOLUCIONÁRIA

Na década de 50, alguns pesquisadores procuraram na natureza inspiração para novas técnicas de busca de soluções. O motivo desse interesse deve-se ao fato da natureza conseguir resolver, de forma satisfatória, problemas altamente complexos, como a própria sobrevivência das espécies. A tentativa de imitar a evolução dos seres vivos na natureza deu origem a Computação Evolucionária (AZEVEDO, 2000).

Computação Evolucionária é o nome genérico, dado a métodos computacionais, inspirados na teoria da evolução. Os algoritmos usados em computação evolucionária são os algoritmos evolucionários.

Os algoritmos evolucionários mais conhecidos são os algoritmos genéticos, a programação Evolucionária e estratégias Evolucionárias. Todos compartilham de uma base conceitual comum, que é a Teoria da Evolução de Charles Darwin (AZEVEDO, 2000).

1.1. ALGORITMOS GENÉTICOS

Dentro de uma perspectiva histórica, os GAs foram desenvolvidos, inicialmente por John Holland, em 1975, no trabalho intitulado “Adaptation in Natural and Artificial Systems”. Holland inspirou-se no mecanismo de evolução das espécies, tendo como base os trabalhos de Darwin e na genética natural devido principalmente a Mendel. Esta teoria prevê a preservação das espécies e melhoria das gerações pela capacidade de auto organização e o comportamento adaptativo desses seres vivos (FERNANDESM, 2003).

Charles Darwin e Alfred Russel Wallace foram os precursores, através de suas evidências para a Teoria da Evolução, em 1858, na revolução tanto do pensamento biológico quanto da filosofia humana. Desde então, esta teoria é uma das mais aceitas pelo mundo científico, o chamado paradigma Neo-Darwiniano.

Charles Darwin, por volta de 1850, fez uma longa viagem a bordo do navio HMS. Ele visitou muitos lugares e sua grande habilidade de observação permitiu que ele percebesse que animais de uma mesma espécie eram ligeiramente diferentes em outros ecossistemas tornando-se

mais adaptados às necessidades e ofertas existentes neste ecossistema. Estas e outras observações culminaram na teoria da evolução da espécie, que diz que na natureza todos os indivíduos em um ecossistema, competem entre si por recursos limitados. Os indivíduos da mesma espécie que não obtêm sucesso tendem a ter uma prole menor e por isso diminui a probabilidade de seus genes serem propagados nas próximas gerações e as combinações de genes dos indivíduos vencedores têm a possibilidade de produzir indivíduos melhores adaptados ao ambiente (LINDEN, 2008).

O paradigma Neo-Darwiniano afirma que a história da existência da vida, em nosso mundo, é atribuída completamente por somente uns poucos processos estatísticos, que agem sobre populações e espécies. Estes processos são: reprodução, mutação, competição e seleção.

Por reprodução podemos entender como a produção de descendentes de uma espécie. Pode-se ter uma reprodução sexuada ou assexuada. Na reprodução sexuada o processo acontece com participação de dois seres diferentes da mesma espécie chamados de macho e fêmea, os quais diferem pela carga genética que contém os cromossomos sexuais, de forma a produzirem células chamadas gametas. Um novo ser é produzido com a informação genética dos pais. Na reprodução assexuada não há diferença genética entre seres da mesma espécie, no que refere ao seu papel na reprodução (AZEVEDO, 2000).

Mutação é a denominação dada a vários mecanismos de alteração genética, os quais, têm em comum, o fato de fazerem o novo cromossomo apresentar pouca informação dos pais. A mutação é um processo genético que procura garantir a diversidade das espécies. Este processo constitui-se de perturbações na cadeia dos cromossomos dando origem a uma nova cadeia, que guardará pouca informação da cadeia mãe.

Sobre Competição entende-se que em um ecossistema, espécies vivas lutam por recursos essenciais para sua sobrevivência e por serem estes por sua vez limitados, os vencedores ganham o direito a sobrevivência e perpetuam sua espécie que é a seleção natural. A seleção possibilita que a cada geração haverá uma população mais apta no ambiente.

Reprodução é uma propriedade óbvia de toda vida. Mas mutação é garantida pelo sistema natural, no qual, ela própria se reproduz continuamente, em um universo positivamente entrópico. Competição e seleção são as consequências inevitáveis de alguma população expandida restringida para uma área finita. Portanto evolução é o resultado desses processos estatísticos, interagindo, fundamentalmente nas populações, geração após geração.

2. BUSCA E OTIMIZAÇÃO

GAs podem ser usados em boa parte de problemas considerados como problemas de busca e otimização. GAs simples normalmente trabalham com descrições de entrada formadas por cadeias de bits de tamanho fixo. Outros tipos de GAs podem trabalhar com cadeias de bits de tamanho variável, como por exemplo GAs usados para Programação Genética. GAs possuem um paralelismo implícito decorrente da avaliação independente de cada uma dessas cadeias de bits, ou seja, pode-se avaliar a viabilidade de um conjunto de parâmetros para a solução do problema de otimização em questão.

O GA é indicado para a solução de problemas de otimização complexos, NP-Completos, como o "caixeiro viajante", que envolvem um grande número de variáveis e, conseqüentemente, espaços de soluções de dimensões elevadas. Além disso, em muitos casos onde outras estratégias de otimização falham na busca de uma solução, os GAs convergem. Os GAs são numericamente robustos, ou seja, não são sensíveis a erros de arredondamento no que se refere aos seus resultados finais.

