

Universidade Federal de Alagoas
Mestrado em Modelagem Computacional
de Conhecimento



Dissertação de Mestrado

**Alinhamento Múltiplo de Proteínas via
Algoritmo Genético Baseado em Tipos
Abstratos de Dados**

Danielle Furtado dos Santos
daniellefurtado@gmail.com

Orientador:
Roberta Vilhena Vieira Lopes

Maceió, Novembro de 2008

Danielle Furtado dos Santos

**Alinhamento Múltiplo de Proteínas via
Algoritmo Genético Baseado em Tipos
Abstratos de Dados**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Curso de Mestrado em Modelagem Computacional de Conhecimento do Instituto de Computação da Universidade Federal de Alagoas.

Orientador:

Roberta Vilhena Vieira Lopes

Maceió, Novembro de 2008

Catálogo na fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico
Bibliotecária Responsável: Maria Auxiliadora G. da Cunha

S231a Santos, Danielle Furtado dos.
Alinhamento múltiplo de proteínas via algoritmo genético baseado em tipos abstratos de dados / Danielle Furtado dos Santos. - Maceió, 2008
ix,119 f. : il.

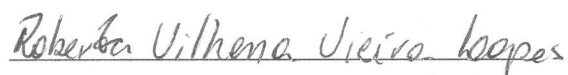
Orientador: Roberta Vilhena Vieira Lopes.
Dissertação (mestrado em Modelagem Computacional de Conhecimento) –
Universidade Federal de Alagoas. Instituto de Computação. Maceió, 2008.

Bibliografia: f. 116-119.

1. Algoritmos genético. 2. Tipos abstratos de dados. 3. População. 4. Cromossomos.
I. Título.

CDU: 004.421:577.2

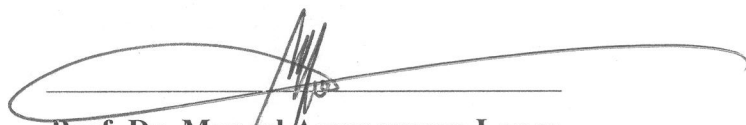
Dissertação apresentada como requisito parcial para a obtenção do grau de Mestre em Modelagem Computacional de Conhecimento pelo Programa Multidisciplinar de Pós-Graduação em Modelagem Computacional de Conhecimento, da Universidade Federal de Alagoas, aprovada pela comissão examinadora que abaixo assina:



Profa. Dra. Roberta Vilhena Vieira Lopes

UFAL – Instituto de Computação

Orientadora



Prof. Dr. Manoel Agamemnon Lopes

UFAL – Centro de Ciências Agrárias

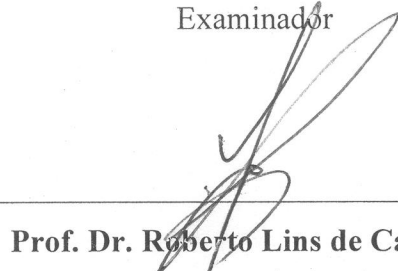
Examinador



Prof. Dr. Cícero Eduardo Ramalho Neto

UFAL – Centro de Ciências Agrárias

Examinador



Prof. Dr. Roberto Lins de Carvalho

LNCC – Laboratório Nacional de Computação Científica

Examinador

Maceió, novembro de 2008.

Resumo

Este trabalho apresenta um novo modelo capaz de realizar o alinhamento múltiplo de proteínas utilizando algoritmo genético baseado em tipos abstratos de dados, denominado GAADT, no qual o cromossomo se dispõe em genes que por sua vez é composto de unidades elementares denominadas bases. Cada cromossomo representa um possível alinhamento entre as seqüências de proteínas e a adaptação do cromossomo é calculada conforme as bases se dispõem no alinhamento. O modelo se difere de outros métodos de alinhamento múltiplo por alinhar as seqüências como um todo, avaliando as colunas (genes) que compõem o alinhamento (cromossomo), ao invés de alinhar seqüências duas a duas progressivamente ou hierarquicamente. As potenciais características do modelo dizem respeito a estrutura de dados organizada, operações genéticas bem definidas sobre os tipos modelados e a convergência para uma solução próxima às encontradas por outras ferramentas, apesar desse algoritmo usar uma quantidade menor de conhecimento frente aos algoritmos existentes. A justificativa de que o modelo é válido foi realizada analisando sua performance com alinhamentos referência, utilizando como estudo de caso um subgrupo de famílias de proteínas.

Abstract

This work presents a new model to the multiple protein alignment problem using genetic algorithm based on abstract data types - GAADT. This model uses a structure called chromosome, which is a set of gene that in turn is composed of basic units called bases. Each chromosome is a possible alignment of the input sequences and the chromosome fitness is calculated according with the bases order in alignment. This model is different from other paradigms of multiple alignment by aligning the input sequences as a whole instead of a progressive pairwise alignment approach. Some characteristics of this model concern to the data structure, well-defined genetic operations and the convergence to a solution close to that found in other tools. The validation was performed comparing reference alignments of protein families subset with the results of the model.

Agradecimentos

A Deus por ter me proporcionado capacidade para cursar uma pós-graduação e forças nos momentos de dificuldade.

Aos meus pais, Jaciara Furtado e Daniel dos Santos, por todo o amor a mim dedicado, por serem eternas fontes de inspiração em minha vida e por serem fortes nos momentos em que muito precisaram e eu não estava presente.

Gostaria de fazer uma dedicação especial a duas pessoas, sem as quais não seria possível a conclusão deste trabalho:

- Ao meu marido, Ulisses Dias, por apoiar minhas decisões com muito amor e paciência, compreendendo o tempo que eu precisei me dedicar a este trabalho. Por acreditar em mim quando eu achava que estava difícil finalizar. Por sua capacidade de me manter calma. Pelas constantes sugestões e críticas que foram dadas ao trabalho. Agradeço pelo mestrado que nos uniu cada vez mais, por todos os momentos de alegria que compartilhamos e por todas as dificuldades que aprendemos a superar. Por tudo que não está escrito nestas linhas, agradeço simplesmente por me amar.
- À professora e orientadora Roberta Lopes, por acreditar que poderia ser realizado um trabalho e assumindo a real responsabilidade de orientação com o tempo já decorrido de mestrado, o que acarretou muitas cobranças todas as vezes que foi necessário pedir mais prazo. Tenho muito a agradecer pela paciência que teve com este trabalho, pelas várias vezes que corrigiu meu texto na expectativa de melhoras e por todas as vezes que me deu ânimo para prosseguir. Enfim, por toda orientação acadêmica e de vida que me doou em longas conversas durante os últimos anos.

Com o tempo do desenvolvimento desta dissertação muitas pessoas incentivaram este trabalho, dentre as quais gostaria de agradecer e enfatizar alguns nomes:

- Ao meu irmão, Rafael Furtado, por me tranquilizar nos momentos de dificuldade familiar para que eu pudesse continuar o desenvolvimento desta dissertação. Por todo incentivo desde o primeiro instante, quando precisei sair de casa para cursar o mestrado, até as chamadas de atenção para eu estudar e terminar a escrita.
- Aos meus sogros, Regina Dias e Otacílio Dias, por toda força e palavras de ânimo.
- Ao Ig Ibert, pois valeu a pena ter cursado o mestrado tão longe de casa simplesmente por ter sua amizade.
- As minhas amigas Elba Quirino e Rosemeire Lima que, assim como eu, casaram-se durante o mestrado e mostraram que é possível conciliar a vida acadêmica com a vida pessoal. Agradeço por serem exemplo de força e determinação.

- Ao professor Agamemnon Lopes por toda acolhida cedida e por me guiar direta ou indiretamente para a vida longe da família.
- À Taciana Almeida, pela sua amizade e por me animar com a sua euforia nos momentos em que eu me sentia enfraquecida.
- Ao André Almeida, pela constante preocupação acerca do trabalho, pelas várias conversas e troca de informações que muito colaboraram para elaboração da fase final desta dissertação.
- Aos amigos Aldilene Maia, Elaine Hayashi, Cristian Viana, Glauber Cabral, Javier Montoya, Juliana de Santi, Marcelo Dias, Michelle Pereira, Neumar Malheiros, Priscila Saboia, Raoni Fassina, Ricardo Freitas e outros amigos cujos nomes podem não estar aqui por conta da minha pequena falha de memória. Agradeço por toda força que cada um me passou em diversas fases do desenvolvimento desta dissertação. Quem tem amigos possui uma corrente forte de energia positiva.

Além dos amigos, gostaria de agradecer aos professores do Mestrado em Modelagem Computacional por contribuírem na minha formação acadêmica.

A todos os membros da banca por contribuírem com a finalização deste trabalho.
À FAPEAL pelo apoio financeiro.

Sumário

| | | |
|----------|---------------------------------------------------------------|-----------|
| 1 | Introdução | 1 |
| 1.1 | Alinhamento Múltiplo de Proteínas | 1 |
| 1.2 | Motivação | 3 |
| 1.3 | Objetivos e Contribuições | 4 |
| 1.4 | Organização da Dissertação | 4 |
| 2 | Introdução à Biologia Molecular | 6 |
| 2.1 | A estrutura do DNA | 6 |
| 2.2 | A estrutura do RNA | 11 |
| 2.2.1 | Classes de RNA | 11 |
| 2.2.1.1 | RNA mensageiro ou mRNA | 11 |
| 2.2.1.2 | RNA ribossomal ou rRNA | 12 |
| 2.2.1.3 | RNA transportador ou tRNA | 14 |
| 2.3 | Código genético | 16 |
| 2.4 | Expressão gênica | 18 |
| 2.4.1 | Transcrição | 18 |
| 2.4.1.1 | O Processo de transcrição | 19 |
| 2.4.2 | Tradução | 20 |
| 2.4.2.1 | O Processo de tradução | 20 |
| 2.4.3 | Proteína | 23 |
| 2.5 | Homologia | 27 |
| 3 | Introdução à Comparação de Sequências | 28 |
| 3.1 | Alinhamento de seqüências | 28 |
| 3.2 | Alinhamento por programação dinâmica | 34 |
| 3.3 | Matrizes de substituição | 38 |
| 3.3.1 | PAM | 38 |
| 3.3.1.1 | Matrix PAM1 | 40 |
| 3.3.2 | BLOSUM | 42 |
| 3.4 | Alinhamento múltiplo | 44 |
| 3.4.1 | Alinhamento progressivo | 46 |
| 3.4.1.1 | Heurística para alinhamento progressivo | 47 |
| 3.4.1.2 | CLUSTALW | 48 |
| 3.4.2 | Heurísticas alternativas ao alinhamento progressivo | 49 |
| 3.5 | SAGA | 49 |
| 3.5.1 | Operadores do SAGA | 52 |
| 3.5.1.1 | Cruzamento | 52 |
| 3.5.1.2 | Mutação | 53 |

| | | |
|-----------|-------------------------------------------------------------------|------------|
| 3.5.1.2.1 | Inserção de <i>gaps</i> | 53 |
| 3.5.1.2.2 | Movimentação de blocos | 54 |
| 3.5.1.2.3 | Pesquisa de blocos | 54 |
| 3.5.1.2.4 | Rearranjo ótimo ou sub-ótimo local | 55 |
| 4 | O Algoritmo Genético Baseado em Tipos Abstratos de Dados | 56 |
| 4.1 | Introdução | 56 |
| 4.2 | Avaliação da Aprendizagem com Mapas Conceituais | 57 |
| 4.3 | Tipos Básicos | 60 |
| 4.4 | Operadores Genéticos | 65 |
| 4.5 | Ambiente | 74 |
| 4.6 | Algoritmo | 75 |
| 5 | GAADT para o Problema de Alinhamento Múltiplo de Proteínas | 80 |
| 5.1 | Introdução | 80 |
| 5.2 | Descrição Formal do Problema: Alinhamento Múltiplo de Proteína | 81 |
| 5.3 | Uma Instanciação do GAADT para Alinhamento Múltiplo | 82 |
| 5.3.1 | Elementos Básicos | 82 |
| 5.3.2 | Operadores Genéticos | 86 |
| 5.3.2.1 | Cruzamento | 94 |
| 5.3.2.2 | Mutação | 96 |
| 5.4 | Ambiente | 98 |
| 5.5 | Algoritmo | 99 |
| 6 | Resultados | 101 |
| 6.1 | Medida de Qualidade do BALIBASE | 102 |
| 6.2 | Parâmetros do Algoritmo | 103 |
| 6.2.1 | Matriz de substituição | 103 |
| 6.2.2 | Parâmetros Relacionados com Algoritmos Genéticos | 108 |
| 6.2.2.1 | Tamanhos da população e do cromossomo | 108 |
| 6.2.2.2 | População inicial | 109 |
| 6.2.2.3 | Método de escolha dos cromossomos pais | 109 |
| 6.3 | Resultados | 111 |
| 7 | Conclusão | 113 |
| 7.1 | Contribuições e Relevância | 113 |
| 7.2 | Sugestões de Trabalhos Futuros | 114 |

Lista de Figuras

| | | |
|------|--------------------------------------------------------|----|
| 2.1 | Molécula de açúcar do DNA | 6 |
| 2.2 | Bases nitrogenadas | 7 |
| 2.3 | Ácido fosfórico | 7 |
| 2.4 | Composição do nucleotídeo | 8 |
| 2.5 | Ligação fosfodiéster | 9 |
| 2.6 | Pareamento de bases | 10 |
| 2.7 | Componentes do RNA que diferem do DNA | 11 |
| 2.8 | Ribossomo procariótico | 13 |
| 2.9 | Ribossomo eucariótico | 13 |
| 2.10 | Ribossomo e seus sítios | 14 |
| 2.11 | RNA transportador | 15 |
| 2.12 | Aminoácido ligado ao tRNA | 21 |
| 2.13 | Formação do complexo de pré-iniciação | 22 |
| 2.14 | Formação do complexo de iniciação | 22 |
| 2.15 | Composição do aminoácido | 24 |
| 2.16 | Ligação peptídica | 24 |
| 3.1 | Exemplo de sobreposição de duas seqüências | 30 |
| 3.2 | Exemplo de árvore filogenética | 47 |
| 4.1 | Exemplo de Mapa Fornecido Pelo Estudante | 58 |
| 4.2 | Alguns Genes Produzidos pelo <i>afg1</i> | 62 |
| 4.3 | Alguns Genes Produzidos pelo <i>afg2</i> | 62 |
| 4.4 | Exemplo de Cromossomos Fornecidos pelo GAADT | 64 |
| 4.5 | Resultado do Cruzamento entre c_1 e c_2 | 71 |
| 4.6 | Exemplo Inserção de Gene | 72 |
| 4.7 | Exemplo Supressão de Gene | 73 |
| 4.8 | Exemplo Troca de Genes | 78 |

Lista de Tabelas

| | | |
|------|--------------------------------------------------------------------|----|
| 2.1 | Nomenclatura dos nucleotídeos de DNA | 8 |
| 2.2 | Nomenclatura dos nucleotídeos de RNA | 11 |
| 2.3 | Os 20 aminoácidos que existem na natureza | 16 |
| 2.4 | Código genético | 17 |
| 2.5 | Código diferente do código universal | 18 |
| | | |
| 3.1 | Exemplo utilizando modelo de <i>gap</i> linear | 32 |
| 3.2 | Exemplo utilizando modelo de abertura de <i>gap</i> | 32 |
| 3.3 | Exemplo de matriz de mérito | 33 |
| 3.4 | Matriz de programação dinâmica | 36 |
| 3.5 | Exemplo de alinhamento com programação Dinâmica | 37 |
| 3.6 | Pontos de mutações aceitáveis | 39 |
| 3.7 | Matriz PAM1 | 41 |
| 3.8 | Exemplo de pontuação com PAM1 | 42 |
| 3.9 | Matriz Blosum62 | 43 |
| 3.10 | Exemplo de pontuação com BLOSUM62 | 44 |
| | | |
| 4.1 | Matriz de Adjacência | 59 |
| 4.2 | Valores das Relações Binárias da Ontologia | 60 |
| 4.3 | Exemplos de AFG para Mapas Conceituais | 62 |
| 4.4 | Exemplos de AFC para Mapas Conceituais | 63 |
| 4.5 | Mapa do Aluno - Correspondência Cromossômica | 64 |
| 4.6 | Mapa do Aluno - Grau dos Genes | 66 |
| 4.7 | Exemplo de Adaptação de Cromossomos em MCs | 68 |
| 4.8 | Exemplo de Fecundação de Cromossomos em MCs | 70 |
| 4.9 | Exemplo Cruzamento | 71 |
| 4.10 | Genes Utilizados no Algoritmo | 76 |
| 4.11 | Iterações do GAADT | 78 |
| 4.12 | População gerada pelo cruzamento | 79 |
| | | |
| 5.1 | Axiomas de formação de genes | 83 |
| 5.2 | Exemplo de Sequências e um Possível Alinhamento Múltiplo | 84 |
| 5.3 | Exemplo de Tipo Gene | 84 |
| 5.4 | Exemplo do Tipo Bloco Gênico | 85 |
| 5.5 | Exemplo de Tipo Cromossomo | 86 |
| 5.6 | Exemplo de Tipo População | 87 |
| 5.7 | Exemplo de Grau de Adaptação do Gene | 92 |
| 5.8 | Exemplo de Grau de Adaptação do Cromossomo | 92 |
| 5.9 | Exemplo Mesma Característica | 94 |

| | | |
|------|-------------------------------------------------------------------------|-----|
| 6.1 | Teste família 1aab com matriz de substituição BLOSUM62 | 105 |
| 6.2 | Teste família 1aab com matriz de substituição DAYHOFF | 105 |
| 6.3 | Teste família 1aab com matriz de substituição PAM40 | 106 |
| 6.4 | Teste família 1aab com matriz de substituição PAM80 | 106 |
| 6.5 | Teste família 1aab com matriz de substituição PAM120 | 107 |
| 6.6 | Teste família 1aab com matriz de substituição PAM250 | 107 |
| 6.7 | Teste Cruzamento: $M = F$ | 110 |
| 6.8 | Teste Cruzamento: $M = 50\%P$ e $F = 50\%P$ | 111 |
| 6.9 | Características das Famílias Protéicas | 111 |
| 6.10 | Resultado Final | 112 |

Capítulo 1

Introdução

1.1 Alinhamento Múltiplo de Proteínas

O estudo do genoma permitiu conhecer a constituição dos genes dos seres vivos. A forma pela qual os genes influenciam no funcionamento dos organismos está diretamente ligada a proteína por eles sintetizada. Ao se isolar as proteínas e investigar suas propriedades estruturais, de acordo com Branden & Tooze (1991) é possível descobrir mecanismos celulares relacionados ao desenvolvimento de doenças, princípios ativos para produção de medicamentos e o funcionamento de compostos químicos.