Existem diversos métodos de busca e funções de avaliação. Na prática, eles são amplamente utilizados, com sucesso, em inúmeras aplicações (WINSTOM, 1992).

Os GAs diferem dos métodos tradicionais de busca e otimização, principalmente em quatro aspectos:

1. Trabalham com uma codificação do conjunto de parâmetros e não com os próprios parâmetros;

2. Trabalham com uma população e não com um único indivíduo;

3. Utilizam informações de custo e não outro conhecimento auxiliar, ou seja, a exigência é o conhecimento da fitness de cada indivíduo e não há necessidade de outra informação ou heurística dependente do problema;

4. Utilizam regras de transição probabilísticas e não determinísticas.

GAs têm se mostrado muito eficientes para busca de soluções ótimas, ou aproximadamente ótimas em uma grande variedade de problemas, pois não impõe muitas das limitações encontradas nos métodos de busca tradicionais. Além de seguir uma estratégia de gerar e testar soluções muito elegantes, por serem baseadas na solução biológica, são capazes de identificar e explorar aspectos do ambiente onde o problema está inserido e convergir globalmente para soluções ótimas, ou aproximadamente ótimos (HOLLAND, 1992).

2.1. CONCEITO GERAL

GA constitui uma técnica de busca, inspirada no processo de evolução dos seres vivos, baseada na seleção natural de Darwin. Os algoritmos evolucionários são uma família de modelos computacionais que incorporam uma solução potencial para um problema específico numa estrutura semelhante à de um cromossomo e aplicam operadores de seleção e "cross-over" a essas estruturas de forma a preservar informações críticas relativas à solução do problema. Normalmente os GAs são vistos como otimizadores de funções, embora a quantidade de problemas para o qual os GAs se aplicam seja bastante abrangente.

Considerando os sistemas biológicos, observa-se que os mesmos desenvolveram, ao longo da sua evolução, estratégias de adaptação de comportamento que possibilitam a sua sobrevivência e a perpetuação de suas espécies. As pressões do ambiente fizeram com que estas estratégias tivessem um forte impacto sobre os organismos biológicos, gerando profundas mudanças nos mesmos. Manifestações destas mudanças podem ser observadas nas especializações estruturais e funcionais, na organização da informação e nas representações internas do conhecimento (LINDEN, 2008).

Baseado nesta analogia com o processo de evolução biológica das espécies, chamada de metáfora biológica, os GAs mantém a informação sobre o ambiente no qual o problema está inserido, acumulando-a por todo o período de adaptação. Eles utilizam tal informação acumulada para limitar o espaço de busca e gerar novas soluções plausíveis dentro do domínio.

GA é um procedimento iterativo que mantém uma população de estruturas (chamadas indivíduos ou “string”), que representam possíveis soluções de um determinado problema. A cada incremento temporal (chamado geração), os indivíduos ou população atual são avaliados de acordo com o valor de sua aptidão para solução do problema. Tendo como base essa avaliação, uma nova população de soluções candidatas é formada, utilizando-se Operadores Genéticos (OG) específicos, tais como, “Cross-over” e mutação.

Os GAs podem ser enquadrados em grande parte dos problemas científicos, a serem formulados como problemas de busca e otimização Problemas desse tipo podem ser solucionados

por meios de métodos probabilísticos, numéricos ou enumerativos, ou por hibridismo destes (AZEVEDO, 2000).

Os métodos numéricos podem ser analíticos cuja função é explicitamente conhecida e derivável ou pode ser aproximada por alguma função derivável até o grau desejado de precisão. Os métodos enumerativos examinam cada ponto do espaço de busca, em busca de pontos ótimos. Por outro lado os métodos probabilísticos empregam a ideia de busca probabilística, isto é, descrevem a variação do sistema, que se realizam, essencialmente sob condições inalteradas. Esses sistemas são chamados de sistemas aleatórios, de forma que a teoria da probabilidade permite modelar seu comportamento (TANOMARU, 1995).

Apesar de aleatórios, os GAs não são buscas aleatórias não-direcionadas, pois exploram informações históricas para encontrar novos pontos de busca onde são esperados melhores desempenhos. Isso é feito por meio de processos iterativos, nos quais cada iteração é chamada geração. Durante cada geração, os princípios de seleção e reprodução são aplicados a uma população de candidatos que podem tornar-se disponíveis.

A seleção é a "escolha de indivíduos que participarão da formação de uma nova geração". São usados para seleção nos GA métodos de sorteio apresentados mais adiante. Por meio da seleção, são determinados quais indivíduos conseguiram se reproduzir, gerando um número determinado de descendentes para a próxima geração, com uma probabilidade determinada pelo seu índice de aptidão. Em outras palavras, os indivíduos com maior adaptação relativa têm maiores chances de se reproduzirem (AZEVEDO, 2000).

3 UTILIZANDO O GA

Em termos da biologia, um indivíduo é formado por um conjunto de cromossomos que determina seu genótipo. Cada cromossomo por sua vez é formado pelos genes que se localizam em cadeias e cada um tem determinado sua posição ("lócus"). O GA faz a analogia biológica criando uma representação do problema e o termo cromossomo é usado para ambos: Biologia e GA.

QUADRO 2 - Termos básicos do GA

Termo	descrição
Cromossomo	Cadeia de bits que representa uma solução possível para o problema.
Gene	Representação de cada parâmetro de acordo com o alfabeto utilizado
Fenótipo	Cromossomo codificado
População	Conjunto de pontos (indivíduos) no Espaço de Busca
Geração	Iteração completa do AG que gera uma nova população
aptidão bruta	Saída gerada pela função de avaliação para um indivíduo da população.
aptidão máxima	Melhor indivíduo da população corrente.

Existem diversos tipos de representação possíveis para os cromossomos. Entre outras podemos citar a binária, inteira e listas. A essa representação se dá o nome de alfabeto do GA. Assim o alfabeto, "alelo", é a representação que cada gene pode assumir. O conjunto de cromossomo, gênese alelos conferem características que são denominadas de fenótipo. De acordo com a classe de problema que se deseje resolver pode-se usar qualquer tipo de representação desde que atenda ao problema em questão.

Uma implementação de um algoritmo genético começa com um conjunto aleatório de cromossomos ("população"). Essas estruturas do conjunto são, então, avaliadas e associadas a uma probabilidade de reprodução calculada através da função de avaliação, de tal forma que as maiores

probabilidades são associadas aos cromossomos que representam uma melhor solução para o problema.

A função de avaliação de um problema de otimização é construída a partir dos parâmetros envolvidos no problema. Ela fornece uma medida da proximidade da solução em relação a um conjunto de parâmetros. Os parâmetros podem ser inversamente proporcionais, ou seja, quando um aumenta o outro diminui. O objetivo é encontrar o ponto ótimo. A função de avaliação permite o cálculo da aptidão bruta de cada indivíduo, que fornecerá o valor a ser usado para o cálculo de sua probabilidade de ser selecionado para reprodução (REZENDE, 2003). A tabela 5 apresenta o resumo dos principais conceitos.

Deve ser observado que cada cromossomo, chamado de indivíduo no GA, corresponde a um ponto no espaço de soluções do problema de otimização. O processo de solução adotado nos algoritmos genético consiste em gerar, através de regras específicas, um grande número de indivíduos (população), de forma a promover uma varredura tão extensa quanto necessária do espaço de soluções. A estrutura básica do algoritmo genético (Figura 04).

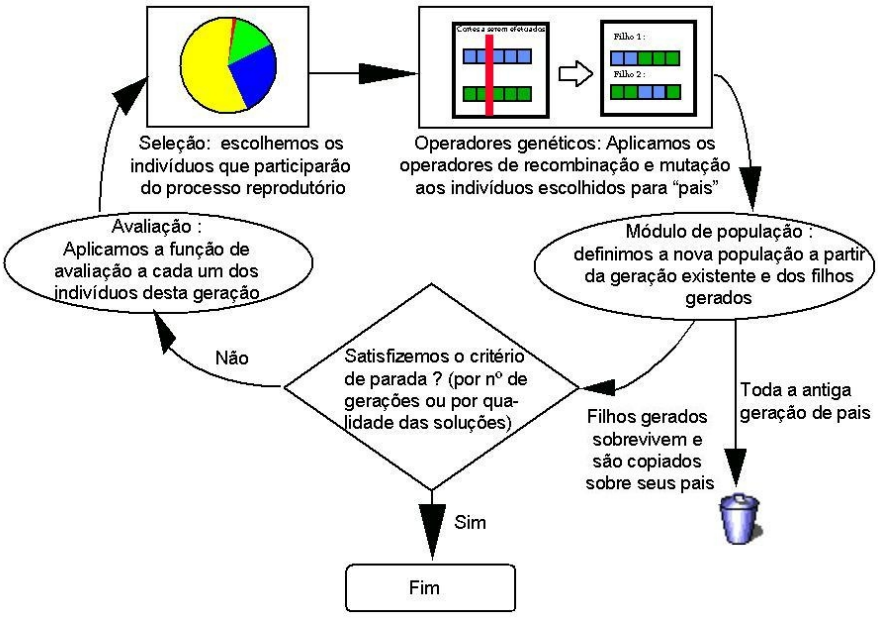


Figura 4 - Estrutura básica de um GA Simples (LINDEN, 2008)

Com referência ao diagrama da figura, observa-se que cada iteração do algoritmo genético corresponde à aplicação de um conjunto de quatro operações básicas: cálculo de aptidão, seleção, cruzamento e mutação. Ao fim destas operações cria-se uma nova população, chamada de geração que, espera-se, representa uma melhor aproximação da solução do problema de otimização que a população anterior. Vejamos a descrição das iterações do GA e as principais estruturas.

4 MODELANDO O GA

O agrupamento de proteínas pode trazer possibilidades de encontrar conhecimentos relevantes para a bio informática. Já existem pesquisas neste sentido.

Este projeto de pesquisa pretende utilizar GA para estabelecer regras para a clivagem de sequências de proteína para buscar agrupá-las por possuírem características, ora desconhecidas, mas que podem gerar conhecimento relevante no entendimento dos problemas biológicos.

Com base no GA, a definição do problema se dá em gerar um indivíduo que será chamado de regra. Uma regra determina uma condição de formação estrutural da sequência protéica. Esta regra será avaliada para um conjunto de sequências de proteínas obtendo um valor de avaliação que será a aptidão desse indivíduo. O valor de aptidão informa quantas sequências foram clivadas com determinada regra. A função matemática de avaliação escolhida é um somatório bem simples. É o somatório de quantas vezes a regra foi válida para um banco de sequências de proteínas. Na verdade, isto é feito para uma população, que é um conjunto de indivíduos que serão sorteados para gerar uma população intermediária. Esta população intermediária será submetida a um operador de cruzamento (cross-over) gerando uma nova população.