O alinhamento de seqüências, em especial o alinhamento de seqüências protéicas, é uma área primordial no estudo de biologia molecular. O alinhamento pode encontrar estruturas similares que conservaram suas características no decorrer da evolução da proteína.

A análise dessas estruturas similares permite identificar se há ou não uma relação evolutiva entre as proteínas. Dessa forma, proteínas semelhantes em seres-vivos diferentes possuem uma grande chance de manterem funcionalidade parecida. A explicação para que isso ocorra está no modo como o processo evolutivo da proteína aconteceu.

Uma estrutura protéica estável possui várias propriedades químicas e físicas importantes para o desempenho de sua função, propriedades estas que, caso danificadas por alguma mutação no curso evolutivo, ocasionam um organismo deficiente ou incapaz de se reproduzir. Dessa forma, as subseqüências semelhantes entre as proteínas são justamente aquelas mais importantes para caracterizar a sua função, pois são as que não poderiam ser modificadas por razões estruturais ou bioquímicas, o que faz da descoberta de similaridade entre proteínas uma área de pesquisa relevante.

Uma outra conseqüência do processo evolutivo é que determinar o alinha-

mento múltiplo ótimo de uma família inteira de proteínas permite identificar prováveis sítios ativos, que são as regiões na proteína onde elas se ligam aos elementos em que atuam. Como os sítios ativos são muito específicos no que diz respeito ao conjunto de elementos que podem se ligar, conclui-se que estão entre as regiões que mais estão conservadas entre as proteínas de uma mesma família, pois sua destruição ocasiona uma perda de funcionalidade. Dessa forma, a identificação de todas as regiões conservadas em uma família de proteínas fornece aos biólogos pistas sobre as interações mecânicas e reações físico-químicas que ocorrem nas células, o que permite diminuir os custos em laboratórios com a realização de testes mais específicos.

Entretanto, para que seja possível prever informações estruturais ou funcionais de uma proteína desconhecida, é necessário que proteínas previamente estudadas sejam armazenadas em bancos de dados acessíveis para futuras análises via alinhamentos.

Atualmente, existem vários bancos de dados com o propósito de fornecer dados para alinhamentos. Cada um dos bancos de dados armazena um aspecto particular acerca da seqüência biológica em questão. Abaixo são listados os bancos de dados mais importantes.

1. **Entrez:** sistema de busca e recuperação que integra informações de vários bancos de dados como, por exemplo, os citados abaixo (GenBank, DDBJ, EMBL, PDB e Swiss-Prot). As principais informações contidas nas bases de dados dizem respeito a seqüências de nucleotídeos e de proteínas, estruturas macromoleculares tridimensionais, genomas completos, literatura médica (MEDLINE) e outros. Serviços e informações estão disponíveis em <http://www.ncbi.nlm.nih.gov/Entrez/>.
2. **GenBank:** banco de dados de seqüência genética, uma coleção anotada das seqüências de nucleotídeos disponíveis publicamente e atualizadas a cada dois meses. Cada entrada do GenBank inclui uma descrição concisa da seqüência, o nome científico, a taxonomia do organismo e uma tabela de características que identificam regiões de genes e outros locais de significado biológico como unidades de transcrição, locais de mutação e *repeats*. Serviços e informações estão disponíveis em <http://www.ncbi.nlm.nih.gov/Genbank/index.html>.
3. **DNA Data Bank of Japan - DDBJ:** é o banco de dados de DNA oficialmente certificado para armazenar seqüências de DNA de pesquisadores e publicar um número de identificação internacionalmente reconhecido. Armazena dados principalmente de pesquisadores japoneses, mas tam-

bém aceita dados e emite o número de identificação para pesquisadores de outras nacionalidades.

Está bem integrado com o resto do mundo, compartilhando diariamente dados com o EMBL e GenBank. Serviços e informações estão disponíveis em <http://www.ddbj.nig.ac.jp/>.

4. **EMBL:** O *EMBL Nucleotide Sequence Database* pertence ao *European Bioinformatics Institute - EBI* e fornece uma coleção de seqüências de nucleotídeos livremente acessíveis, incluindo as anotações. A versão atual do banco de dados (*Release 96, 03 de setembro de 2008*) está disponível na página do EBI: <http://www.ebi.ac.uk/embl/>.
5. **Protein Data Bank - PDB:** é o único repositório mundial de informação sobre as estruturas tridimensionais de macromoléculas, incluindo proteínas e ácidos nucléicos. Avanços em ciência e tecnologia proporcionaram o crescimento do PDB, podendo ser acessadas as estatísticas e detalhes do crescimento. Além do número de estruturas armazenadas, aumentou-se também a complexidade das informações sobre macromoléculas. Serviços e informações estão disponíveis em <http://www.pdb.org>.
6. **SWISS-PROT:** Banco de dados de seqüências protéicas fundado em 1986 que provê alto nível de anotação, com redundância mínima e integração com outros bancos de dados. Juntamente com o UniProtKB/TrEMBL constitui um componente de pesquisa universal de proteína, permitindo fácil acesso a todas as informações publicamente disponíveis a respeito de seqüências de proteínas. Mantido por colaboração entre *Swiss Institute for Bioinformatics - SIB* e o EBI. O banco de dados atual (*Release 56, 23 de setembro de 2008*) e outras informações estão disponíveis em <http://www.ebi.ac.uk/swissprot/>.

1.2 Motivação

A tarefa de alinhar várias seqüências biológicas gerou muitos trabalhos de pesquisa durante as duas últimas décadas. Esses trabalhos, na sua maioria, realizam o alinhamento de múltiplas seqüências biológicas analisando-as duas a duas. Tem-se como exemplo o CLUSTALW (Thompson et al. 1994a.) que, atualmente, está na versão 2.0 (Larking et al. 2007). Apesar do CLUSTAL ser uma ferramenta consolidada, outras heurísticas têm surgido para melhorar a proposta de alinhamento progressivo com o intuito de tornar mais

preciso o alinhamento final, tais como o PRP (Gotoh 1996) e o MUSCLE (Edgar 2004).

Começaram a surgir então outros softwares com o propósito de realizar o alinhamento múltiplo de proteínas analisando todas elas ao mesmo tempo, tais como o SAGA (Notredame & Higgins 1996) e o DIALIGN (Mongengstern et al. 1998).

A principal motivação desta dissertação é a criação de um modelo que forneça um bom alinhamento múltiplo, não utilize a estratégia de alinhamento progressivo e cujo conhecimento sobre a evolução das seqüências biológicas analisadas não exista ou é incompleto.

1.3 Objetivos e Contribuições

O presente trabalho tem como objetivo:

- Especificar uma instância do GAADT para alinhamento múltiplo de proteínas. A especificação descreve tanto as estruturas de tipos quanto os operadores genéticos de cruzamento e mutação;
- Implementar um modelo utilizando uma linguagem orientada a objetos;
- Verificar a capacidade do modelo em realizar o alinhamento múltiplo.

1.4 Organização da Dissertação

Para cumprir os objetivos desta dissertação, organizou-se este trabalho em sete capítulos, incluindo esta introdução.

O capítulo 2 apresenta uma introdução à biologia molecular para que seja possível compreender detalhes acerca da síntese de proteínas.

O capítulo 3 está relacionado a comparação de seqüências, com descrições de métodos de alinhamento múltiplo, dentre eles o CLUSTAL, que é mais utilizado atualmente, e o SAGA, que é um método que utiliza algoritmos genéticos.

O capítulo 4 descreve o modelo de algoritmos genéticos baseado em tipos abstrato de dados proposto no trabalho de Vieira (2003). Para o melhor entendimento de como funciona a especificação dos tipos e dos operadores genéticos, optou-se por utilizar um exemplo fora da área de biologia molecular.

O capítulo 5 descreve a modelagem do algoritmo genético baseado em tipos abstratos de dados para o problema tratado nesta dissertação.

O capítulo 6 mostra os resultados da análise estatística do modelo.

O capítulo 7 apresenta as conclusões da dissertação e indica potenciais melhorias que podem ser aplicadas em trabalhos futuros.

Capítulo 2

Introdução à Biologia Molecular

A presente dissertação está diretamente relacionada a conceitos que envolvem biologia molecular e alinhamento de seqüências. Este capítulo introdutório traz conceitos de biologia molecular e algumas notações para que o leitor tenha uma visão geral dos termos tratados no decorrer do texto.

2.1 A estrutura do DNA

O **DNA** é uma molécula de cadeia longa constituída de uma variedade de unidades individuais, chamadas monômeros. O conjunto de monômeros numa cadeia extensa é denominado **polímero**. Para a compreensão das propriedades físico-químicas do DNA é preciso conhecer seus monômeros, cujas unidades básicas são constituídas por **nucleotídeos**. **Nucleotídeos** são constituídos por uma molécula de açúcar, uma base nitrogenada e ácido fosfórico.

- Molécula de Açúcar: é uma pentose chamada 2'desoxirribose onde os 5 átomos de carbono encontram-se em forma anelar (Brown 1999) (figura 2.1).

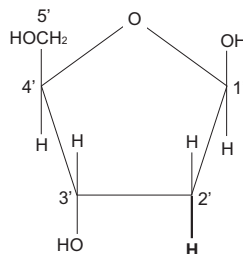


Figura 2.1: Molécula de açúcar do DNA

Os átomos de carbono são numerados como 1', 2', 3', 4' e 5', lê-se: um linha, dois linha, três linha, quatro linha e cinco linha. O termo *linha* é

usado para diferenciar os átomos de carbono da molécula de açúcar e os átomos de carbono da base nitrogenada, bem como identificar em qual posição do anel de açúcar os outros componentes do nucleotídeo estão ligados.

A denominação 2' desoxirribose do açúcar indica alteração na estrutura padrão da ribose por substituição do grupamento hidroxila (-OH) ligado ao átomo de carbono 2' por um hidrogênio (-H).

- Bases Nitrogenadas: são estruturas químicas no formato anelar podendo ser uma purina de anel duplo (adenina ou guanina) ou uma pirimidina de anel simples (citosina ou timina), ligada ao carbono 1' do açúcar (figura 2.2).

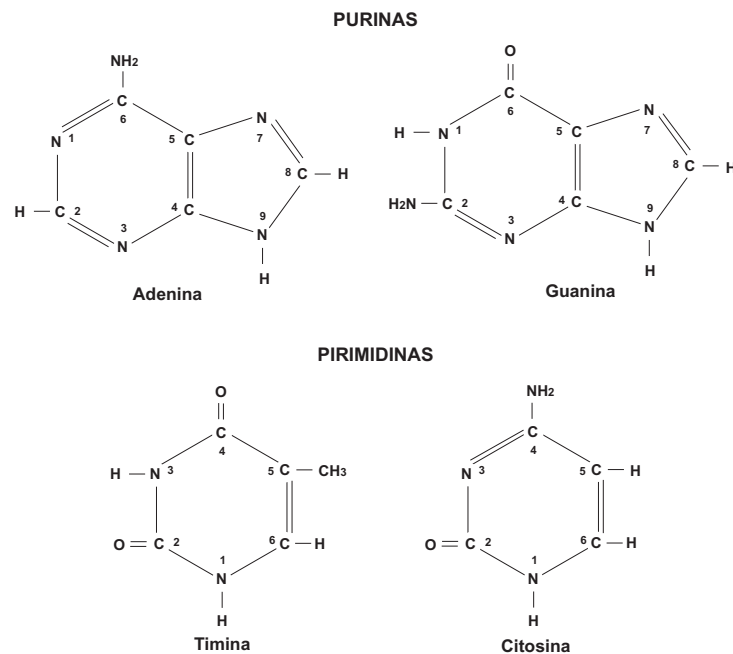


Figura 2.2: Bases nitrogenadas

- Ácido Fosfórico: no DNA são ligados em série (alpha, beta e gama, respectivamente) formando o trifosfato (figura 2.3).

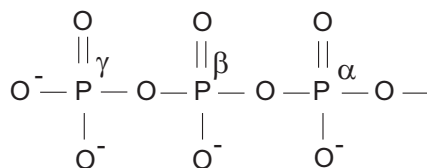


Figura 2.3: Ácido fosfórico

Quando a molécula de açúcar está unida a uma base nitrogenada é denominada uma molécula de nucleosídeo. Para formar um nucleotídeo é preciso que um grupamento ácido fosfórico esteja ligado ao carbono 5' do açúcar, conforme pode ser visto na figura 2.4. Os nucleotídeos podem ser componentes de ácidos nucléicos ou moléculas independentes (Brown 1999).

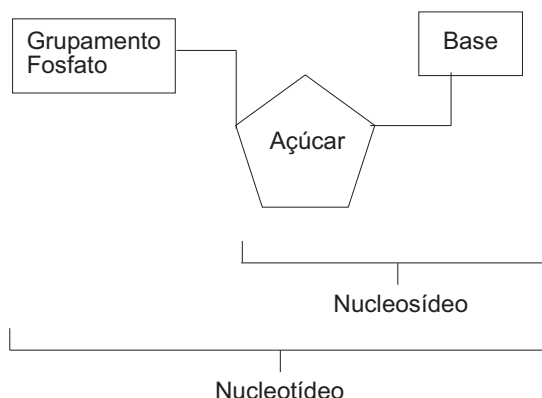


Figura 2.4: Composição do nucleotídeo

Para simplificar a nomenclatura dos nucleotídeos é usada a abreviação da base nitrogenada que o identifica (Tabela 2.1), mas para uma modelagem computacional menos abstrata é importante enfatizar que cada nucleotídeo é uma molécula complexa.

| Base Nitrogenada | Abrev. | Estrutura | Nomenclatura do Nucleotídeo | Abrev. |
|------------------|--------|--------------|---------------------------------|--------|
| Adenina | A | Anel Duplo | 2-desoxiadenosina 5'-trifosfato | dATP |
| Citosina | C | Anel Simples | 2-desoxicitidina 5'-trifosfato | dCTP |
| Guanina | G | Anel Duplo | 2-desoxiguanosina 5'-trifosfato | dGTP |
| Timina | T | Anel Simples | 2-desoxitimidina 5'-trifosfato | dTTP |

Tabela 2.1: Nomenclatura dos nucleotídeos de DNA

O polímero linear do DNA é formado pela união dos nucleotídeos por **ligações fosfodiéster** (figura 2.5), através da ligação do grupamento alpha fosfato, acoplado ao carbono 5' de um dos nucleotídeos, ao carbono 3' do próximo nucleotídeo na cadeia (Brown 1999). Isto torna distintas as terminações dos polinucleotídeos, onde a leitura pode ser realizada tanto no sentido 5' para 3' como no sentido 3' para 5'.

A partir do conhecimento que o DNA contém polinucleotídeos, duas descobertas influenciaram na determinação da estrutura do DNA. A primeira foi a descoberta das proporções de bases de Chargaff (Chargaff 1950), cuja experiência mostrou que o número de adeninas era igual ao número de timinas e

que o número de guaninas era igual ao de citosinas. Além disso, o total de purinas (A+G) é igual ao número de pirimidinas (T+C).

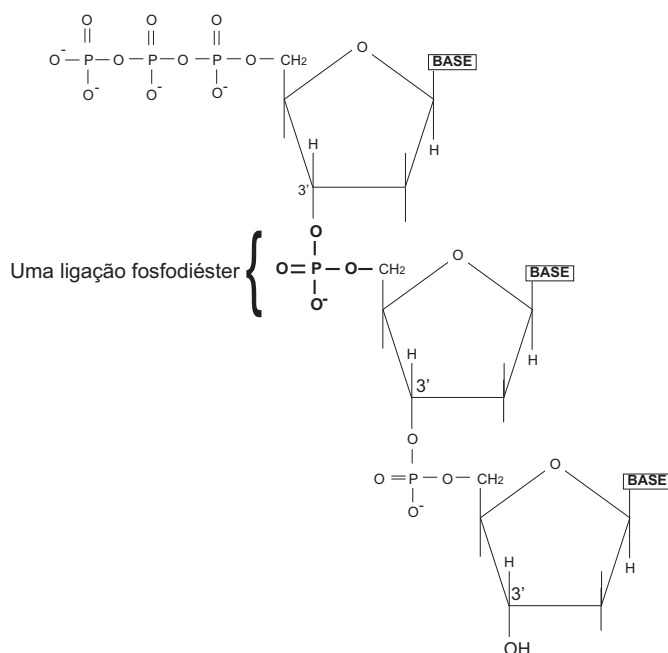


Figura 2.5: Ligação fosfodiéster

A segunda foi a análise por difração de raio X indicando que o DNA é uma molécula helicoidal.

Assim, sem entrar em maiores detalhes, destacam-se algumas características do DNA segundo Brown (1999):

1. A dupla hélice contém dois polinucleotídeos;
2. As bases nitrogenadas dos dois polinucleotídeos interagem através de pontes de hidrogênio;
3. As bases nitrogenadas estão empilhadas no interior da hélice;
4. Ocorrem dez pares de bases por giro da hélice, com pequenas variações descobertas após a pesquisa de Watson & Crick (1953);
5. Os dois filamentos da dupla hélice têm sentidos inversos (3'5' - 5'3');
6. A dupla hélice possui dois sulcos diferentes;
7. Na forma descoberta por Watson e Crick, a dupla hélice possui giro para a direita¹.

¹Rosalind Franklin foi a personagem principal dessa descoberta, pois foi quem realizou análises de difração de raio-x que indicaram a estrutura do DNA. Contudo, não foi reconhecida e não pôde ser premiada após sua morte.

A explicação para as proporções de bases de Chargaff está diretamente relacionada à atração entre os dois polinucleotídeos da dupla hélice, de tal modo que uma adenina em um dos polinucleotídeos é atraída pela timina no outro filamento, assim como a guanina é atraída pela citosina. Esta correspondência de pareamento das bases também é chamada complementaridade de bases (figura 2.6).