A cada geração o processo se repete até que cada vez que uma nova geração é gerada melhores regras sejam encontradas. No entanto, o processo natural muitas vezes pode gerar uma convergência genética. Uma convergência genética produz indivíduos semelhantes, ou seja, as avaliações dos indivíduos gerados são praticamente iguais, o que não é bom para o agrupamento pretendido.

Então é necessário estabelecer um operador que modifique as soluções convergentes e garanta a diversidade das soluções. Este operador é a mutação. Está então definido o nosso problema. Precisamos agora codificar o problema, escolher uma boa função de avaliação e criar operadores eficientes. O próximo item é destinado a este assunto.

Dispondo de sequências de proteínas que são formadas de 20 aminoácidos, então o GA criará regras que especificam o(os) gene(s) que devem ocupar uma posição na sequência. Uma boa regra agrupará um grande número de sequências. Não se sabe a priori de que forma se quer agrupar as sequências ou por qual característica irá agrupá-las. Neste sentido o GA é uma boa estratégia de algoritmo para ser aplicado já que sabe-se que, qualquer regra pode ser criada e o conhecimento sobre o agrupamento pode ainda não ter sido percebido. A cada nova geração de regras, outras regras diferentes são geradas, mas sempre com características da geração anterior, na tentativa de encontrar uma boa regra de clivagem.

As sequências são representadas por letras de um alfabeto. O alfabeto possui 20 letras que são os símbolos dos aminoácidos presentes nas proteínas dos seres vivos. Uma regra estabelece um padrão de aminoácidos, classificando as sequências de um arquivo (clivagem) e agrupando-as. Será considerado o parâmetro tamanho da nossa sequência para um valor 6. Então a posição do aminoácido pode ser de 1 até 6.

Uma regra pode ser definida como uma combinação lógica de condições, onde posição é igual ou diferente de um determinado aminoácido. A regra é a codificação do indivíduo, dado pela seguinte sintaxe:

Uma regra é um aminoácido atribuído a uma posição ou uma sentença lógica envolvendo regras.

A BNF da regra está representada a seguir:

<REGRA>::=<posição><op_aritmetico><aminoácido>|
<regra><operador_lógico><regra>

<POSIÇÃO>::=1 | 2 | 3 | 4 | 5 | 6

<OP_ARITIMÉTICO>::=< IGUALDADE> |< DESIGUALDADE>

<AMINOACIDO>::= A | B | C | D | E | F | G | H | K | L | M | N | P | Q | R | S | T | V | Y | W

<IGUALDADE>::= "=="

<DESIGUALDADE>::= "!="

<operador_lógico>::=AND|OR

O GA possui as fases de inicialização, cálculo da aptidão, seleção, cruzamento e mutação como foi descrito no capítulo II. A inicialização aqui tem a função de gerar uma população de regras suficientemente grande para varrer o espaço de soluções para as sequências de proteínas. A geração de uma regra é feita por uma função randômica que gera um operador lógico e a partir destes os operandos que são letras para os aminoácidos. O algoritmo de geração de uma população inicial será:

QUADRO 3: inicialização

```
1        definir o tamanho da população    (C_TAM_POPULACAO)
2        Enquanto i<C_TAM_POPULACAO
                gere um indivíduo
3       fim do enquanto
```

Cada regra é gerada como uma árvore binária, estrutura escolhida principalmente devido à natureza recursiva dessa estrutura e por ser bastante rápida a leitura e recuperação de dados numa árvore desse tipo. E sua adequação como representação de uma estrutura conceitual explicita na regra de formação de uma condição colocada no formato BNF acima.

Uma árvore binária é uma estrutura que possui um nó chamado raiz e esta possui sempre dois filhos: o nó da direita e o da esquerda (figura 10). Cada nó por sua vez pode possuir até dois filhos. Nós que possuam filhos são chamados nós internos e nós sem filhos são as folhas ou nós terminais. A árvore binária pode ser varrida (ler os elementos da árvore) de três formas: ordem in fixa, pré-fixada e pós-fixada. Para algumas funções do GA a melhor escolha da forma de leitura da estrutura é de acordo com o objetivo definido. Por exemplo, para verificar se uma proteína é clivada por uma regra, as regras devem ser lidas na árvore na ordem in fixa, pois as folhas devem ser regras e os nós mais internos operadores lógicos.

Depois de gerada a população inicial, deve-se calcular a aptidão de cada indivíduo estabelecendo a nota (fitness) para ser utilizada na seleção dos indivíduos que deverão sofrer a ação do operador de cross-over. O algoritmo da função de avaliação é o seguinte:

QUADRO 4: avaliação da população

```
1 Ler  indivíduo
2 Enquanto não chegar ao fim do arquivo ler a sequência  e sua avaliação esperada
(valor) do arquivo
3       avaliar se a clivagem é satisfeita "chamar função avalia"
4      Comparar o retorno da função avalia com o valor esperado
                Verdade soma um
                Falso fim se
5 Fim do enquanto
6 retorna a fitness que é o total da soma
```

A função de avaliação faz a chamada a uma função (avalia) que faz o teste para saber se a sequência é clivada ou não para aquela regra. Fazer uma chamada a uma função significa que alguma rotina será executada externamente ao módulo que está sendo executado. Neste caso a função avalia faz a avaliação da regra para uma proteína a cada proteína que é lida no arquivo. O retorno da função é uma sinalização de falso ou verdadeiro.