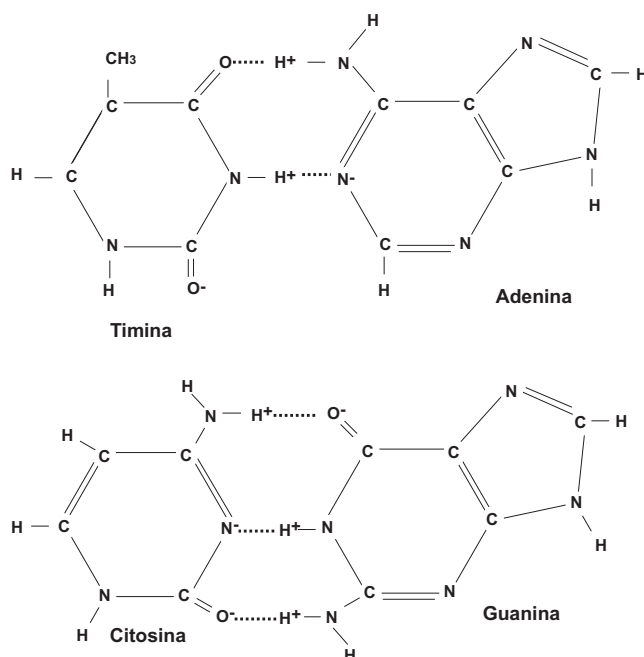


Figura 2.6: Pareamento de bases

Mostrada a estrutura do DNA, é relevante mencionar que ele transporta informação biológica através dos genes. **Genes** são segmentos de DNA que, por si só, não possuem função bioquímica, mas carregam a informação necessária para os processos que ocorrem no interior da célula.

Todas as células de certo organismo armazenam a mesma informação genética, porém apenas um conjunto de genes é expresso em cada tipo celular. Por exemplo, a informação gênica das proteínas produzidas pelo linfócito está presente em todo organismo, mas só é expressa em células sanguíneas.

A seqüência de nucleotídeos é a característica crucial do gene. A informação biológica é transportada por um dos polinucleotídeos da dupla hélice, chamado de **filamento molde**. O filamento molde atua exatamente como um modelo para a síntese protéica, da mesma forma que apresenta o conteúdo para a produção de moléculas RNA.

Além do filamento molde, proteínas auxiliam a produção de moléculas de RNA, que por sua vez fornece auxílio mútuo em todas as fases da expressão

gênica. Dessa forma, torna-se necessário apresentar a estrutura do RNA antes de mostrar como as proteínas são produzidas.

2.2 A estrutura do RNA

RNA também é um polímero, contudo seu açúcar é uma ribose. A base nitrogenada timina não está presente no RNA, ela é substituída pela **URACILA** que é complementar ao nucleotídeo Adenina. Estas diferenças contidas no RNA são apresentadas na figura 2.7.

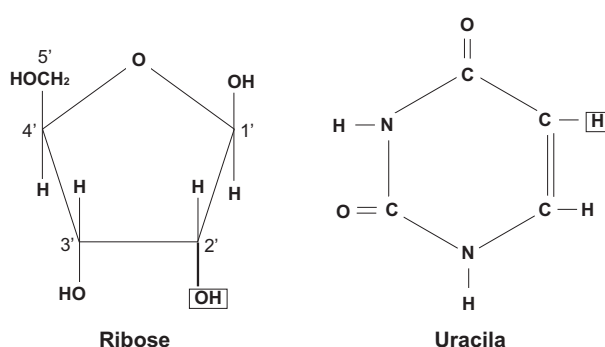


Figura 2.7: Componentes do RNA que diferem do DNA

O fato do açúcar do RNA não ser uma ribose alterada diferencia a estrutura e a nomenclatura dos nucleotídeos do RNA com relação aos nucleotídeos do DNA, a nomenclatura é descrita na tabela 2.2.

| Base Nitrogenada | Abrev. | Estrutura | Nomenclatura do Nucleotídeo | Abrev. |
|------------------|--------|--------------|-----------------------------|--------|
| Adenina | A | Anel Duplo | adenosina 5'-trifosfato | ATP |
| Citosina | C | Anel Simples | citidina 5'-trifosfato | CTP |
| Guanina | G | Anel Duplo | guanosina 5'-trifosfato | GTP |
| Uracila | U | Anel Simples | uridina 5'-trifosfato | UTP |

Tabela 2.2: Nomenclatura dos nucleotídeos de RNA

O polímero de RNA, semelhante ao que ocorre no DNA, apresenta ligações 3'-5' fosfodiéster para unir os nucleotídeos. Existem três classes de RNA e a conformação espacial da molécula difere para cada uma delas.

2.2.1 Classes de RNA

2.2.1.1 RNA mensageiro ou mRNA

O RNA mensageiro é uma molécula unifilamentar que é produzida durante a primeira fase da expressão gênica, denominada transcrição, a partir do fila-

mento molde de DNA. Em geral, as moléculas de mRNA não têm vida longa na célula, servindo apenas de molde para a tradução em proteínas (seção 2.4.2).

Uma possível explicação para a designação RNA mensageiro deve-se ao fato de que os genes são localizados no núcleo celular, enquanto as proteínas são sintetizadas nos ribossomos, que se localizam no citoplasma. Dessa forma, o mRNA é a molécula responsável pela transmissão de informação do núcleo ao citoplasma.

Não há razão para detalhar a estrutura do mRNA isoladamente sem mencionar o processo de transcrição apresentado na seção 2.4.1. Pela importância da molécula de mRNA como transmissor de informação sua designação muitas vezes é abreviada somente para RNA.

2.2.1.2 RNA ribossomal ou rRNA

Os rRNAs juntamente com proteínas ribossomais compõem os **ribossomos**. Assim, pode-se referir aos ribossomos como estruturas multimoleculares cuja principal função é servir de sítio para a tradução (seção 2.4.2), o que faz com que sejam numerosos na maioria das células (Brown 1999).

Ribossomos dos seres eucarióticos² são maiores que os ribossomos dos seres procarióticos³ e boa parte das suas proteínas são diferentes, sendo quatro moléculas de rRNA nos eucarióticos e três nos procarióticos.

Os ribossomos são estruturas grandes e estimar sua massa molecular não é algo exato. Por sua vez, o tamanho destas estruturas é determinado por uma medida da taxa de sedimentação de um componente em uma centrífuga, relacionando o peso molecular à forma 3-D do componente, tal medida é denominada **unidade Svedberg** (abreviada pela letra **S**).

Quanto à estrutura, os ribossomos são formados de duas subunidades de tamanhos desiguais, referenciadas como as subunidades maior e menor. As figuras 2.8 e 2.9 representam uma abstração de um ribossomo procariótico e eucariótico, respectivamente.

No ribossomo procariótico (figura 2.8), a subunidade maior possui 34 proteínas e duas moléculas de rRNA, uma de 5S com 120 nucleotídeos e outra de 23S com 2904 nucleotídeos. Esta subunidade apresenta coeficiente de sedimentação⁴ de **50S**. A subunidade menor contém apenas um rRNA de 16S com 1541 nucleotídeos e unido com 21 proteínas tem coeficiente de sedimentação **30S**. As unidades 50S e 30S unidas formam o ribossomo procariótico **70S**.

²Eucarióticos: organismos que possuem membrana nuclear ou carioteca.

³Procarióticos: organismos que não possuem membrana nuclear, ou seja, não possuem uma nuclear propriamente definida.

⁴Coefficiente de sedimentação: refere-se à unidade Svedberg.

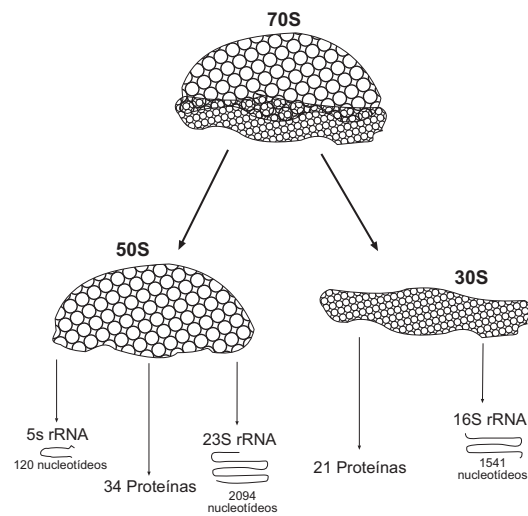


Figura 2.8: Ribossomo procariótico

Essas quantidades de nucleotídeos e proteínas informadas acima podem ter pequenas variações mesmo entre ribossomos procarióticos. Entretanto, para referenciar o ribossomo optou-se por expor as características de *E.coli*, coletadas de Brown (1999).

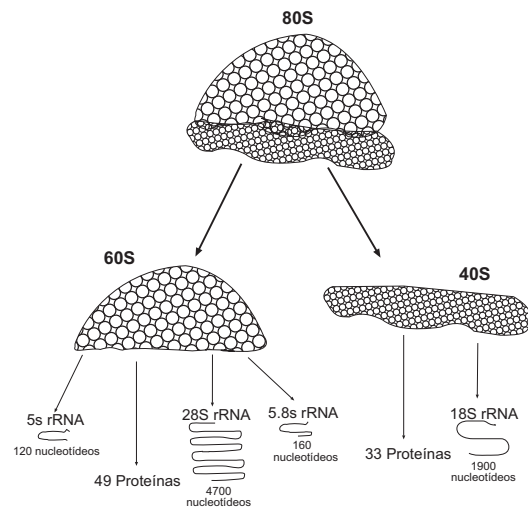


Figura 2.9: Ribossomo eucariótico

O comprimento dos rRNAs eucarióticos e o número de proteínas que se ligam às subunidades também não são os mesmos para todas as espécies. Contudo, todos os ribossomos eucarióticos tem em comum muitas características estruturais e químicas e, dessa forma, foram também coletadas as informações de Brown (1999), as quais referem-se especificamente aos ribossomos de mamíferos.

No ribossomo eucariótico (figura 2.9), a subunidade maior possui 50 proteí-

nas e três moléculas de rRNA: 5S (com 120 nucleotídeos), 28s (com aproximadamente 4718 nucleotídeos) e 5,8S (com 160 nucleotídeos). Esta subunidade apresenta coeficiente de sedimentação **60s**. A subunidade menor tem um único rRNA de 18S (com 1874 nucleotídeos) e unido com 33 proteínas tem coeficiente de sedimentação **40S**. Combinadas, as unidades 60S e 40S formam o ribossomo eucariótico **80S**.

As subunidades ribossomais se interligam quando é necessário sintetizar proteínas e desempenham papel importante nos sinais de início da tradução. Quando a subunidade menor e a subunidade maior estão acopladas, apresentam três sítios de ligação: sítio E (exit), sítio P (peptidil) e sítio A (aminoacil), respectivamente da esquerda para direita como mostra a figura 2.10. Tais sítios são fundamentais para o processo de síntese de proteínas como será visto mais adiante.

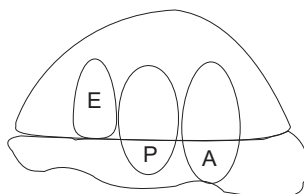


Figura 2.10: Ribossomo e seus sítios

2.2.1.3 RNA transportador ou tRNA

Os tRNAs são moléculas adaptadoras cuja principal função é reconhecer a enzima para a ligação do **aminoácido**⁵ durante a tradução. Quando os nucleotídeos de DNA são transcritos em uma seqüência de RNA formando uma molécula de tRNA, a conformação 2D apresenta uma estrutura chamada trevo, contendo entre 74 e 95 nucleotídeos. Cada organismo transcreve uma quantidade de tRNAs diferentes, porém todos os tRNAs assumem a mesma estrutura após a síntese, sendo constituída pelos componentes listados a seguir e ilustrados na figura 2.11.

Braço acceptor É formado por uma série de, em média, 7 pares de bases, pertencentes a ponta 3' e 5' do tRNA. Na formação da proteína a unidade em formação se liga ao braço acceptor;

Braço D ou DHD Sua extremidade é quase que invariavelmente formada por uma pirimidina modificada denominada diidrouracila;

⁵Proteínas são estruturas formadas por monômeros denominados aminoácidos.

Braço anticódon O tRNA possui uma seqüência de três nucleotídeos (o anticódon) complementar a uma seqüência de três nucleotídeos no mRNA. Por isso esse braço recebe a denominação de anticódon, tendo um papel imprescindível na decodificação das unidades de proteínas. Cada tRNA é específico de um aminoácido (ex: a designação tRNA^{trp} corresponde a um tRNA específico de triptofano), porém pode existir mais de um tRNA para cada um dos aminoácidos.

Braço extra, opcional ou variável Pode ser uma alça de apenas 3 a 5 nucleotídeos, ou uma haste-alça de 13 a 21 nucleotídeos com até cinco pares de base na haste;

Braço T ψ C Esse braço sempre contém a seqüência de nucleotídeos T, ψ , C (ψ é um nucleotídeo contendo a pseudouracila).

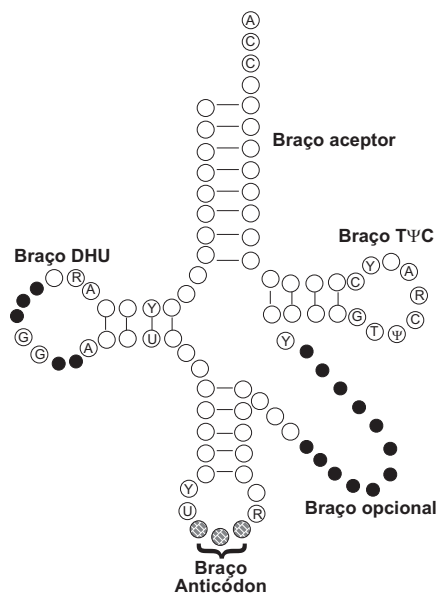


Figura 2.11: RNA transportador

Pode-se observar que o tRNA não contém apenas os nucleotídeos de RNA mencionados no início da seção 2.4 (A, U, C, G). Uma característica do tRNA é o seu alto número de bases modificadas após a transcrição: inosina (I), pseudouridina (ψ), diidrouridina (UH2 ou D), ribotimina (T) e derivados metilados de guanossina e inosina.

2.3 Código genético

Após a descoberta da dupla hélice do DNA na década de 50, veio à tona que os genes são colineares com as proteínas que codificam, ou seja, a ordem de nucleotídeos no gene se correlaciona diretamente com a ordem de aminoácidos na cadeia protéica (Brown 1999).

A tabela 2.3 apresenta os 20 aminoácidos que codificam proteínas, cada linha da tabela apresenta o nome do aminoácido, a abreviação padrão com três letras e a abreviação de uma letra (normalmente utilizada nos algoritmos para designar a seqüência de proteínas).

| Aminoácido | 3 letras | 1 letra |
|------------------------------|-----------------|----------------|
| Ácido aspártico ou Aspartato | asp | D |
| Ácido glutâmico ou Glutamato | glu | E |
| Alanina | ala | A |
| Arginina | arg | R |
| Asparagina | asn | N |
| Cisteína | cis ou cys | C |
| Fenilalanina | fen ou phe | F |
| Glicina | gli ou gly | G |
| Glutamina ou Glutamida | gln | Q |
| Histidina | his | H |
| Isoleucina | ile | I |
| Leucina | leu | L |
| Lisina | lis ou lys | K |
| Metionina | met | M |
| Prolina | pro | P |
| Serina | ser | S |
| Tirosina | tir ou tyr | Y |
| Treonina | tre ou thr | T |
| Triptofano ou Triptofana | trp | W |
| Valina | val | V |

Tabela 2.3: Os 20 aminoácidos que existem na natureza

Como o gene codificaria uma proteína era uma questão freqüente, algumas hipóteses foram abordadas. A primeira hipótese era que cada nucleotídeo de RNA codificaria uma proteína, porém a quantidade destes nucleotídeos não eram suficientes para codificar 20 proteínas. A outra possibilidade era que os aminoácidos eram codificados por combinações de dois nucleotídeos o que acarretaria em apenas 16 aminoácidos, também não sendo suficientes.

Contudo, a próxima hipótese seria um código de trincas, a combinação de cada três nucleotídeos iria produzir 64 possíveis aminoácidos, o que seria mais que suficiente. Experimentos comprovaram que o código genético é

fundamentado na seqüência de nucleotídeos de RNA e que os aminoácidos são codificados por exatamente três nucleotídeos.

Esses grupos de três nucleotídeos de RNA são denominados **códons** e são pareados por complementariedade de bases com o braço anti-códon do tRNA durante a tradução.

Conforme pode ser visto na tabela 2.4, o código genético é não-ambíguo, ou seja, qualquer que seja a trinca ela será traduzida em um e somente um aminoácido.

| | | Segunda Base do Códon | | | | |
|-------------------------------|----------|------------------------------|----------|----------|----------|----------|
| | | U | C | A | G | |
| Primeira Base do Códon | U | Fen | Ser | Tir | Cis | U |
| | | Fen | Ser | Tir | Cis | C |
| | | Leu | Ser | (Stop) | (Stop) | A |
| | | Leu | Ser | (Stop) | Trp | G |
| | C | Leu | Pro | His | Arg | U |
| | | Leu | Pro | His | Arg | C |
| | | Leu | Pro | Gln | Arg | A |
| | | Leu | Pro | Gln | Arg | G |
| | A | Ile | Tre | Asn | Ser | U |
| | | Ile | Tre | Asn | Ser | C |
| | | Ile | Tre | Lis | Arg | A |
| | | Met (Start) | Tre | Lis | Arg | G |
| | G | Val | Ala | Asp | Gli | U |
| | | Val | Ala | Asp | Gli | C |
| | | Val | Ala | Glu | Gli | A |
| | | Val | Ala | Glu | Gli | G |

Tabela 2.4: Código genético

Contudo, o mesmo aminoácido pode ser especificado por códons diferentes, existindo assim códons sinônimos. Com exceção da metionina e do triptofano, todos os demais aminoácidos possuem mais de um códon. Alguns códons sinônimos estão agrupados em famílias (como exemplo os códons CCU, CCC, CCA, CCG todos codificam a prolina). Devido a existência de códons sinônimos diz-se que o código genético é degenerado ou redundante.

Isso torna o código flexível a falhas de replicação ou mutação, uma vez que se esses processos acarretaram na mudança de algum nucleotídeo, o organismo ainda pode ser capaz de especificar o mesmo aminoácido, tendo chances de não perder a função protéica correspondente.

Acreditava-se que tal esquema codificava todos os genes. Não obstante, foi descoberto que os genes da mitocôndria humana utilizam um código ligeiramente diferente, conforme tabela 2.5. Alega-se, desde então, que o código genético é quase universal.

| Organismo | Genes | Códon | Significado Universal | Significado Verdadeiro |
|--------------|---------------------|------------------------|---------------------------|---------------------------|
| Mamíferos | Mitocondriais | UGA AGA, AGG AUA | Finalização Arg ile | Tri Finalização Met |
| Protozoários | Todos Nucleares | UAA, UAG | Finalização | Gln |
| Mamíferos | Glutaião peroxidase | UGA | Finalização | SeCis |

Tabela 2.5: Código diferente do código universal

Para efeito de curiosidade, algumas anomalias quanto ao código também foram descobertas para alguns protozoários, os quais usam um código genético fora do padrão para seus genes nucleares. Como exemplo a *Tetrahyaena* e *Paramecium*, onde dois dos códons finalizadores especificam glutamina (tabela 2.5).

Mais intrigante ainda foi a descoberta de que o significado de um códon pode variar de gene para gene no mesmo organismo, como a codificação do códon de finalização UGA em genes humanos, que para as enzimas glutaião peroxidase e iodotironina 5'-desiodinase especifica um aminoácido incomum chamado selenocisteína (uma cisteína com o átomo de enxofre substituído por selênio).

Na maioria dos genes trabalha-se com o código descrito acima, mas pesquisas continuam em andamento para verificar se há outros exemplos de códons que possuem significados diferentes nos diferentes genes.

2.4 Expressão gênica

A expressão gênica envolve dois processos principais: transcrição e tradução, cujos passos envolvem o reconhecimento do filamento molde que é transcrito em mRNA e o reconhecimento da sua seqüência que deve ser traduzida em proteínas, respectivamente descritos nas seções 2.4.1 e 2.4.2. Como resultado destes processos, a seção 2.4.3 apresenta mais detalhes sobre proteínas.

2.4.1 Transcrição

Todos os genes passam pelo processo biológico denominado transcrição. A transcrição ocorre no interior da célula toda vez que é necessária a produção de um produto gênico.

Dentre as moléculas de RNAs citadas, o RNA ribossomal e o RNA transportador já são produtos finais, ou seja, desempenham suas funcionalidades como moléculas de RNA, sendo também chamados RNA estáveis. Por outro

lado, o RNA mensageiro não tem vida longa, estando presente nas duas fases da expressão gênica: transcrição e tradução.

2.4.1.1 O Processo de transcrição

O processo de transcrição consiste em três etapas: iniciação, alongamento e término. Quando o organismo necessita de certo produto gênico uma enzima atua para acelerar o processo e, no caso de ser necessário sintetizar um mRNA, a enzima responsável é denominada **RNA polimerase II**. Esta enzima tem a função de reconhecer uma região específica (região promotora) que antecede a seqüência gênica.

A região promotora, também chamada de promotor, é uma seqüência de nucleotídeos de DNA que é reconhecida não somente pela RNA polimerase II, mas também por outras moléculas que servem de auxílio na transcrição, denominadas fatores de transcrição.

A iniciação da transcrição ocorre quando a RNA polimerase II e os fatores de transcrição atuam na região promotora com a finalidade de reconhecer os sítios de ligação para formar o complexo de iniciação. Desta forma, os ativadores transcricionais podem ativar o complexo para que a RNA polimerase escape do promotor para o reconhecimento da seqüência codificadora.

Após a ligação, começa a etapa de alongamento e desenrolamento da dupla hélice. A RNA polimerase migra ao longo da molécula de DNA, possibilitando deste modo a separação das duas fitas, onde uma pequena parte entre as fitas é gradualmente dissociada, permitindo que a dupla hélice volte ao seu estado original após a transcrição do gene.

A reação química para a formação da molécula de RNA é modulada pela presença do filamento molde, lido no sentido 3'-5' e orientando a ordem na qual são polimerizados os ribonucleotídeos individuais no RNA (Griffiths et al. 2001). A cada nucleotídeo de DNA que o complexo de transcrição reconhece é adicionado um novo ribonucleotídeo à extremidade 3' livre do polímero em formação, por complementaridade com a seqüência nucleotídica da cadeia simples de DNA molde.

O processo termina quando a enzima RNA polimerase II reconhece uma seqüência de terminação específica. Assim, o mRNA sintetizado é transportado do núcleo para o citoplasma da célula, onde ocorre a tradução da informação genética nele contida.

Contudo, em células eucarióticas o transcrito é chamado pré-mRNA e ainda passa por três etapas:

1. Modificação da extremidade 5': adição de uma estrutura chamada *cap 5'*.

A presença desta estrutura aumenta a estabilidade do mRNA e influencia a remoção de regiões não codificantes (introns);

2. Poliadenilação: são adicionados à extremidade 3', de 50 a 250 adeninas formando uma calda polyA, para proporcionar estabilidade e aumentar o tempo pelo qual o mRNA permanece intacto e disponível para a tradução antes da degradação;
3. *Splicing*: remoção das regiões não codificantes (introns) do mRNA precursor e união das regiões codificantes (exons).

Por fim, o polinucleotídeo torna-se maduro, sendo transportado do núcleo para o citoplasma, para que a informação gênica seja traduzida em proteína.

2.4.2 Tradução

As seções anteriores tiveram como escopo introduzir alguns conceitos de biologia molecular relevantes para a explanação da expressão gênica. Como esse trabalho aborda proteínas é de fundamental que haja um reflexo de como funcionam os processos biológicos. Dessa forma, os símbolos usados como aminoácidos não devem ser vistos apenas como caracteres. A seguir são descritos os passos da formação da cadeia protéica.

2.4.2.1 O Processo de tradução

Quando o organismo necessita de uma dada proteína o processo começa através da transcrição que resulta no RNA organizado em códon. A finalidade da tradução então é servir de dicionário para que a seqüência de nucleotídeos do mRNA seja acoplada com aminoácidos específicos que formarão a cadeia polipeptídica (seção 2.4.3).

Contudo, cada códon do RNA mensageiro e o respectivo aminoácido são incapazes de se reconhecerem automaticamente, a molécula de tRNA (seção 2.2.1.3) é a verdadeira responsável para que a mensagem chegue ao seu destino e sintetize o aminoácido desejado.

Tratando em termos moleculares, cada célula contém vários tipos de tRNAs distintos uns dos outros pelas diferentes seqüências, apesar de conservarem os nucleotídeos invariantes e semi-invariantes descritos na seção 2.2.1.3. Cada tRNA forma uma ligação com o seu aminoácido e com nenhum outro mais, podendo reconhecer e se ligar a um códon que especifica aquele aminoácido.

O processo biológico que efetiva essa ligação é denominado **aminoacilação**, a qual é catalisada por um grupo de enzimas chamadas aminoacil-tRNA sintetases, resultando no aminoácido ligado à extremidade do braço acceptor do tRNA (figura 2.12).

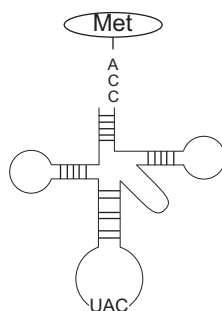


Figura 2.12: Aminoácido ligado ao tRNA

Uma vez que o aminoácido correto está ligado ao braço acceptor do tRNA, a molécula aminoacilada tem seu requisito alvo pronto para atuar na tradução. Contudo, o reconhecimento do códon é uma tarefa específica da alça anticódon do tRNA, pois é a parte do tRNA que contém o trinucleotídeo denominado **anticódon** que é complementar ao códon do mRNA e pode, desta forma, ligar-se ao mRNA através do pareamento de bases.

O processo de tradução consiste em três etapas: iniciação da tradução, alongamento da cadeia polipeptídica e finalização da cadeia.

A tradução é iniciada com a formação do complexo de iniciação. Quando os ribossomos não estão envolvidos na tradução suas subunidades permanecem dissociadas, isso é devido a necessidade de outros componentes se interligarem a ele para a formação do complexo de iniciação.

Dessa forma, a subunidade menor se liga a uma molécula de mRNA em um ponto específico que antecede o códon de iniciação do gene. O ponto exato de ligação é sinalizado pelo sítio de ligação do ribossomo que localiza a seqüência consenso.

Localizada a seqüência consenso a subunidade menor se move ao longo do mRNA até encontrar a mensagem de início de tradução sinalizada pelo códon AUG. Uma vez identificado o códon de iniciação ele passa a localizar-se numa região do ribossomo denominada sítio P (peptidil).

O reconhecimento do códon é então realizado por um tRNA específico, cuja alça anticódon contém o trinucleotídeo complementar UAC que será pareado com as bases AUG do mRNA. Neste caso inicial, o aminoacil-tRNA ligado carrega o aminoácido metionina, pois é quem codifica o códon AUG (figura 2.13).

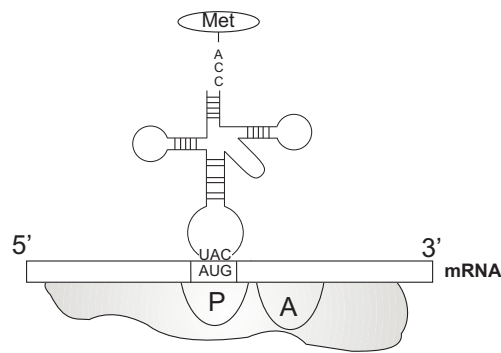


Figura 2.13: Formação do complexo de pré-iniciação

Além das ligações de moléculas descritas acima, existem outras proteínas que não são as ribossomais, mas são imprescindíveis para a formação do complexo de iniciação, denominadas **fatores de iniciação**. Essa estrutura formada por: subunidade menor, fatores de iniciação, mRNA e tRNA aminoacilado é chamada complexo de pré-iniciação (figura 2.13). Para que o complexo de iniciação fique completo a subunidade maior do ribossomo é ligada (figura 2.14), assim formando três sítios:

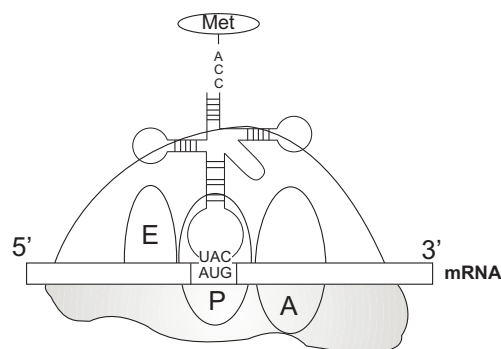


Figura 2.14: Formação do complexo de iniciação

- Sítio Peptidil ou P: onde se localiza o tRNA que especifica o códon deste sítio;
- Sítio Aminoacil ou A: está posicionado sobre o segundo códon do gene e, a priori, está vazio;
- Sítio Exit ou E: onde se posiciona o tRNA que irá liberar o aminoácido para a formação da cadeia polipeptídica.

A próxima etapa é a fase de alongamento da cadeia polipeptídica, cujo primeiro passo é o encaixamento do segundo aminoacil-tRNA no sítio A por pareamento de bases entre o anticódon do tRNA e o segundo códon do mRNA.

Isto requer alguns fatores de alongamento bem como uma molécula GTP para fornecer a energia necessária.

Como o sítio P e o sítio A estão ocupados por tRNAs aminoacilados, os aminoácidos estão em contato direto. Contudo, para que os aminoácidos possam formar um dipeptídeo, é necessário que haja uma ligação peptídica, detalhada na seção 2.4.3, entre o grupamento carboxila do primeiro e o grupamento amino do segundo aminoácido. Neste momento, a enzima **peptidiltransferase** atua para catalisar a formação da ligação peptídica entre os aminoácidos e a enzima **tRNA desacilase** atua para quebrar a ligação tRNA-aminoácido, isso resulta em apenas um tRNA no sítio P e no sítio A um tRNA ainda aminoacilado, mas com um dipeptídeo.

O próximo passo do alongamento diz respeito a **translocação**, onde o ribossomo desliza ao longo do complexo tRNAs-mRNA. O conteúdo do sítio P (tRNA não carregado) é deslocado para o sítio E, bem como o conteúdo do sítio A (complexo *aminoácido 1 - aminoácido 2 - tRNA - mRNA*) é deslocado para o sítio P, de tal forma que o deslocamento ocorre de três em três nucleotídeos com relação ao mRNA. Assim, o códon que resta no sítio A fica sujeito a aceitação de um novo aminoacil-tRNA, sendo que, quando o aminoacil-tRNA forma pares de base com este códon um processo de translocação se completa para que um novo ciclo de alongamento seja repetido.

No decorrer do processo, os demais aminoacil-tRNAs ligam-se primeiramente ao sítio A. Quando o ribossomo é deslocado, eles automaticamente são transferidos para o sítio P e finalmente para o sítio E de onde são liberados.

O alongamento ocorre até que se encontre no sítio A um dos códon de finalização (UAA, UAG ou UGA). Para os três códon de finalização não há moléculas de tRNA com anticódons capazes de realizar o pareamento de bases, isso significa que não há aminoácidos correspondentes, servindo apenas de sinalização para o fim da tradução. A finalização propriamente dita ocorre quando **fatores de liberação** atuam para liberação do polipeptídeo, dissociação das subunidades do ribossomo e liberação do mRNA. A cadeia polipeptídica então se dobra em sua estrutura terciária e começa sua vida funcional dentro da célula.

2.4.3 Proteína

A tradução é a última fase do processo de expressão gênica, da qual resulta a produção da proteína. Os três tipos de RNA descritos na seção 2.2.1 trabalham em conjunto durante a tradução, contudo as proteínas são formadas a partir das seqüências de nucleotídeos da molécula de mRNA e as regras que

determinam qual aminoácido deve ser formado estão contidas no código genético.

Cada aminoácido é constituído por um átomo de carbono central ligado a um grupo amino (lado esquerdo da figura 2.15), um grupo carboxila (lado direito da figura 2.15), um átomo de hidrogênio e um radical denominado radical R ou cadeia lateral.

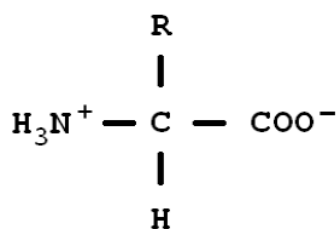


Figura 2.15: Composição do aminoácido

Os aminoácidos se distinguem um dos outros pela formação da cadeia lateral, as quais variam em estrutura, carga elétrica e tamanho.

Um peptídeo é uma estrutura molecular que apresenta um ou mais aminoácidos unidos por ligações peptídicas, sendo que a ligação é formada pela condensação entre o grupamento carboxila de um aminoácido e o grupamento amino de um segundo aminoácido.

Dessa forma, ligação peptídica une dois aminoácidos resultando na formação de um dipeptídeo com a perda de uma molécula de água (figura 2.16). Uma cadeia com vários peptídeos é chamada de **polipeptídeo**, tratando-se da proteína propriamente dita.

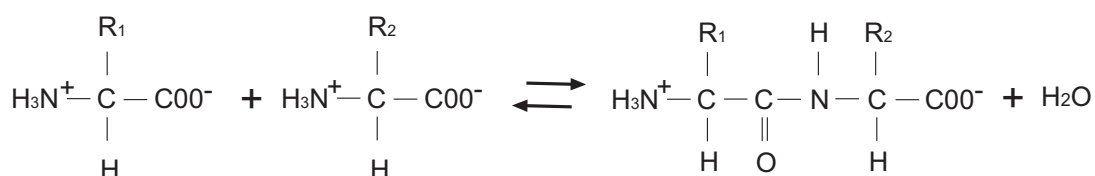


Figura 2.16: Ligação peptídica

Apesar de toda proteína ser formada por seqüência de aminoácidos, sabe-se que sua estrutura é subdividida em quatro níveis estruturais:

Estrutura Primária é formada pela própria seqüência de aminoácidos e as ligações correspondentes, resultando em uma longa cadeia linear com uma extremidade terminal amino e uma extremidade terminal carboxila.

Na maioria dos *softwares* de bioinformática a estrutura primária é identificada pela abreviação de cada aminoácido por uma letra, formando uma cadeia de caracteres. Isto faz com que este nível estrutural seja algo simples de se trabalhar computacionalmente. O fato de abreviar algo que na verdade é uma estrutura química não deixa o resultado menos significativo ou irrelevante, pois na verdade é da simples seqüência de aminoácidos que deriva todo o arranjo espacial da molécula.

Estrutura Secundária o segundo nível estrutural refere-se a interligação de alguns aminoácido próximos que estabilizam uma estrutura não linear por meio principalmente de pontes de hidrogênio. Tal configuração chega a ser regular, de modo a observar os padrões de estrutura que se repetem, dos quais os mais importantes são a hélice α e a folha β (Branden & Tooze 1991). Estas estruturas regulares são estabilizadas por meio de ponte de hidrogênio entre o grupamento carboxila e amino de aminoácidos diferentes.

Estrutura Terciária corresponde a conformação tridimensional formada por dobramento de estruturas secundárias, tal estrutura mantém-se por meio de uma variedade de interações além das pontes de hidrogênio. O dobramento tridimensional natural tenta manter os grupamentos R dos aminoácidos não-polares protegidos da água.

Estrutura Quaternária corresponde a formação de uma multisubunidade formada por dois ou mais polipeptídeos, podendo envolver várias moléculas da mesma cadeia protéica, moléculas semelhantes ou polipeptídeos totalmente diferentes.

Sabe-se que a seqüência primária (seqüência dos aminoácidos) determina cada um dos níveis mais elevados. Na estrutura secundária esta característica é mais clara, pois é possível perceber que certos aminoácidos têm a tendência natural de se ligarem para a formação de hélice α , enquanto outros estimulam a formação de uma folha β , e ainda, certos aminoácidos são comumente encontrados fora de estruturas regulares. Estas características são tão bem compreendidas que várias pesquisas estão em andamento para prever a estrutura secundária assumida pela seqüência de aminoácidos, dentre elas pode-se citar Garnier et al. (1998) e Sen et al. (2005). Já nas estruturas terciária e quaternária as interações entre os aminoácidos são tão complexas que as regras de predição de estrutura são difíceis de identificar.

Dada uma proteína com sua característica estrutural formada por desdobramentos, caso ela seja desnaturada (por exemplo adicionando-se um des-

naturador químico ou através de aquecimento brando), perde seus níveis estruturais e assume uma conformação não organizada. Mas se for renaturada (resfriada novamente) ela ainda conserva a capacidade espontânea de voltar a estrutura correta. Isto ratifica que as propriedades de desdobramento precisam estar contidas na seqüência de aminoácidos.

Uma outra importante ligação do estudo da estrutura primária diz respeito a funcionalidade protéica. Como se pode observar ao longo da seção sobre expressão gênica, em quase todos os processos menciona-se o fato de existir alguma proteína para auxiliá-los.

Tomando como exemplo as proteínas de ligação com o DNA, algumas estruturas tridimensionais são semelhantes em proteínas distintas e são responsáveis pela interação com seqüências específicas do DNA, estas estruturas são designadas **regiões de estrutura conservadas**.