A função recebe como parâmetros uma estrutura árvore (regra) e uma string (sequência).

Se o retorno é 0 significa que a proteína é clivada. Ou se for 1 ela não é clivada. Esta chamada à função "avalia", será feita para *n* sequências e a cada iteração será incrementado, ou não, o total da regra até que chegue ao final do arquivo. A avaliação do indivíduo (regra) é o somatório final.

Então, estabelecido o valor da avaliação, aplica-se a seleção para escolha da população intermediária, usando-se o método da roleta viciada, no qual cada indivíduo tem uma percentagem de chance de ser escolhido proporcional à sua qualidade.

Constituída a população intermediária está na hora de utilizar os operadores genéticos. O operador de cruzamento (cross-over) possui um valor de probabilidade próximo de 1 nesta fase inicial, enquanto o operador de mutação possui uma avaliação quase nula. Isto quer dizer que, até que se tenha várias gerações, o operador de mutação será pouco utilizado. Nas próximas populações geradas, quando for calculada a avaliação das regras, os valores são verificados. Pode acontecer que, em dada geração, indivíduos tenham valores de aptidão muito próximos, ou seja, o somatório das regras é muito parecido. Isso caracteriza uma situação de convergência. Se, em alguma geração futura, houver uma convergência muito acentuada, então neste momento o cross-over reduz sua probabilidade e a mutação tem a sua aumentada. Em seguida está descrito como será o algoritmo do cross-over:

Quadro 5: escolha do cross over

1. *Ler as regras escolhidas;*
2. *Criar uma estrutura auxiliar (aux:tipo árvore);*
3. *Para i de 0 a C_TAM_POPULACAO/2*
4. *Calcular o número de nós da estrutura 1(árvore);*
5. *Sortear um ponto de corte entre 1 e o número obtido;*
6. *Cortar o galho que possui o nó sorteado;*
7. *Aux recebe o galho cortado;*
8. *Calcular o número de nós da estrutura 2(árvore);*
9. *Sortear um ponto de corte entre 1 e o número obtido;*
10. *Cortar o galho que possui o nó sorteado;*
11. *O nó –1 da estrutura 1 recebe o galho da estrutura 2;*
12. *O nó cortado –1 da estrutura 2 recebe aux;*
13. *Fim enquanto*
14. *Libera menória de aux;*

O operador de cross-over tem como objetivo realizar uma troca de informação entre dois indivíduos da população de uma maneira análoga à reprodução sexuada.

Seu uso implica no intercâmbio entre dois indivíduos, de pedaços de antecedentes de regras (“material genético”).

São gerados dois “filhos” que possuem fragmentos de regras de cada um dos pais, compartilhando suas qualidades na modelagem dos dados. Como pode ser observado o cross-over utilizado aqui é o cross-over de um ponto. O ponto de corte é sorteado até um valor igual ao valor do número de nodos da árvore.

Escolhe-se então um nó aleatoriamente em cada uma das árvores e realiza-se o intercâmbio entre as sub-árvores enraizadas em cada um destes nós (Figura 5). O operador de cross-over troca duas sub-árvores entre dois cromossomos pais formando dois filhos.