Um tipo de região de estrutura conservada capaz de se ligar ao DNA é a **estrutura hélice-giro-hélice** onde três fragmentos são comuns entre as proteínas com esta estrutura:

1. O primeiro fragmento corresponde a uma hélice α , a qual pode variar dependendo do organismo, mas em geral sua estrutura tem como objetivo o encaixe no sulco maior do DNA para reconhecimento e interação com as bases nitrogenadas;
2. O segundo fragmento não apresenta conformação regular, sendo caracterizada somente por um giro;
3. O terceiro fragmento também é uma hélice α que fica mais separada do DNA, porém forma um ângulo com o primeiro fragmento.

Tais características fazem com que as proteínas desempenhem a mesma função. Se o dobramento estivesse ausente ou orientado incorretamente a capacidade de ligar-se ao DNA seria perdida. Como a funcionalidade depende da estrutura e, por sua vez, a estrutura depende da seqüência primária, temos que a seqüência de aminoácidos também é fundamental para preservar a função protéica.

O estudo de seqüências protéicas também pode se correlacionar ao ato de alinhar seqüências. Para biólogos moleculares, uma das finalidades do alinhamento de seqüências é a análise do resultado obtido para predizer se as seqüências em estudo são homólogas. Dessa forma, antes de partir para o capítulo sobre alinhamento de seqüências, a seção a seguir apresenta uma abordagem mais detalhada do termo homologia.

2.5 Homologia

O termo homologia está diretamente relacionado ao estudo comparativo dos organismos, mais precisamente a evolução das espécies. Entende-se por **evolução** o processo através do qual ocorrem mudanças ou transformações nos seres vivos ao longo do tempo, dando origem a novas espécies.

Homologia é a semelhança entre estruturas de diferentes organismos que se acentuam unicamente por possuírem a mesma origem embriológica. Entretanto, estruturas homólogas podem ou não exercer a mesma função.

Como exemplo de estruturas que têm a mesma origem embriológica pode-se citar o braço do homem, a pata do cavalo, a asa do morcego e a nadadeira da baleia. Todas estas estruturas são homólogas entre si, uma vez que apresentam a mesma origem embriológica, mas não há similaridade funcional.

A **homologia** entre estruturas de organismos distintos sugere que se originaram de um **grupo ancestral comum**, mesmo que não indique um grau de proximidade comum.

Ao contrário da homologia, pode ocorrer de estruturas não apresentarem a mesma origem embriológica, mas devido a sua adaptação para execução da mesma função acabam manifestando certas semelhanças, como é o caso das asas dos insetos e das aves adaptadas ao voo. Essa semelhança entre estruturas de diferentes organismos é devida unicamente à adaptação a uma mesma função, denomina-se **analogia**.

A evolução convergente é caracterizada pela adaptação de diferentes organismos à mesma condição ecológica. Assim, as formas dos corpos do golfinho, do tubarão, dentre outros peixes têm semelhanças devido a adaptação a natação. Isto não induz qualquer grau de parentesco, mas sim o resultado da adaptação desses organismos ao ambiente aquático. A semelhança entre estruturas análogas são consideradas resultado da evolução convergente.

Os termos analogia e homologia são também utilizados para a comparação de seqüências. Analogia de seqüências, assim como colocado no conceito de evolução, ocorre devido à evolução convergente, fazendo com que haja similaridades de função e de seqüências. Homologia de seqüências ocorre quando elas compartilham um ancestral comum e apresentam estruturas semelhantes, mas as seqüências podem ser similares ou não.

O alinhamento possui a finalidade de medir a similaridade entre seqüências possuindo um significado quantitativo, pois um valor final é obtido refletindo uma pontuação para o alinhamento. Enquanto isso, a homologia é a observação qualitativa, pois trata-se de uma inferência obtida com a análise do alinhamento (seqüências são homólogas ou não).

Capítulo 3

Introdução à Comparação de Sequências

O objetivo geral da comparação de seqüências é descobrir informações evolucionárias, estruturais e funcionais em seqüências de proteína ou DNA. Uma das formas de realizar a comparação é evidenciando as similaridades e as diferenças existentes entre as seqüências de organismos distintos, com o intuito de inferir informações funcionais e evolutivas importantes. Para tal, recorre-se muitas vezes à construção de um alinhamento entre as seqüências dos organismos.

3.1 Alinhamento de seqüências

Alinhar seqüências consiste basicamente em colocar uma seqüência sobre a outra de forma que a correspondência entre elas se torne evidente. Os seguintes conceitos são importantes:

- **Similaridade:** medida do quanto as seqüências são parecidas, ou seja, refere-se a porcentagem de nucleotídeos idênticos ou de aminoácidos com propriedades químicas semelhantes;
- **Homologia:** refere-se à relação evolutiva entre seqüências. Duas seqüências são homólogas se são derivadas de uma seqüência ancestral comum.
- **Distância de Edição:** refere-se ao número mínimo de operações (inserções, remoções e substituições de monômeros) para transformar uma seqüência em outra. É importante notar que, quando se observa dois aminoácidos iguais, isto não é contabilizado pela operação de edição.

Ao se comparar seqüências, o interesse pode ser medir a similaridade entre elas ou a distância de edição. As medidas de distância não são apropriadas para a comparação envolvendo subseqüências, ou seja, são restritas apenas a comparação da seqüência como um todo. Dessa forma, o alinhamento por similaridades é um caminho alternativo para aplicações biológicas (Setubal & Meidanis 1997), motivo pelo qual será utilizada esta medida no decorrer do texto.

A tarefa de alinhamento é utilizada tanto na comparação de seqüências de DNA quanto na de proteínas. Para realizar o alinhamento, considera-se que as seqüências são cadeias de caracteres pertencentes a determinado alfabeto. Para o DNA há o alfabeto $A_{DNA} = \{A, C, T, G\}$ e para proteínas há o alfabeto $A_{Prot} = \{D, E, A, R, N, C, F, G, Q, H, I, L, K, M, P, S, Y, T, W, V\}$.

Definição 3.1 *Seja s uma **seqüência** (ou cadeia) de caracteres pertencentes ao alfabeto A_s :*

- $|s|$ denota o tamanho de uma seqüência ou o número de caracteres que ela possui;
- s_i é o símbolo que ocupa a i -ésima posição;
- $|s| = 0$ representa uma seqüência vazia.

As seqüências que são submetidas ao alinhamento possuem tamanhos variados. Inicialmente coloca-se *gaps*, que são espaços em pontos arbitrários inseridos nas extremidades ou no interior das seqüências, de modo que as seqüências modificadas apresentam o mesmo comprimento. O *gap* é representado pelo caractere "-".

A importância dos *gaps* reside no fato de permitirem que cada caracter ou *gap* em uma seqüência seja comparado a um caracter ou *gap* em todas as outras seqüências. No modelo do alinhamento, cada par caracter-caracter ou caracter-*gap* recebe um peso, que será utilizado para medir a similaridade entre as seqüências.

Entretanto, a inserção de *gaps* não contribui apenas para fazer com que as seqüências resultantes possuam o mesmo tamanho. Um *gap* indica a ocorrência de uma possível mutação, que é a ocorrência de uma operação de inserção ou supressão de monômeros (Isaev 2004). Inserção e supressão de monômeros são conhecidos por afetar a precisão do alinhamento múltiplo, onde cada método de alinhamento escolhe a sua forma de tratar a presença de *gaps* (Golubchik & M. J. Wise 2007).

Definição 3.2 *Sejam duas seqüências $x = x_1 \dots x_m$ e $y = y_1 \dots y_n$, onde $x_1 \dots x_m \in A_s$, com $m > 0$ e $y_1 \dots y_n \in A_s$, com $n > 0$. O alinhamento entre as seqüências x e y é um mapeamento de x e y nas seqüências x' e y' , respectivamente, cujos símbolos pertencem ao alfabeto $A'_s = A_s \cup \{-\}$, tal que:*

- $|x'| = |y'| = t$;
- A remoção de gaps é uma função $rem : (A'_s)^* \rightarrow A_s^*$, tal que:

$$rem(x_1 \dots x_w \dots x_n) = \begin{cases} rem(x_1 \dots x_{w-1} x_{w+1} \dots x_n) & \text{se } w \neq 0 \\ x_1 x_2 \dots x_n & \text{c.c.} \end{cases}$$

onde x_w é o primeiro gap que ocorre na seqüência $x_1 x_2 \dots x_n$.

Isso mostra que para melhorar o alinhamento entre pares de seqüências, os caracteres, além de serem alinhados entre si, também são alinhados com gaps.

A figura 3.1 mostra as seqüências $x : NLEDPSTIE$ e $y : NLDWSTHV$ com alguns possíveis alinhamentos. As cadeias resultantes são sobrepostas, alinhando-se cada caractere de uma seqüência com o caractere da outra seqüência localizado na mesma posição.

| | | | | | | | | | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|--------------------------|-------|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|------------------------|-------|------------------------|
| <p>a)</p> <table style="border-collapse: collapse; margin-left: 20px;"> <tr><td style="text-align: center;">1 2 3 4 5 6 7 8 9</td></tr> <tr><td style="text-align: center;">x: N L E D P S T I E</td></tr> <tr><td style="text-align: center;"> : : :</td></tr> <tr><td style="text-align: center;">y: N L D W S T H V V</td></tr> </table> | 1 2 3 4 5 6 7 8 9 | x: N L E D P S T I E | : : : | y: N L D W S T H V V | <p>b)</p> <table style="border-collapse: collapse; margin-left: 20px;"> <tr><td style="text-align: center;">1 2 3 4 5 6 7 8 9 10</td></tr> <tr><td style="text-align: center;">x: N L E D P S T I E -</td></tr> <tr><td style="text-align: center;"> </td></tr> <tr><td style="text-align: center;">y: N L - D W S T H V V</td></tr> </table> | 1 2 3 4 5 6 7 8 9 10 | x: N L E D P S T I E - | | y: N L - D W S T H V V |
| 1 2 3 4 5 6 7 8 9 | | | | | | | | | |
| x: N L E D P S T I E | | | | | | | | | |
| : : : | | | | | | | | | |
| y: N L D W S T H V V | | | | | | | | | |
| 1 2 3 4 5 6 7 8 9 10 | | | | | | | | | |
| x: N L E D P S T I E - | | | | | | | | | |
| | | | | | | | | | |
| y: N L - D W S T H V V | | | | | | | | | |
| <p>c)</p> <table style="border-collapse: collapse; margin-left: 20px;"> <tr><td style="text-align: center;">1 2 3 4 5 6 7 8 9 10 11</td></tr> <tr><td style="text-align: center;">x: N L E D P S T - - I E</td></tr> <tr><td style="text-align: center;"> :</td></tr> <tr><td style="text-align: center;">y: N L - D W S T H V V -</td></tr> </table> | 1 2 3 4 5 6 7 8 9 10 11 | x: N L E D P S T - - I E | : | y: N L - D W S T H V V - | <p>d)</p> <table style="border-collapse: collapse; margin-left: 20px;"> <tr><td style="text-align: center;">1 2 3 4 5 6 7 8 9 10</td></tr> <tr><td style="text-align: center;">x: N L E D P S T - I E</td></tr> <tr><td style="text-align: center;"> : : :</td></tr> <tr><td style="text-align: center;">y: N L D W S T H V V -</td></tr> </table> | 1 2 3 4 5 6 7 8 9 10 | x: N L E D P S T - I E | : : : | y: N L D W S T H V V - |
| 1 2 3 4 5 6 7 8 9 10 11 | | | | | | | | | |
| x: N L E D P S T - - I E | | | | | | | | | |
| : | | | | | | | | | |
| y: N L - D W S T H V V - | | | | | | | | | |
| 1 2 3 4 5 6 7 8 9 10 | | | | | | | | | |
| x: N L E D P S T - I E | | | | | | | | | |
| : : : | | | | | | | | | |
| y: N L D W S T H V V - | | | | | | | | | |

Figura 3.1: Exemplo de sobreposição de duas seqüências

Quando os caracteres alinhados são iguais, diz-se que houve uma igualdade de caracteres. A situação de alinhamento entre um caractere da primeira cadeia com um gap na segunda, ou vice-versa, é designada de espaçamento. Caracteres diferentes numa mesma posição indicam uma situação de desigualdade de caracteres, neste caso, pode ou não haver aminoácidos semelhantes.

Para medir quão bom é um alinhamento, são pré-determinados os valores para as comparações entre os caracteres. A formalização para comparação entre símbolos de A'_{prot} foi dividida nas definições 3.3 e 3.4.

Definição 3.3 A pontuação atribuída a dois caracteres é dada pela função $p(x_i, y_i) : A_{prot} \times A_{prot} \rightarrow \mathbb{R}$, onde $p(x_i, y_i)$ denota o valor do alinhamento entre o símbolo x_i e o símbolo y_i , $\forall x_i, y_i \in A_{prot}$.

Antes de definir o valor do alinhamento para pares de seqüências, é necessário declarar uma outra função, a qual representa a comparação de regiões de *gaps* com uma região de caracteres de A_{prot} . A definição 3.4 apresenta formalmente a função para penalidade da região de *gaps*.

Definição 3.4 Dado que se pode encontrar na seqüência regiões de *gaps* $\langle g_1, \dots, g_i \rangle$, com $i \geq 1$, gerou-se a função $G(g) : A_{prot} \times - \rightarrow \mathbb{R}$ que denota o alinhamento de um caractere de A_{prot} com um *gap* $-$. A $G(g)$ pode ser definida de múltiplas maneiras, dentre as quais é possível mencionar:

- **Modelo de gap linear:** assume que a inserção de *gaps* na seqüência ocorre de forma independente. Assim, o modelo é formalizado através da função constante:

$$G(g_i) = G(g_{i+1}) = -\rho$$

onde ρ representa a penalidade para cada *gap* encontrado, com $\rho > 0$;

- **Penalidade de abertura de gap:** para representar uma subseqüência contígua de *gaps* criou-se a função:

$$G(g_i) = \begin{cases} -\rho & \text{se } g_{i-1} \neq - \\ -e & \text{c.c.} \end{cases}$$

onde ρ é a penalidade para o primeiro *gap*, de onde vem a denominação para abertura de *gap*, e e representa o peso para a extensão de um *gap*. O surgimento de um novo *gap* é mais comum nas adjacências de outros *gaps*. Dessa forma, $\rho > e$.

Exemplo 1 Comparação entre duas seqüências que contém *gaps*. Para este exemplo adotou-se os seguintes valores: $\rho = 2$ e $e = 0,5$.

A tabela 3.1 apresenta um exemplo de modelo de *gap* linear. Como as colunas com *gap* são comparadas sem considerar possíveis *gaps* adjacentes a esquerda, então todas as colunas que contém *gap* recebem o mesmo valor.

A tabela 3.2 mostra a mesma comparação utilizando modelo de abertura de *gap*. A posição que inicia a inserção (ou com *gap* isolado) recebe o valor de abertura $G = -\rho = -2$, enquanto a coluna que apresenta extensão de região de *gap* tem penalidade menor.

| | | | | | | | | | | | |
|----|---|---|----|---|---|---|---|----|----|----|----|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| x: | N | L | E | D | P | S | T | - | - | - | - |
| | | | | | | | | | | | |
| y: | N | L | - | D | W | S | T | H | V | V | K |
| | | | -2 | | | | | -2 | -2 | -2 | -2 |

Tabela 3.1: Exemplo utilizando modelo de *gap* linear

| | | | | | | | | | | | |
|----|---|---|----|---|---|---|---|----|------|------|------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| x: | N | L | E | D | P | S | T | - | - | - | - |
| | | | | | | | | | | | |
| y: | N | L | - | D | W | S | T | H | V | V | K |
| | | | -2 | | | | | -2 | -0,5 | -0,5 | -0,5 |

Tabela 3.2: Exemplo utilizando modelo de abertura de *gap*

Se o propósito fosse alinhar duas cadeias de tamanho N e não fosse permitida a inclusão de *gaps*, então existiria apenas um alinhamento possível. O fato do alinhamento consistir na existência de *gaps*, permite obter vários alinhamentos entre duas seqüências.

Assim, torna-se importante avaliar o significado do alinhamento. Como foi colocado anteriormente, este trabalho seguirá medindo o alinhamento pela similaridade. Para medir a similaridade é atribuído um valor que corresponde ao mérito de comparações entre caracteres e caracteres com *gap*.

Definição 3.5 O valor de mérito atribuído a comparação dos caracteres x'_i e y'_i da i -ésima coluna do alinhamento, $\forall x'_i, y'_i \in A'_{prot}$, é dado pela função μ , tal que:

$$\mu : A'_{prot} \times A'_{prot} \rightarrow \mathbb{R}$$

$$\mu(x'_i, y'_i) = \begin{cases} G(x'_i) & \text{se } y'_i \neq - \text{ e } x'_i = - \\ G(y'_i) & \text{se } x'_i \neq - \text{ e } y'_i = - \\ p(x'_i, y'_i) & \text{c.c} \end{cases} \quad (3.1)$$

Cada coluna é submetida a função μ de modo que é possível encontrar uma das opções abaixo:

- Caracteres com *gap*: se $(x'_i, -)$ ou $(-, y'_i)$;
- Caracteres diferentes: se $x'_i \neq y'_i$; ou
- Caracteres idênticos: se $x'_i = y'_i$.

A determinação de valores de mérito para as colunas tende a valorizar aminoácidos idênticos ou semelhantes, enquanto penalizam comparações de aminoácidos diferentes e com *gap*. Os valores de mérito podem ser agrupados em sistemas de pontuação. A escolha do sistema de pontuação mais adequado depende do tipo de seqüências que são submetidas ao alinhamento, ficando normalmente a critério do usuário.

Exemplo 2 Um sistema de pontuação simples tem valores de mérito μ_{simples} dados por:

$$\mu_{\text{simples}}(x'_i, y'_i) = \begin{cases} G(x'_i) = -2 & \text{se } y'_i \neq - \text{ e } x'_i = - \\ G(y'_i) = -2 & \text{se } x'_i \neq - \text{ e } y'_i = - \\ p(x'_i, y'_i) = -1 & \text{se } x'_i \neq y'_i \\ p(x'_i, y'_i) = +1 & \text{se } x'_i = y'_i \end{cases}$$

Caso os valores de mérito para comparação entre aminoácidos sejam baseados na freqüência em que são encontrados na natureza, os valores para função $p(x_i, y_i)$ podem ser organizados numa matriz de mérito. A matriz de mérito contém os valores numéricos associados a todos os pares de símbolos comparados.

Definição 3.6 Dados x_i e $y_i \in A_{\text{prot}}$, a pontuação $p(x_i, y_i)$ é representada numa matriz de cardinalidade $|A_{\text{prot}}| \times |A_{\text{prot}}|$ denominada **matriz de mérito**, a qual apresenta as pontuações para todas as comparações de pares de caracteres do alfabeto A_{prot} .

Exemplo 3 Sejam cinco símbolos de A_{prot} e os valores de mérito do exemplo 2, um exemplo de matriz de mérito é apresentado na tabela 3.3. A diagonal representa identidade de caracteres.

| | D | E | A | R | N |
|---|----|----|----|----|----|
| D | +1 | -1 | -1 | -1 | -1 |
| E | -1 | +1 | -1 | -1 | -1 |
| A | -1 | -1 | +1 | -1 | -1 |
| R | -1 | -1 | -1 | +1 | -1 |
| N | -1 | -1 | -1 | -1 | +1 |

Tabela 3.3: Exemplo de matriz de mérito

A medição em determinada coluna no alinhamento é feita independente das demais, isto porque as mutações nas diferentes posições da seqüência ocorrem independentes umas das outras. O valor final do alinhamento corresponde à soma dos méritos atribuídos a todas as colunas, sendo assim calculado pela função de mérito aditiva.

Definição 3.7 O valor do alinhamento é dado pela **função de mérito aditiva**:

$$\text{Alin}(x, y) = \sum_{i=1}^t \mu(x'_i, y'_i) \quad (3.2)$$

O objetivo do alinhamento não envolve apenas examinar o somatório das pontuações, envolve principalmente o alcance do maior número de comparações idênticas e similares possíveis, com o intuito valorizar o alinhamento. Para tal propósito é necessário maximizar a função de mérito aditiva.

Definição 3.8 Dadas duas seqüências x e y um **alinhamento ótimo** é definido como:

$$\text{Alin}(x, y)_{\text{otim}} = \max \sum_{i=1}^t \mu(x'_i, y'_i)$$

Encontrar um alinhamento ótimo envolve dispor as seqüências verificando todos os alinhamentos possíveis e isto requer um custo computacional elevado. A próxima seção abordará alguns modelos baseados em computação convencional para a solução do problema de alinhamento.

3.2 Alinhamento por programação dinâmica

Existem vários modelos capazes de fazer uma aproximação da similaridade entre duas seqüências. Para encontrar o alinhamento ótimo é preciso gerar todas as possibilidades e selecionar uma combinação que seja considerada a melhor. Contudo, o número de alinhamentos é exponencial ao número de seqüências e a abordagem resulta em um algoritmo lento (Setubal & Meidanis 1997).

Para resolver esse problema pode-se utilizar a técnica conhecida como programação dinâmica, cujos algoritmos solucionam os problemas de otimização em que há muitas soluções possíveis, porém apenas soluções ótimas são desejáveis.

Essa técnica de resolução recebe instâncias de um problema e divide em subproblemas. Cada subproblema só pode ser solucionado quando, partindo-se do início, as instâncias precedentes já foram resolvidas. Dessa forma, a cada subproblema resolvido o resultado obtido é armazenado em uma tabela junto com a pontuação e, quando todos estão resolvidos, seleciona-se a seqüência de resultados que possui a maior pontuação.

Esse método foi primeiramente descrito nos anos 50 por Richard Bellman da Princeton University como uma técnica de otimização geral, cujo objetivo é obter todas as máximas pontuações do subproblema (Gibas & Jambeck 2001).

A programação dinâmica foi introduzida na comparação de seqüências biológicas por Saul Needleman e Christian Wunsch e o estudo deles ficou conhecido como Algoritmo Needleman-Wunsch (Needleman & Wunsch 1970), o qual sempre encontra todas os alinhamentos ótimos globais. Originalmente, o algoritmo Needleman-Wunsch assume o modelo de *gap* linear, para o qual $\mu(-, a) = \mu(a, -) = G(-) = -\rho, \forall a \in A_{prot}$, com $\rho > 0$.

Definição 3.9 *Dadas duas seqüências $x = x_1x_2 \dots x_i \dots x_m$ e $y = y_1y_2 \dots y_j \dots y_n$, tal que m e n são os tamanhos das seqüências. Então é construída a matriz M de ordem $(m + 1) \times (n + 1)$, onde o valor $M(i, j)$ é o valor do alinhamento ótimo entre $x = x_1 \dots x_i$ e $y = y_1 \dots y_j$.*

$$M(i, j) = \max \begin{cases} M(i - 1, j - 1) + p(x_i, y_j) \\ M(i - 1, j) + G(y_i) \\ M(i, j - 1) + G(x_i) \end{cases} \quad (3.3)$$

O algoritmo inicia pelo preenchimento da primeira coluna da matriz - $M(i, 0)$ para $i = 1, \dots, m$, a qual representa valores de mérito do alinhamento de $x_1 \dots x_i$ com uma região de *gap* de tamanho i . Analogamente, a primeira linha da matriz - $M(0, j)$ para $j = 1, \dots, n$, são os valores de mérito do alinhamento de $y_1 \dots y_j$ com uma região de *gap* de tamanho j .

Os outros elementos da matriz são atribuídos recursivamente, inicializando $M(0, 0) = 0$ e partindo-se do canto superior direito para o canto inferior esquerdo, calcula-se $M(i, j)$ através da equação 3.3. O preenchimento das células da matriz pode ser feito de uma das seguintes formas:

- A primeira opção é seguir a diagonal alinhando o aminoácido x_i com o y_j , adicionando o valor do alinhamento com o valor anterior, $M(i - 1, j - 1)$;
- A segunda opção é seguir a vertical alinhando o aminoácido x_i com um *gap*, adicionando o valor anterior, $M(i - 1, j)$, a penalidade p ;
- A terceira opção é caso o valor máximo seja seguir pela horizontal alinhando o aminoácido y_i com um *gap* e adicionando o valor anterior, $M(i, j - 1)$, a penalidade p .

A seguir é mostrado um algoritmo que preenche a matriz $M_{m+1, n+1}$ utilizando a técnica de programação dinâmica, tal que o valor da célula $M[m, n]$ é o *score*¹ para o alinhamento ótimo entre x e y .

¹O valor do alinhamento é chamado de *score*

Algorithm 1 Algoritmo de preenchimento da matriz de programação dinâmica

```

1: Matriz( $x, y$ )
2: for  $i = 0$  to  $m$  do
3:    $M[i, 0] \leftarrow i * G(y_i)$ 
4: end for
5: for  $j = 0$  to  $n$  do
6:    $M[0, j] \leftarrow j * G(x_j)$ 
7: end for
8: for  $i = 0$  to  $m$  do
9:   for  $j = 0$  to  $n$  do
10:     $M[i, j] \leftarrow \max (M[i - 1, j - 1] + p(x_i, y_j), M[i - 1, j] + G(y_j), M[i, j - 1] + G(x_j) )$ 
11:   end for
12: end for
13: return  $M[m, n]$ 

```

A tabela 3.4 mostra o exemplo de matriz de programação dinâmica para as seqüências $x = NLVNSEHRM$ e $y = NLYVPSEMI$ usando o sistema de pontuação simples do exemplo 2.

| F | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|-----|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| | | N | L | Y | V | P | S | E | M | I |
| 0 | 0 | -2 | -4 | -6 | -8 | -10 | -12 | -14 | -16 | -18 |
| 1 | N | -2 | +1 | -1 | -3 | -5 | -7 | -9 | -11 | -13 |
| 2 | L | -4 | -1 | +2 | 0 | -2 | -4 | -6 | -8 | -10 |
| 3 | V | -6 | -3 | 0 | +1 | +1 | -1 | -3 | -5 | -7 |
| 4 | N | -8 | -5 | -2 | -1 | 0 | 0 | -2 | -4 | -6 |
| 5 | S | -10 | -7 | -4 | -3 | -2 | -1 | +1 | -1 | -3 |
| 6 | E | -12 | -9 | -6 | -5 | -4 | -3 | -1 | +2 | 0 |
| 7 | H | -14 | -11 | -8 | -7 | -6 | -5 | -3 | 0 | +1 |
| 8 | R | -16 | -13 | -10 | -9 | -8 | -7 | -5 | -2 | -1 |
| 9 | M | -18 | -15 | -12 | -11 | -10 | -9 | -7 | -4 | -1 |

Tabela 3.4: Matriz de programação dinâmica

Na tabela 3.4 foram construídas arestas partindo de $M[m, n]$ com a finalidade de indicar de onde partiu a pontuação máxima até chegar em $M[0, 0]$. Observa-se que pode haver mais de um caminho, o que indica mais de um alinhamento ótimo entre duas seqüências.

Para obter o alinhamento ótimo a partir da tabela, não é necessário construir as arestas explicitamente. Em Setubal & Meidanis (1997) pode-se encontrar um algoritmo que realiza a busca na matriz de programação dinâmica.

Dados x , y e a matriz $M_{m+1, n+1}$, o algoritmo que obtém as seqüências x' e y' , alinhadas respectivamente nos vetores V_x e V_y , pode ser representado pelo pseudo-código a seguir.

Algorithm 2 Algoritmo de busca pelo melhor alinhamento

```

1:  $Alin(m, n, t)$ 
2: if  $i = 0$  and  $j = 0$  then
3:    $t \leftarrow 0$ 
4: else
5:   if  $(i > 0)$  and  $(M[i, j] = M[i - 1, j] + G(y_i))$  then
6:      $Alin(i - 1, j, t)$ 
7:      $t \leftarrow t + 1$ 
8:      $V_x[t] \leftarrow x_i$ 
9:      $V_y[t] \leftarrow -$ 
10:  else
11:    if  $(i > 0)$  and  $(j > 0)$  and  $(M[i, j] = M[i - 1, j - 1] + p(x_i, y_j))$  then
12:       $Alin(i - 1, j - 1, t)$ 
13:       $t \leftarrow t + 1$ 
14:       $V_x[t] \leftarrow x_i$ 
15:       $V_y[t] \leftarrow y_i$ 
16:    else
17:      if  $(j > 0)$  and  $(M[i, j] = M[i, j - 1] + G(x_i))$  then
18:         $Alin(i, j - 1, t)$ 
19:         $t \leftarrow t + 1$ 
20:         $V_x[t] \leftarrow -$ 
21:         $V_y[t] \leftarrow y_i$ 
22:      end if
23:    end if
24:  end if
25: end if

```

Exemplo 4 Seja a matriz M da figura 3.4 submetida ao algoritmo para achar o(s) alinhamento(s) ótimo(s) entre $x = NLVNSEHRM$ e $y = NLYVPSEMI$, resulta em três possibilidades para V_x e V_y , respectivamente (tabela 3.5).

| | | | | | | | | | | |
|---------|-------------------------------------|----|----|----|----|----|----|----|----|----|
| t | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| V_x : | N | L | - | V | N | S | E | H | R | M |
| V_y : | N | L | Y | V | P | S | E | M | I | - |
| μ | +1 | +1 | -2 | +1 | -1 | +1 | +1 | -1 | -1 | -2 |
| score | $\sum_{i=1}^t \mu(x'_i, y'_i) = -2$ | | | | | | | | | |
| t | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| V_x : | N | L | - | V | N | S | E | H | R | M |
| V_y : | N | L | Y | V | P | S | E | M | - | I |
| μ | +1 | +1 | -2 | +1 | -1 | +1 | +1 | -1 | -2 | -1 |
| score | $\sum_{i=1}^t \mu(x'_i, y'_i) = -2$ | | | | | | | | | |
| t | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| V_x : | N | L | - | V | N | S | E | H | R | M |
| V_y : | N | L | Y | V | P | S | E | - | M | I |
| μ | +1 | +1 | -2 | +1 | -1 | +1 | +1 | -2 | -1 | -1 |
| score | $\sum_{i=1}^t \mu(x'_i, y'_i) = -2$ | | | | | | | | | |

Tabela 3.5: Exemplo de alinhamento com programação Dinâmica

3.3 Matrizes de substituição

O sistema de pontuação utilizado durante o alinhamento pode ter alguma relevância biológica durante a avaliação da similaridade entre as seqüências. Ao se comparar proteínas a utilização do sistema de pontuação simples (exemplo 2) não é suficiente, pois é de conhecimento que certos aminoácidos possuem propriedades físico-químicas semelhantes. Como exemplo, pode-se citar a substituição da isoleucina por valina (que são pequenos e hidrofóbicos) e a serina por treonina (os quais pertencem ao grupo de aminoácidos polares).

Dois aminoácidos são considerados similares se um puder ser substituído pelo outro com uma margem de vantagem positiva dentro de uma matriz de substituição (Gibas & Jambeck 2001).

Dessa forma, as matrizes de mérito (definição 3.6) podem ser representadas pelas chamadas matrizes de substituição. Matrizes de substituição armazenam valores que indicam a probabilidade de ocorrer um par de resíduos distintos em uma coluna qualquer ao se examinar um alinhamento de seqüências. Estas matrizes possuem ordem 20×20 .

A utilização de matrizes de substituição permite discernir se o alinhamento é aleatório ou significativo, visto que as probabilidades são retiradas a partir da observação direta das freqüências de substituição dos diversos pares de resíduos entre proteínas relacionadas. As matrizes de substituição mais populares para alinhar aminoácidos são as matrizes PAM e BLOSUM.

3.3.1 PAM

Matrizes PAM (point accepted mutation) foram criadas em 1978 por Margaret Dayhoff em conjunto com outros pesquisadores que estabeleceram um modelo das regras pelas quais mudanças de aminoácidos acontecem em proteínas no decorrer da evolução, catalogando várias proteínas existente em determinados grupos taxonômicos e comparando as seqüências fortemente relacionadas entre muitas famílias.

Considerou-se a questão na qual é possível observar substituições de aminoácidos específicos quando duas seqüências de proteína homólogas estão alinhadas.

Para determinar todas as possíveis trocas de resíduos, os pesquisadores examinaram 1572 substituições em 71 grupos de proteínas fortemente relacionadas, ou seja, determinaram mutações aceitáveis a partir de mudanças de aminoácido empiricamente observadas. A tabela 3.6 mostra a freqüência que pares de aminoácidos foram alinhados.

| O | A | R | N | D | C | Q | E | G | H | I | L | K | M | F | P | S | T | W | Y | V |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| O | Ala | Arg | Asn | Asp | Cys | Gln | Glu | Gly | His | Ile | Leu | Lys | Met | Phe | Pro | Ser | Thr | Tip | Tyr | Val |
| A | | | | | | | | | | | | | | | | | | | | |
| R | 30 | | | | | | | | | | | | | | | | | | | |
| N | 109 | 17 | | | | | | | | | | | | | | | | | | |
| D | 154 | 0 | 532 | | | | | | | | | | | | | | | | | |
| C | 33 | 10 | 0 | 0 | | | | | | | | | | | | | | | | |
| Q | 93 | 120 | 50 | 76 | 0 | | | | | | | | | | | | | | | |
| E | 266 | 0 | 94 | 831 | 0 | 422 | | | | | | | | | | | | | | |
| G | 579 | 10 | 156 | 162 | 10 | 30 | 112 | | | | | | | | | | | | | |
| H | 21 | 103 | 226 | 43 | 10 | 243 | 23 | 10 | | | | | | | | | | | | |
| I | 66 | 30 | 36 | 13 | 17 | 8 | 35 | 0 | 3 | | | | | | | | | | | |
| L | 95 | 17 | 37 | 0 | 0 | 75 | 15 | 17 | 40 | 253 | | | | | | | | | | |
| K | 57 | 477 | 322 | 85 | 0 | 147 | 104 | 60 | 23 | 43 | 39 | | | | | | | | | |
| M | 29 | 17 | 0 | 0 | 0 | 20 | 7 | 7 | 0 | 57 | 207 | 90 | | | | | | | | |
| F | 20 | 7 | 7 | 0 | 0 | 0 | 0 | 17 | 20 | 90 | 167 | 0 | 17 | | | | | | | |
| P | 345 | 67 | 27 | 10 | 10 | 93 | 40 | 49 | 50 | 7 | 43 | 43 | 4 | 7 | | | | | | |
| S | 772 | 137 | 432 | 98 | 117 | 47 | 86 | 450 | 26 | 20 | 32 | 168 | 20 | 40 | 269 | | | | | |
| T | 590 | 20 | 169 | 57 | 10 | 37 | 31 | 50 | 14 | 129 | 52 | 200 | 28 | 10 | 73 | 696 | | | | |
| W | 0 | 27 | 3 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 13 | 0 | 0 | 10 | 0 | 17 | 0 | | | |
| Y | 20 | 3 | 36 | 0 | 30 | 0 | 10 | 0 | 40 | 13 | 23 | 10 | 0 | 260 | 0 | 22 | 23 | 6 | | |
| V | 365 | 20 | 13 | 17 | 33 | 27 | 37 | 97 | 30 | 661 | 303 | 17 | 77 | 10 | 50 | 43 | 186 | 0 | 17 | |
| O | A | R | N | D | C | Q | E | G | H | I | L | K | M | F | P | S | T | W | Y | V |
| O | Ala | Arg | Asn | Asp | Cys | Gln | Glu | Gly | His | Ile | Leu | Lys | Met | Phe | Pro | Ser | Thr | Tip | Tyr | Val |

Table 3.6: Pontos de mutações aceitáveis

Dessa forma, verificou-se que a substituição de um aminoácido por outro em uma proteína é aceita pela seleção natural. Uma mudança de aminoácido que é aceita por seleção natural acontece quando:

- um gene sofre uma mutação no DNA de modo a codificar um aminoácido diferente;
- algumas espécies adotam aquela mudança como a forma predominante da proteína.

Outros pesquisadores propuseram atualizações nos dados publicados em 1978, porém os resultados de Dayhoff são admissíveis e possibilitam mostrar os aminoácidos que são normalmente trocados. Alguns aminoácidos são menos mutáveis que outros, isso se deve a importância de regras estruturais e funcionais em proteínas, tal que a consequência de substituir um dado aminoácido por outro é prejudicial ao organismo (Pevsner 2003).

Alguns aminoácidos semelhantes como asparagina e serina sofrem substituições muito freqüentes, enquanto outros resíduos raramente sofrem mutação, como é o caso do tripofano e da cisteína.

3.3.1.1 Matrix PAM1

Com os dados da tabela 3.6 e a probabilidade de ocorrência de cada aminoácido, Dayhoff e seus companheiros de equipe geraram uma matriz de probabilidade de mutação (tabela 3.7). Cada célula da matriz indica a probabilidade de um aminoácido j ser substituído por outro aminoácido i num intervalo evolutivo específico.