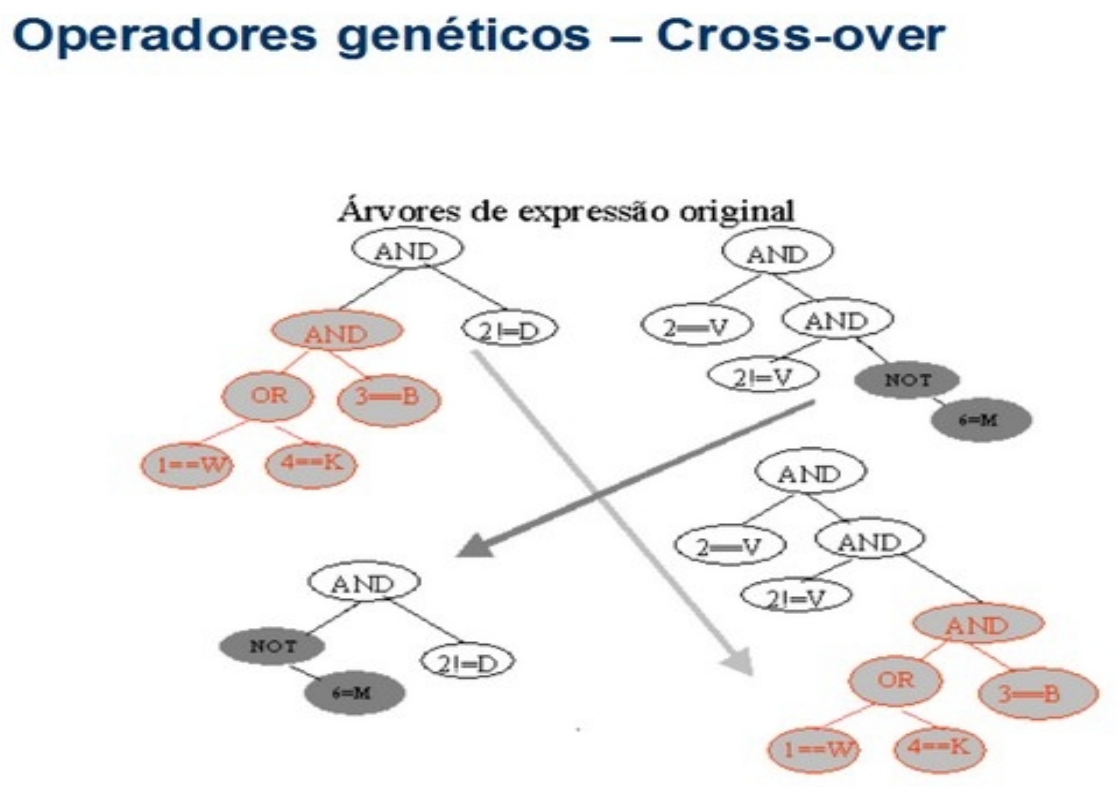


Figura 5 - Cross-over aplicado a duas Regras

O operador de mutação tem como função inserir variabilidade genética na população que está sendo evoluída.

Sem a aplicação deste operador, a população tende a convergir em algumas gerações para um grupo muito pequeno de cromossomos.

O tempo de convergência é função apenas do tamanho do espaço de busca e do tamanho da população.

Quadro 6: mutação

1 ler a regra escolhidaa;
2 criar uma estrutura auxiliar (aux:tipo árvore);
3 Calcular o número de nós da estrutura (árvore);

4 sortear um ponto de corte entre 1 e o número de nós obtido;
5 Aux recebe uma nova estrutura gerada pela função gera_regra
6 cortar o galho que possui o nodo sorteado;
7 o nodo pai do nodo sorteado para corte recebe AUX;

O operador de Mutação: escolhe um nó aleatoriamente na árvore de regra e substitui aquela sub-árvore por outra gerada pelo mesmo gerador que criou a população inicial (Figura 6). Desta forma os operadores são aplicados ao espaço de busca que são as diversas regras geradas dentro das combinações de posições e dos 20 aminoácidos existentes nas células dos organismos vivos.