Então, foi definido o intervalo de 1(um) PAM a unidade de divergência evolucionária na qual 1% dos aminoácidos são trocados entre duas seqüências protéicas. O intervalo evolucionário desta Matriz PAM é definido em termos de porcentagem dos aminoácidos divergentes e não em unidade tempo.

Examinando a tabela 3.7, algumas características são observáveis. O *score* mais alto se localiza na diagonal, justamente porque ela representa a possibilidade de cada um dos aminoácidos permanecer inalterado, ou seja, certo aminoácido ser substituído por ele mesmo sobre uma distância evolucionária de um PAM.

A soma de cada coluna é igual a 10.000, o que equivale, por exemplo, a probabilidade de uma alanina ser substituída por cada um dos 20 aminoácidos. Dessa forma, este somatório de colunas equivale a 100%. Assim, a probabilidade de uma alanina permanecer intacta é 98,67% e a probabilidade de ser substituída por uma valina é de 0,13%.

| O | A | R | N | D | C | Q | E | G | H | I | L | K | M | F | P | S | T | W | Y | V |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| Ala | Arg | Asn | Asp | Cys | Gln | Glu | Gly | His | Ile | Leu | Lys | Met | Phe | Pro | Ser | Thr | Trp | Tyr | Val | |
| 9867 | 2 | 9 | 10 | 3 | 8 | 17 | 21 | 2 | 6 | 4 | 2 | 6 | 2 | 22 | 35 | 32 | 0 | 2 | 2 | 18 |
| 1 | 9913 | 1 | 0 | 1 | 10 | 0 | 0 | 10 | 3 | 1 | 19 | 4 | 1 | 4 | 6 | 1 | 8 | 0 | 0 | 1 |
| 4 | 1 | 9822 | 36 | 0 | 4 | 6 | 6 | 21 | 3 | 1 | 13 | 0 | 1 | 2 | 20 | 9 | 1 | 4 | 1 | 1 |
| 6 | 0 | 42 | 9859 | 0 | 6 | 53 | 6 | 4 | 1 | 0 | 3 | 0 | 0 | 1 | 5 | 3 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 9973 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 5 | 1 | 0 | 3 | 2 | 2 |
| 3 | 9 | 4 | 5 | 0 | 9876 | 27 | 1 | 23 | 1 | 3 | 6 | 4 | 0 | 6 | 2 | 2 | 0 | 0 | 0 | 1 |
| 10 | 0 | 7 | 56 | 0 | 35 | 9865 | 4 | 2 | 3 | 1 | 4 | 1 | 0 | 3 | 4 | 2 | 0 | 1 | 2 | 2 |
| 21 | 1 | 12 | 11 | 1 | 3 | 7 | 9935 | 1 | 0 | 1 | 2 | 1 | 1 | 3 | 21 | 3 | 0 | 0 | 5 | 5 |
| 1 | 8 | 18 | 3 | 1 | 20 | 1 | 0 | 9912 | 0 | 1 | 1 | 0 | 2 | 3 | 1 | 1 | 1 | 4 | 1 | 1 |
| 2 | 2 | 3 | 1 | 2 | 1 | 2 | 0 | 0 | 9872 | 9 | 2 | 12 | 7 | 0 | 1 | 7 | 0 | 1 | 33 | 33 |
| 3 | 1 | 3 | 0 | 0 | 6 | 1 | 1 | 4 | 22 | 9947 | 2 | 45 | 13 | 3 | 1 | 3 | 4 | 2 | 15 | 15 |
| 2 | 37 | 25 | 6 | 0 | 12 | 7 | 2 | 2 | 4 | 1 | 9926 | 20 | 0 | 3 | 8 | 11 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 5 | 8 | 4 | 9874 | 1 | 0 | 1 | 2 | 0 | 0 | 4 | 4 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 2 | 8 | 6 | 0 | 4 | 9946 | 0 | 2 | 1 | 3 | 28 | 0 | 0 |
| 13 | 5 | 2 | 1 | 1 | 8 | 3 | 2 | 5 | 1 | 2 | 2 | 1 | 1 | 9926 | 12 | 4 | 0 | 0 | 2 | 2 |
| 28 | 11 | 34 | 7 | 11 | 4 | 6 | 16 | 2 | 2 | 1 | 7 | 4 | 3 | 17 | 9840 | 38 | 5 | 2 | 2 | 2 |
| 22 | 2 | 13 | 4 | 1 | 3 | 2 | 2 | 1 | 11 | 2 | 8 | 6 | 1 | 5 | 32 | 9871 | 0 | 2 | 9 | 9 |
| 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 9976 | 1 | 0 | 0 |
| 1 | 0 | 3 | 0 | 3 | 0 | 1 | 0 | 4 | 1 | 1 | 0 | 0 | 21 | 0 | 1 | 1 | 2 | 9945 | 1 | 1 |
| 13 | 2 | 1 | 1 | 3 | 2 | 2 | 3 | 3 | 57 | 11 | 1 | 17 | 1 | 3 | 2 | 10 | 0 | 2 | 9901 | 9901 |

Table 3.7: Matriz PAM1

Outra informação que se pode extrair dessa tabela, são as características de mudança no curso evolucionário. O aminoácido que mais sofre mutação tem o mais baixo valor na diagonal principal, ou seja, a asparagina tem probabilidade de 98.22% de permanecer inalterada. Enquanto o aminoácido menos mutável, triptofano, tem 99,76% de chance de não ser alterado.

Para cada aminoácido original (topo superior da matriz) facilmente observa-se por quais aminoácidos comumente são substituídos caso uma mudança aconteça. Estas informações são úteis para alinhamento par a par porque elas formarão a base de um sistema de pontuação na qual razoáveis substituições de aminoácidos em um alinhamento são recompensadas, enquanto substituições improváveis são penalizadas (Mount 2001). Estes conceitos também são relevantes em pesquisas em banco de dados de seqüências.

Exemplo 5 Dados $x = NLVNSEHRM$ e $y = NLYVPSEMI$, um possível alinhamento entre estas seqüências utilizando a matriz de substituição PAM1 é mostrado na tabela 3.8.

| | | | | | | | | | | |
|--------------|----------------------------------------|------|----|------|---|------|------|---|---|----|
| $V_x:$ | N | L | - | V | N | S | E | H | R | M |
| $V_y:$ | N | L | Y | V | P | S | E | M | I | - |
| μ | 9822 | 9947 | -2 | 9901 | 2 | 9840 | 9865 | 0 | 2 | -2 |
| <i>score</i> | $\sum_{i=1}^t \mu(x'_i, y'_i) = 49375$ | | | | | | | | | |

Tabela 3.8: Exemplo de pontuação com PAM1

3.3.2 BLOSUM

Além da matriz PAM, outra alternativa muito comum para matriz de mérito é a série de matrizes de substituição de blocos (*blocks substitutions matrix - BLOSUM*).

A Matriz BLOSUM foi primeiramente descrita com base no banco de dados *BLOCKS*² (Henikoff & Henikoff 1991). O banco de dados *BLOCKS* compreende segmentos de alinhamentos múltiplos (seção 3.4) sem a presença de *gaps*, onde tais segmentos representam regiões muito conservadas de famílias de proteínas.

As matrizes BLOSUM são derivadas de *BLOCKS* cujos alinhamento correspondem a matriz BLOSUM- X , tal que X representa a percentagem mínima de identidade no alinhamento de proteínas. Como exemplo pode-se destacar a matriz BLOSUM-62 apresentada na tabela 3.9.

²Informações sobre o banco de dados *BLOCKS* estão em: <http://blocks.fhrc.org/>.

| | A | R | N | D | C | Q | E | G | H | I | L | K | M | F | P | S | T | W | Y | V |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0 | Ala | Arg | Asn | Asp | Cys | Gln | Glu | Gly | His | Ile | Leu | Lys | Met | Phe | Pro | Ser | Thr | Trp | Tyr | Val |
| A | 4 | -1 | -2 | -2 | 0 | -1 | -1 | 0 | -2 | -1 | -1 | -1 | -1 | -2 | -1 | 1 | 0 | -3 | -2 | 0 |
| R | -1 | 5 | 0 | -2 | -3 | 1 | 0 | -2 | 0 | -3 | -2 | 2 | -1 | -3 | -2 | -1 | -1 | -3 | -2 | -3 |
| N | -2 | 0 | 6 | 1 | -3 | 0 | 0 | 0 | 1 | -3 | -3 | 0 | -2 | -3 | -2 | 1 | 0 | -4 | -2 | -3 |
| D | -2 | -2 | 1 | 6 | -3 | 0 | 2 | -1 | -1 | -3 | -4 | -1 | -3 | -3 | -1 | 0 | -1 | -4 | -3 | -3 |
| C | 0 | -3 | -3 | -3 | 9 | -3 | -4 | -3 | -3 | -1 | -1 | -3 | -1 | -2 | -3 | -1 | -1 | -2 | -2 | -1 |
| Q | -1 | 1 | 0 | 0 | -3 | 5 | 2 | -2 | 0 | -3 | -2 | 1 | 0 | -3 | -1 | 0 | -1 | -2 | -1 | -2 |
| E | -1 | 0 | 0 | 2 | -4 | 2 | 5 | -2 | 0 | -3 | -3 | 1 | -2 | -3 | -1 | 0 | -1 | -3 | -2 | -2 |
| G | 0 | -2 | 0 | -1 | -3 | -2 | -2 | 6 | -2 | -4 | -4 | -2 | -3 | -3 | -2 | 0 | -2 | -2 | -3 | -3 |
| H | -2 | 0 | 1 | -1 | -3 | 0 | 0 | -2 | 8 | -3 | -3 | -1 | -2 | -1 | -2 | -1 | -2 | -2 | 2 | -3 |
| I | -1 | -3 | -3 | -3 | -1 | -3 | -3 | -4 | -3 | 4 | 2 | -3 | 1 | 0 | -3 | -2 | -1 | -3 | -1 | 3 |
| L | -1 | -2 | -3 | -4 | -1 | -2 | -3 | -4 | -3 | 2 | 4 | -2 | 2 | 0 | -3 | -2 | -1 | -2 | -1 | 1 |
| K | -1 | 2 | 0 | -1 | -3 | 1 | 1 | -2 | -1 | -3 | -2 | 5 | -1 | -3 | -1 | 0 | -1 | -3 | -2 | -2 |
| M | -1 | -1 | -2 | -3 | -1 | 0 | -2 | -3 | -2 | 1 | 2 | -1 | 5 | 0 | -2 | -1 | -1 | -1 | -1 | 1 |
| F | -2 | -3 | -3 | -3 | -2 | -3 | -3 | -3 | -1 | 0 | 0 | -3 | 0 | 6 | -4 | -2 | -2 | 1 | 3 | -1 |
| P | -1 | -2 | -2 | -1 | -3 | -1 | -1 | -2 | -2 | -3 | -3 | -1 | -2 | -4 | 7 | -1 | -1 | -4 | -3 | -2 |
| S | 1 | -1 | 1 | 0 | -1 | 0 | 0 | 0 | -1 | -2 | -2 | 0 | -1 | -2 | -1 | 4 | 1 | -3 | -2 | -2 |
| T | 0 | -1 | 0 | -1 | -1 | -1 | -1 | -2 | -2 | -1 | -1 | -1 | -1 | -2 | -1 | 1 | 5 | -2 | -2 | 0 |
| W | -3 | -3 | -4 | -4 | -2 | -2 | -3 | -2 | -2 | -3 | -2 | -3 | -1 | 1 | -4 | -3 | -2 | 11 | 2 | -3 |
| Y | -2 | -2 | -2 | -3 | -2 | -1 | -2 | -3 | 2 | -1 | -1 | -2 | -1 | 3 | -3 | -2 | -2 | 2 | 7 | -1 |
| V | 0 | -3 | -3 | -3 | -1 | -2 | -2 | -3 | -3 | 3 | 1 | -2 | 1 | -1 | -2 | -2 | 0 | -3 | -1 | 4 |

Table 3.9: Matriz Blosum62

Os valores internos da matriz, identidade ou substituição de aminoácidos, são pontuações baseadas na freqüências que os resíduos são observados dentro do alinhamento de proteínas relacionadas. As substituições mais prováveis recebem valores positivos e as menos prováveis recebem valores negativos.

A matriz BLOSUM62 (tabela 3.9) é uma opção para matriz de mérito, ela foi produzida a partir de BLOCKS contendo mais que 62% de aminoácidos idênticos. Esta é a matriz padrão que o BLAST utiliza para pontuar alinhamentos.

Exemplo 6 Dados $x = NLVNSEHRM$ e $y = NLYVPSEMI$, um possível alinhamento entre estas seqüências utilizando a matriz de substituição BLOSUM62 é mostrado na tabela 3.10.

| | | | | | | | | | | |
|--------|-------------------------------------|---|----|---|----|---|---|----|----|----|
| $V_x:$ | N | L | - | V | N | S | E | H | R | M |
| $V_y:$ | N | L | Y | V | P | S | E | M | I | - |
| μ | 6 | 4 | -2 | 4 | -2 | 4 | 5 | -2 | -3 | -2 |
| score | $\sum_{i=1}^t \mu(x'_i, y'_i) = 12$ | | | | | | | | | |

Tabela 3.10: Exemplo de pontuação com BLOSUM62

Todas as matrizes BLOSUM são baseadas em alinhamentos observados, o que evidência a importância do grau de relacionamento entre as seqüências alinhadas. Matrizes BLOSUM com numeração alta são projetadas para alinhar seqüências fortemente relacionadas. Enquanto as matrizes com números mais baixos são mais indicadas para alinhar seqüências não tão relacionadas (Zomaya, 2006).

Dessa forma, a matriz BLOSUM80 é utilizada para comparar alinhamentos mais semelhantes, ou seja, que pouco divergiram. Já a matriz BLOSUM45 é utilizada para alinhamentos mais divergentes.

3.4 Alinhamento múltiplo

O alinhamento de várias proteínas concomitantemente é uma técnica muito utilizada em análise de seqüências. Dentre as tarefas que o alinhamento múltiplo de proteínas auxilia pode-se destacar: prever a estrutura secundária ou terciária, encontrar homologia entre as seqüências, identificar regiões conservadas em proteínas da mesma família e reconstrução filogenética.

O alinhamento múltiplo é definido de maneira análoga ao alinhamento de pares de seqüências, são embutidos *gaps* em cada seqüência de modo tal que elas tenham o mesmo comprimento. Para definir a pontuação do alinhamento é preciso calcular a pontuação de cada coluna. A definição 3.10 formaliza a determinação do *score* do alinhamento múltiplo pela função de mérito aditiva.

Definição 3.10 *Seja um alinhamento múltiplo $Mult$ entre as seqüências s_1, s_2, \dots, s_n . A i -ésima coluna de $Mult$ é chamada de $Mult_i$. Então, o score do alinhamento múltiplo é dado por:*

$$score(Mult) = G_{pen}(g) + \sum_i p_{mult}(Mult_i) \quad (3.4)$$

onde:

- $G_{pen}(g)$ é a função para penalizar gap;
- $p_{mult}(Mult_i)$ é a pontuação atribuída à comparação dos elementos que não são gaps na i -ésima coluna de $Mult$.

O método da soma dos pares (sum of pairs - SP) (Sperschneider 2008) é uma das formas de calcular a pontuação das colunas do alinhamento múltiplo.

Definição 3.11 *Sejam os elementos a_i^k e a_i^l , denotando respectivamente o k -ésimo e o l -ésimo aminoácido localizados na coluna de $Mult_i$. Soma dos pares é o método para calcular a função p_{mult} , tal que:*

$$p_{mult}(Mult_i) = SP = \sum_{k < l} p(a_i^k, a_i^l) \quad (3.5)$$

onde os elementos $a_i^k \neq -$ e $a_i^l \neq -$.

Utilizando a definição 3.10 como método para pontuação de uma alinhamento, seria possível generalizar o algoritmo de programação dinâmica para calcular o alinhamento ótimo de n seqüências, com $n \geq 3$.

Entretanto, ao alinhar n seqüências $s_1 = x_1^1 \dots x_{t_1}^1$, $s_2 = x_1^2 \dots x_{t_2}^2$, \dots , $s_n = x_1^n \dots x_{t_n}^n$, tal que $t_1, t_2, \dots, t_{n-1}, t_n$ representam os tamanhos das n seqüências de comprimentos variados, a matriz gerada na definição 3.9 tornar-se-ia n -dimensional ao invés de bi-dimensional e o alinhamento ótimo seria um caminho interno da célula M_{t_1, t_2, \dots, t_n} até $M_{0, \dots, 0}$.

O algoritmo de programação dinâmica para alinhamento de pares de seqüências, devido a sua complexidade $O(n^2)$, acaba sendo computacionalmente oneroso com muitas seqüências. Assim, é necessário investir em métodos de otimização e novas heurísticas, pois é inviável tentar encontrar a solução ótima. A seguir, são descritas algumas possibilidades para alinhar múltiplas seqüências.

3.4.1 Alinhamento progressivo

O método de alinhamento progressivo produz o alinhamento múltiplo a partir de vários alinhamentos par a par. Seu funcionamento básico envolve a seleção de duas das n seqüências, que são submetidas a algum algoritmo de alinhamento par a par como, por exemplo, o algoritmo Needleman & Wunsch (1970) apresentado na seção 3.2. O alinhamento resultante é então fixado e, em seguida, uma terceira seqüência é selecionada e alinhada com as duas primeiras. O processo se repete para todas as seqüências não alinhadas até que todas sejam utilizadas.

Algoritmos de alinhamento progressivo podem ser diferentes nos seguintes aspectos:

- A forma que se escolhe a ordem das seqüências para fazer o alinhamento;
- Se a progressão envolve somente o incremento de novas seqüências ao alinhamento ou se subfamílias são moldadas em uma estrutura de árvore;
- O modo que o alinhamento progride e se há atribuição de pesos diferentes para cada uma das seqüências.

A maioria dos algoritmos de alinhamento progressivos usam alguma heurística. Um ponto positivo é que geralmente as heurísticas não separam o processo de escolher o sistema de pontuação, da parte que envolve o algoritmo de alinhamento. Outra vantagem é que a execução é relativamente rápida e, em muitos casos, o alinhamento múltiplo resultante é aceitável.

A heurística mais utilizada cria um pré-alinhamento dos pares de seqüências mais semelhantes, os quais são organizados em uma árvore guia que representa com eficiência este princípio.

Segundo Isaev (2004), alguns métodos desse tipo são heurísticos por natureza, pois usam as relações evolutivas entre as seqüências selecionadas, modeladas em uma árvore evolutiva denominada árvore filogenética³, onde as seqüências de entrada são representadas pelas ramificações mais externas, as folhas da árvore, e convergem para um nó interno comum (figura 3.2).