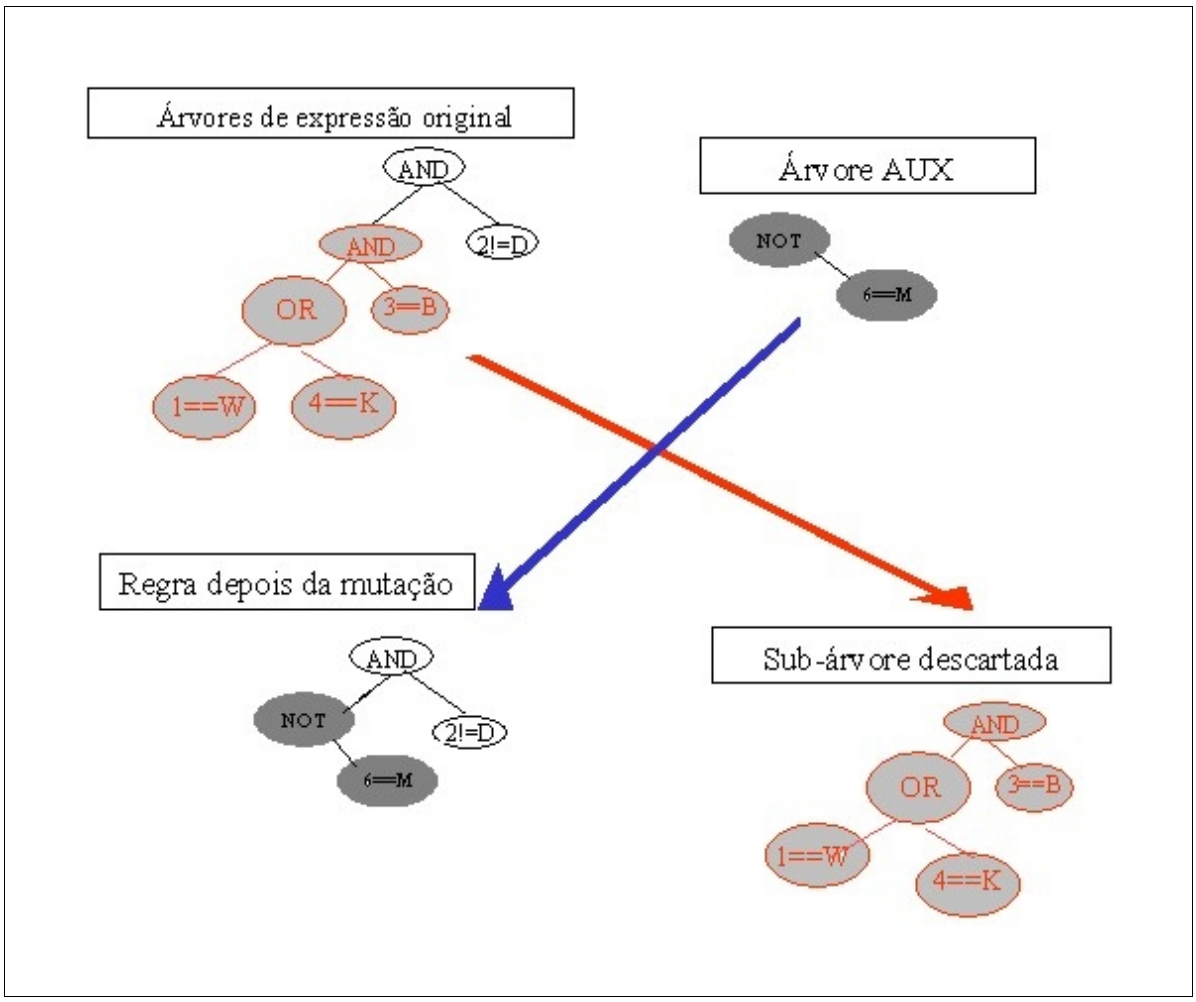


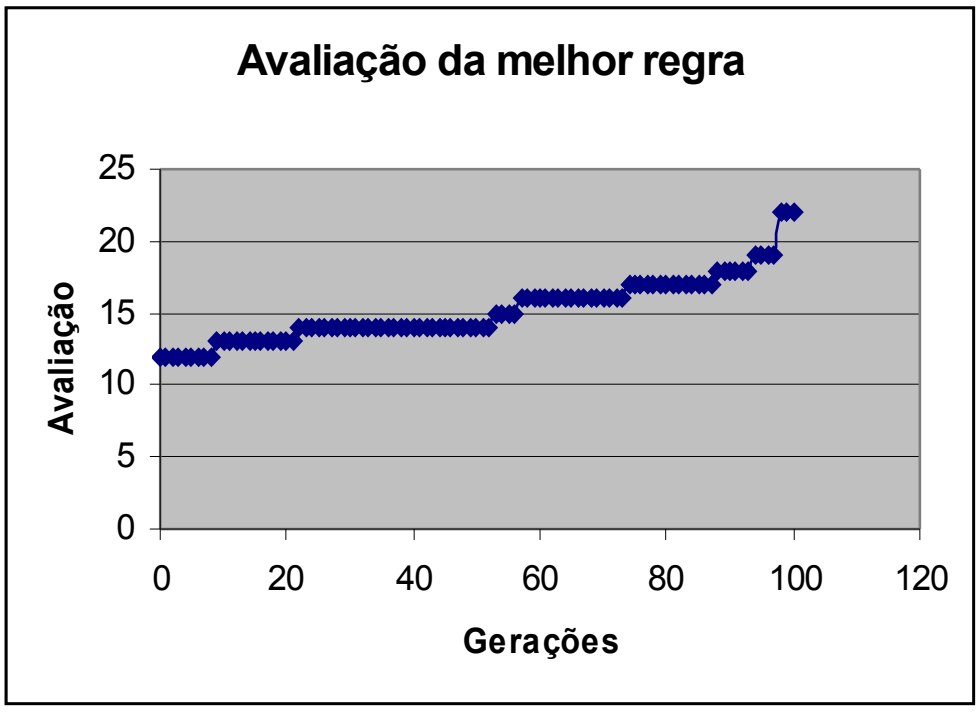
Figura 6 - Atuação do Operador de Mutação na aplicação de Regras

EXECUTANDO UM GA NO BIOINFO.EXE

Para o teste do modelo desenvolvido este trabalho utilizou dados fictícios em arquivo texto onde cada linha do arquivo representava uma sequência de caracteres e um valor inteiro. Se 1, a sequência lida não está validada pelo conjunto de regras geradas pelo GA. Se 0 (zero) a validação ocorreu.

Foi utilizado o arquivo Prot1.txt que continha 38 sequências e foram validadas 22 na operação do GA. O desempenho do GA não é o desempenho esperado. A utilização de outra função de avaliação pode ser testada para encontrar uma melhora nas gerações. Os parâmetros para o tratamento do GA são o tamanha da população e a taxa de elitismo no módulo da população inicial , o percentual de utilização de cross-over ea taxa de utilização de mutação. A condição de parada para o GA pode se dá se houver um valor máximo de validação, uma melhora de 50% da população (desempenho global) ou o número de gerações. A adoção de um parâmetro de 50% diz respeito ao fato de que uma geração precisa ser melhor que a anterior para que seja possível encontrar um ponto de clivagem. O desempenho do GA foi mostrado no gráfico em função das gerações de regras criadas pelo GA (Gráfico 01).

GRÁFICO 01 – Desempenho do GA para o arquivo prot1.txt



A análise revela que na execução do GA para uma população de 200 indivíduos, durante 100 gerações houve uma evolução mais lenta nas gerações iniciais; o grau de validação ainda pode ser melhorado com a pesquisa de novas funções de avaliação, porque validou um bom número das seqüências analisadas, mas não atingiu o máximo ideal. O GA busca a melhor regra comparando os valores de fitness, sendo que se houver mais de uma com o mesmo valor máximo, a escolhida é a primeira ocorrência.

RESULTADOS E CONCLUSÃO

Para o desenvolvimento do GA para este experimento foi desenvolvido um programa computacional denominado Bio.c. O programa foi desenvolvido na linguagem de programação C por esta se mostrar robusta e de grande poder de computação.

A estrutura de dados escolhida após análise do problema foi a árvore binária, dada sua natureza recursiva de co-geração. Assim as regras para localização de um aminoácido em uma sequência serão criadas por uma função recursiva de árvore binária. A origem da informação será um arquivo externo com sequências de letras que representam os aminoácidos seguidas de um valor inteiro. Para a recuperação destes dados foi utilizado um vetor de seis posições de caracteres que se constitui de uma estrutura bem simples de manipulação, e uma variável inteira.

O programa produziu um resultado muito homogêneo, o que poderá ser melhorado mediante a utilização de outra função de avaliação para o desempenho das novas gerações. Este estudo sobre a utilização de técnicas computacionais como ferramentas para a área de bioinformática, traz algumas reflexões importantes:

- A necessidade do conhecimento integrado entre a biologia e a informática: para um biólogo tratar uma grande massa de dados sobre sequências de proteínas, sem uma ferramenta computacional, levaria muito tempo e poderia não resultar eficientemente uma pesquisa. Entretanto, com o uso de GA como ferramenta computacional esses dados são tratados de forma mais rápida e mais eficiente.

- A importância de se utilizar técnicas de heurísticas em problemas de difícil solução e espaço de buscas intratáveis, formaliza a utilização de GA em problema de forma bastante interessante: o estudo das cadeias protéicas, o que faz concluir sua relevância para a produção acadêmica, alcançando o objetivo pretendido, que é verificar a aplicabilidade da informática em campos de conhecimentos diversos.

A análise dos resultados obtidos nas execuções do programa desenvolvido para este trabalho, traz algumas considerações importantes, já comentadas nas literaturas sobre o funcionamento de um algoritmo genético.

O primeiro é que, por ser uma heurística, nem sempre um GA evolui para um resultado ótimo. Se o problema tivesse uma complexidade menor ou um espaço de busca mais restrito, não seria interessante usar um procedimento tal como o GA, mas sim um algoritmo tradicional de busca. Nos exemplos usados para este programa, o GA mostrou uma evolução progressiva, tendendo a um resultado positivo, encontrando várias regras que obtiveram a mesma avaliação. As regras de última geração atingiram mais de 50% de clivagem da proteína estudada.

Finalmente, concluímos que GA´a são aplicáveis a problemas estudados pela biologia, e servem como ferramenta para buscar soluções que envolvem um volume substancial de dados.

Como trabalho futuro aponta-se para a necessidade de se estudar novas formas avançadas de GA, testando novos operadores e verificando se os resultados apresentarão melhorias significativas. Os GA´s poderiam ter operadores de mais de um ponto para aumentar a diversidade e eficiência dos resultados. A base científica adquirida nesta experiência acadêmica nos leva a concluir que este tema pode representar teses mais aprofundadas com estudos mais profundos com estudos mais dedicados, para alcançar melhores resultados para GA em solução de biologia.

REFERÊNCIAS

ALGARVE, A S. **Simulação do sistema circulatório com controle Neural** Central. Exame de qualificação para Doutorado UFSC, Florianópolis, Brasil 1997

AZEVEDO, F M. **Redes neurais" com aplicações em controle e em Sistemas especialistas** Editora Visual Books- Florianópolis – 2000

CRUCK, K. **"Applying Genetics to Fuzzy Logic"** – AI Expert, march, 1991

FERNANDES, A M R. **"Inteligência Artificial – Noções Gerais".** Editora Visual Books-Florianópolis – 2003

FOGEL, D B. **Evolutionary Computation: Tovard a New Philosophy of Machine Inteligence**. IEEE Press, New York, USA, 1995.

GREFENTETTE, J.J. **Foundations of Genetic Algorithms** *-2-,* D. Whitley, ed., Morgan Kaufmann. pp: 75-91, 1993,

_____ **Optiminization of Control parameters for genetic algorithms.** IEEE Transactions on Systems, Man, and Cybernetics, v. SMC-16, N1, P.122-128, 1986.

HOD, H J. Adaption in Natural and Artificial Systems. MIT Press, Cambridge, Massachusetts, 1975.

HORNER, H. **A C++ class library for genetic programming:** The Vienna University of Economics - genetic programming kernel", *Internet Draft,* maio de 1996.

LINDEN, R. **Algoritmos Genéticos**, 2a edição, Ed. Brasport, Rio de Janeiro, 2008

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

MUHLENBEIN, H. **How genetic algorithms really work:** I. Mutation and Hillclimbing, *Paralell Problem Solving from Nature -2-,* R. Manner and B. Manderick, North Holland, 1992.

NIX, A; VOSE, M. **Modeling Genetic Algorithms with Markov Chains**. *Annals of Mathematics and Artificial Intelligence.* 5:79-88. 1998.

OLIVEIRA, H A. **"Lógica Difusa" Aspectos práticos e aplicações**. Editora Interciência – Rio de Janeiro – 1999

POIAN, A T. **Bioquímica** vol. 01 Fundação CECIERJ – Rio de Janeiro 2004

REZENDE, S O. **Sistemas Inteligentes: Fundamentos e aplicações** -Barueri SP Manole 2003

SETÚBAL. J C (2003). **A origem e o sentido da Bioinformática**. *ComCiencia*. Campinas, SP.

TANOMARU, J J. **Motivação, Fundamentos e aplicações de Algoritmos genéticos.** In: Procesings do II Congresso Brasileiro de Rdes Neurais, v.1, p. 331-411. Curitiba, Brasil,1995.

WHITLEY, D. **A Genetic Algorithm Tutorial**, Computer Science Department, Colorado State University.

www.icb.ufmg.br

www1.folha.uol.com.br/folha/especial/2003/dna/

http://quimica.fe.usp.br/

http://www.virtual.unifesp.br

http://www.gta.ufrj.br

http://www.wu-wien.ac.at/usr/h88/h8850092