É importante ressaltar que cada nó interno da árvore filogenética é fruto de uma análise de ancestralidade que representa uma incerteza da localização de possíveis aminoácidos que foram substituídos ou possíveis inserções de *gaps*.

O desafio desse método de alinhamento múltiplo está em utilizar a combinação apropriada de seqüências, matriz de substituição e penalidades de

³Representação em forma de árvore das relações evolutivas entre várias espécies de seres vivos que possuem um antepassado comum.

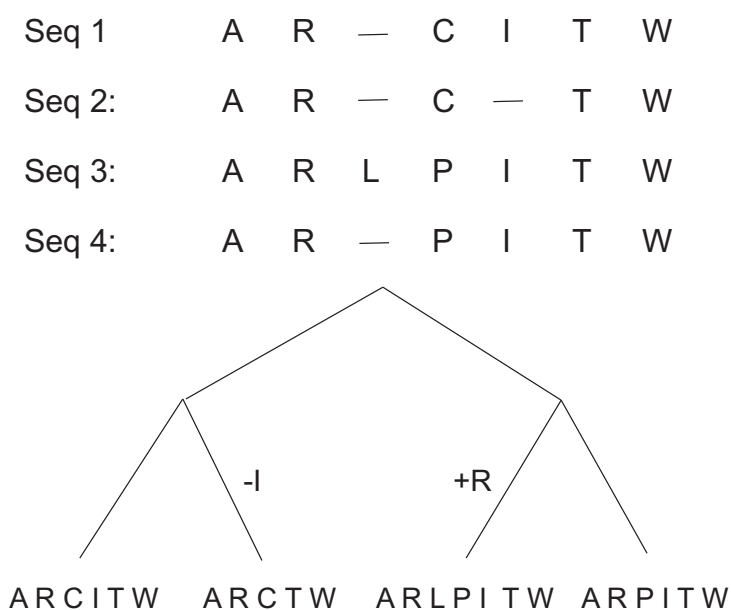


Figura 3.2: Exemplo de árvore filogenética

gaps, de forma que as séries corretas de mudanças evolutivas possam ser encontradas.

3.4.1.1 Heurística para alinhamento progressivo

Uma heurística de alinhamento progressivo foi proposta por Feng & Doolittle (1987). O algoritmo começa com a formação de uma matriz diagonal, para representar os *scores* dos alinhamentos de todos os pares de seqüências possíveis. A partir da análise dos alinhamentos contidos na matriz é mapeada uma árvore pelo algoritmo de Fitch & Margoliash (1967), que faz um agrupamento rápido para construir árvores evolutivas com base na matriz de distâncias.

A partir do primeiro nó adicionado à árvore, alinha-se os nós filhos que podem ser dois alinhamentos, duas seqüências ou uma seqüência e um alinhamento. Os demais nós são analisados seguindo a ordem do par mais semelhante para o menos semelhante, informação que pode ser obtida da árvore citada no parágrafo anterior. O processo se repete até que todas as seqüências estejam alinhadas.

A acurácia imposta pela matriz de distância não é tão eficiente, mas como sua relevância diz respeito apenas a construção da árvore, a falta de exatidão em si não causa grande impacto.

Quanto ao sistema de pontuação, a proposta de Feng & Doolittle (1987) utiliza matriz de substituição PAM e penalidade de abertura de *gap*. Após o trabalho de Feng & Doolittle (1987), surgiu um *software* com algoritmo pa-

recido chamado Clustal (Higgins & Sharp 1988), que se propõe alinhar um grande número de seqüências. Esta ferramenta melhorou no decorrer dos anos e, atualmente, é a mais usada em termos de alinhamento. A seguir será mostrado uma abordagem recente do CLUSTAL.

3.4.1.2 CLUSTALW

O ClustalW é o programa de alinhamento múltiplo mais utilizado para análise de seqüências. São fornecidos ao usuário uma série de parâmetros ajustáveis usados no alinhamento e que causam grande impacto no resultado obtido (Thompson et al. 1994a.).

O algoritmo do ClustalW possui uma idéia semelhante a proposta em Feng & Doolittle (1987). Uma matriz é construída usando o algoritmo Needleman & Wunsch (1970). No entanto, o preenchimento da matriz não utiliza somente o resultado do alinhamento, mas também utiliza um modelo probabilístico para calcular a distância evolutiva entre as seqüências (Kimura 1980).

Além do método padrão para determinação dos nós da árvore guia e realização do alinhamento, o ClustalW tem embutido algumas heurísticas:

- Pesos para as seqüências: para subfamílias grandes atribuí-se peso para que fique explícito sua proximidade;
- Matriz de substituição: a escolha da matriz depende da expectativa de similaridade esperada. Se as seqüências são próximas é conveniente utilizar matriz BLOSUM80. Enquanto que para seqüências que previamente sabe-se que não há similaridade é mais adequado aplicar a matriz BLOSUM45 (seção 3.3.2);
- Redução de penalidade para *gap*: Dada a presença de um *gap*, a penalidade de abertura fica sujeita a observação dos resíduos na posição específica das seqüências subjacentes. Há também redução na penalidade de extensão, quando os *gaps* estendidos compartilham posição com certos tipos de aminoácidos, como por exemplo os hidrofílicos;
- Quanto ao alinhamento progressivo: há possibilidade de adiar a atribuição do valor de parte do alinhamento quando este resulta num *score* baixo. Quando se tem informações dos alinhamentos filhos os devidos ajustes podem ser feitos na árvore e no alinhamento progressivo.

3.4.2 Heurísticas alternativas ao alinhamento progressivo

A maioria dos métodos de alinhamento múltiplo envolvem alinhamento progressivo de Feng & Doolittle (1987) ou outras variações, as quais foram heurísticas pioneiras que propuseram algoritmos para alinhamento múltiplo com vantagem no tempo de execução. Entretanto, o grande problema destas propostas está relacionado ao aparecimento de mínimos locais⁴.

Segundo Notredame & Higgins (1996), uma alternativa para preencher essa lacuna é a determinação de funções objetivo (uma medida de qualidade para o alinhamento), as quais podem inferir preferências a alinhamentos, ou até mesmo, apontar um bom alinhamento.

Há duas possíveis heurísticas que se enquadram nessa idéia. A primeira é por aprendizagem de máquina (Krogh et al. 1994), simulando um alinhamento ao mesmo tempo que tenta encontrar um modelo probabilístico para alterações detectadas (substituições, inserções e exclusões de resíduos), com o intuito de inferir qual a melhor escolha a ser tomada. Contudo, apesar da vantagem de fazer análise de probabilidade, torna-se limitada quando se deseja alinhar muitas seqüências (Notredame & Higgins 1996).

A segunda alternativa é determinar funções objetivos que por si só possam medir a qualidade e apresentar bons *scores*. Se a função objetivo for determinada por critérios específicos referentes às características das seqüências, então há grande chance de se obter alinhamentos de qualidade (Notredame & Higgins 1996).

Medir a qualidade do alinhamento múltiplo não é algo tão simples, pois o número de acessos também é diretamente proporcional ao número de seqüências.

3.5 SAGA

SAGA (do inglês *sequence alignment by genetic algorithm*) é um software que propõe uma heurística de alinhamento múltiplo via algoritmo genético. As seqüências submetidas ao SAGA não passam por um pré-alinhamento, mas ainda assim é possível encontrar alinhamentos múltiplos ótimos, ou quase ótimos, com um tempo razoável (Notredame & Higgins 1996).

O algoritmo de SAGA é baseado no algoritmo genético proposto por Goldberg (1989). Possíveis soluções, ou seja, alinhamentos propriamente ditos, são agrupados em uma população. A população inicial é criada aleatoria-

⁴A conformação do alinhamento é boa, porém existem conformações melhores que não puderam ser encontradas.

mente (geração zero - G_0) segundo um tamanho pré-fixado que se mantém constante durante todas as gerações.

Dada uma geração G_n , a geração seguinte, G_{n+1} , é obtida mediante a utilização de um operador genético. Os operadores do SAGA são divididos em dois conjuntos:

- Cruzamento: mescla o conteúdo de dois alinhamentos pais;
- Mutação: modifica a informação contida no único alinhamento pai selecionado.

O algoritmo proposto por Notredame & Higgins (1996) fornece vinte e dois possíveis operadores genéticos, apresentados na seção 3.5.1 a seguir, e consiste em quatro fases: inicialização, avaliação, procriação e finalização.

Inicialização Consiste na criação aleatória de um conjunto de alinhamentos para fazer parte da geração zero (G_0). Fixou-se que o tamanho da população seria de 100 alinhamentos múltiplos em qualquer geração.

Avaliação Cada alinhamento da geração G_i é avaliado de acordo com a função de mérito selecionada.

Seleção De acordo com a avaliação são selecionados os melhores indivíduos para permanecerem na geração seguinte.

Substituição Como é realizada uma seleção dos melhores indivíduos, então aqueles que são considerados inapropriados são substituídos por novos gerados durante a procriação. Assim, se for determinado que 50% da população será substituída, então metade dos alinhamentos na geração G_{n+1} será composto dos 50% melhores alinhamentos da geração G_i . A outra metade dos indivíduos da geração G_{n+1} será composta pelos alinhamentos filhos.

Procriação Os alinhamentos filhos são gerados a partir dos pais mediante a utilização de um operador genético. O operador que será aplicado é selecionado aleatoriamente de acordo com o peso a ele atribuído dinamicamente. Um aspecto importante da estrutura de população de SAGA é que há a restrição de não haver duplicatas na mesma geração, ou seja, antes de um indivíduo filho ser adicionado é preciso verificar se ele não é idêntico a outro existente.

Finalização O Saga utiliza a estabilização como critério de parada. Quando a busca não apresenta mais melhoras, depois de um número específico de

gerações é considerado que a população se tornou estável. Esta decisão é amplamente usada quando se projeta populações sem duplicatas (Davis 1991).

O pseudo-código abaixo apresenta em linhas gerais o funcionamento do SAGA.

Algoritmo SAGA:

1. Cria G_0
2. Avalie a população da geração n (G_n)
3. Se a população está estabilizada então vá para linha 13
4. Selecione os indivíduos para substituir
5. Avalie o possível descendente esperado
6. Selecione os pais de G_n
7. Selecione o operador
8. $n = n + 1$
9. Gere os novo indivíduo
10. Mantenha ou descarte o no filho na geração G_{n+1}
11. Vá para linha 9 até que todos os filhos tenham sido sucessivamente adicionados a G_{n+1}
12. Vá para linha 2
13. Fim

Uma das vantagens do SAGA é a possibilidade de utilizar qualquer função de mérito, deixando o usuário selecionar entre as pré-definidas ou criar novas. Isto está mais compatível com a realidade biológica, pois a determinação do alinhamento depende das características das seqüências.

A princípio, são utilizadas duas funções de mérito simples para medir a qualidade do alinhamento, ambas usam o método da soma dos pares (definição 3.11) e penalidade de abertura de *gap* (definição 3.4). Com a tentativa de aproximar as seqüências mais semelhantes, é atribuído pesos $W_{k,l}$ para o par de seqüências k e l . Assim, para o software SAGA, a soma dos pares se resume a:

$$SP = \sum_{k < l} W_{k,l} * p(a_i^k, a_i^l)$$

onde $p(a_i^k, a_i^l)$ é originalmente obtido da matriz de substituição PAM250.

Quanto as maneiras de avaliação de *gaps*, algumas possibilidades são descritas em Altschul (1989). Destas, o SAGA analisa:

1. *Quasi-natural gap penalties*;

2. *Natural gap penalties.*

A descrição de ambos os métodos pode ser encontrada em Altschul (1989) e vão além do escopo desta dissertação.

As funções de mérito propostas pelo SAGA diferenciam justamente no modelo de penalidade de *gap*. Contudo, outras variações são possíveis, adaptando-se novas funções de mérito, as quais dizem respeito a:

- usar diferentes conjuntos de pesos para pares de seqüências;
- diferentes custos de substituição (outras matrizes PAM ou tabelas BLO-SUM);
- esquemas diferentes para a pontuação de *gaps*.

3.5.1 Operadores do SAGA

Conforme as especificações tradicionais de algoritmo genético (Goldberg 1989), dois tipos de operadores são representados em SAGA, a mutação e o cruzamento e a diferença entre ambos é que a mutação recebe como parâmetro um alinhamento enquanto o cruzamento é aplicado em dois alinhamentos selecionados.

- **Mutação:** é um programa que realiza modificação em uma seqüência de entrada e resulta em um alinhamento diferente;
- **Cruzamento:** é um programa que seleciona dois alinhamentos da população (pais) para combinar as informações contidas neles e gerar um novo descendente (alinhamento filho).

A seguir são detalhadas as possibilidades de operadores genéticos.

3.5.1.1 Cruzamento

O SAGA fornece dois operadores de cruzamento:

- **Cruzamento pontual:** combina dois alinhamentos progenitores por uma troca simples. Para realizar a combinação das características é escolhida uma posição vertical, ao acaso, no primeiro alinhamento, para que seja traçado um corte linear. Como no segundo alinhamento a conformação vertical dos aminoácidos não é a mesma, ao invés de escolher simplesmente uma posição para que o corte seja feito, é necessário fazer uma busca para que os lados esquerdo e direito de ambos os alinhamentos

possuam os mesmos aminoácidos. Isto possibilita fazer o cruzamento preservando a seqüência original dos aminoácidos.

- **Cruzamento Uniforme:** promove múltiplas trocas entre dois pais pai_1 e pai_2 de uma maneira mais hábil que o cruzamento pontual. Inicialmente são encontrados todos os pontos consistentes entre os pais, sendo que um ponto p_i é dito consistente se as regiões r_i^1 e r_i^2 entre p_i e p_{i+1} possuem os mesmos aminoácidos nos alinhamentos pai_1 e pai_2 , respectivamente. Para a geração de um indivíduo filho basta atribuir a ele a região (r_i^1 ou r_i^2) que apresentar o maior *score*, conforme a função de mérito escolhida para o alinhamento. Entretanto, o SAGA ainda propõe utilizar alguma heurística não determinística para selecionar a região que será passada para a próxima geração.

3.5.1.2 Mutação

3.5.1.2.1 Inserção de *gaps*

Este operador é o mais simples de todos e tem como objetivo prolongar alinhamentos pela inserção de *gaps* nas seqüências. As seqüências do alinhamento são divididas nos grupos G_1 e G_2 . Em cada grupo são inseridos *gaps* de tal forma que, se nas seqüências do grupo G_1 forem inseridos t *gaps* então o grupo de seqüências G_2 receberá também t *gaps* para que as seqüências permaneçam do mesmo tamanho. Existem duas versões para este operador.

- **Versão estocástica:** Para o primeiro grupo de seqüências G_1 a posição de inserção (P_1) é escolhida aleatoriamente. A quantidade de *gaps* que serão adicionados também é escolhida aleatoriamente. Então os t *gaps* são inseridos no grupo G_1 , na mesma posição para todas as seqüências do grupo. Uma região de *gaps* também é adicionada em todas as seqüências do grupo G_2 em posições iguais P_2 . Isso ocorre sem critérios de distância máxima entre P_2 e P_1 .
- **Versão subida de encosta:** A versão subida de encosta para este operador é similar à versão estocástica. Contudo, ao invés de todos os parâmetros serem aleatórios, a posição P_1 é tratada diferentemente, pois todos os seus possíveis valores são testados e analisados, então a posição P_1 que obter melhor alinhamento é escolhida.

3.5.1.2.2 Movimentação de blocos

Esse operador tem a finalidade de movimentar blocos de *gaps* ou de resíduos, ora à direita ora à esquerda para alcançar outra conformação. Esse tipo de mutação parte do pressuposto de que, em um grande número de casos, para encontrar um alinhamento ótimo basta realizar uma movimentação de blocos.

A definição usual de bloco é uma subseção do alinhamento que não contém *gaps* e todas as subsequências possuem o mesmo tamanho. Entretanto, para o propósito deste operador, a melhor definição de um bloco de resíduos é um conjunto de trechos sem *gaps* cujas regiões se sobrepõem em uma ou mais seqüências que são delimitadas por *gaps* ou pelas extremidades da cadeia.

Um bloco é escolhido pela seleção de um resíduo ou uma posição que contenha *gap*, então obtém-se o bloco que contém esta posição escolhida. Os blocos de resíduos podem ser movidos dentro do alinhamento para gerar novas configurações.

Um *gap* só pode ser deslocado até que se encontre outro *gap*. Da mesma forma, um bloco de resíduos só pode ser deslocado até se encontrar com outro bloco de resíduos. Abaixo as diferentes maneiras que esta operação pode ser usada:

- Mover um bloco de *gaps* ou um bloco de resíduos;
- Separar o bloco horizontalmente e mover um dos sub-blocos à esquerda ou à direita;
- Separar o bloco verticalmente e mover a metade a esquerda ou a direita;
- O movimento do bloco pode ser feito calculando a melhor posição possível ou de uma maneira estocástica.

Fazendo diferentes combinações para o operador que movimenta os blocos surgem um total de 16 possíveis operadores e todos os 16 operadores são implementados em SAGA.

3.5.1.2.3 Pesquisa de blocos

O operador de pesquisa de blocos agiliza o processo de encontrar um alinhamento próximo do ótimo. Este operador de mutação resume-se a um método que, dada uma subsequência em uma das seqüências, tenta localizar no alinhamento o bloco ao qual ela provavelmente pertence.

Primeiramente são selecionados de forma aleatória o tamanho da subsequência e a posição em uma das seqüências. Então, são comparadas todas as subsequências de mesmo comprimento em outra seqüência. O melhor emparelhamento encontrado é então selecionado e esta nova subsequência é adicionada à primeira, de modo a formar um pequeno perfil.

O operador continua pesquisando nas seqüências restantes o melhor emparelhamento que, ao ser localizado, é adicionado ao perfil. Este processo continua até que um emparelhamento tenha sido identificado em todas as seqüências. Ao final do processo, as seqüências são então movidas para reconstruir um bloco dentro do alinhamento.

Segundo Notredame & Higgins (1996), o operador de pesquisa de blocos gera mudanças mais significativas que os outros operadores. Entretanto, um algoritmo genético que não possui este operador também alcançará um resultado aceitável, só que seria mais lento.

3.5.1.2.4 Rearranjo ótimo ou sub-ótimo local

Algumas situações geram a presença de um mínimo local muito estável, o que dificulta a busca por uma configuração ótima. Para contornar esse problema, um último operador foi projetado como uma tentativa de otimizar o padrão de *gaps* dentro de um dado bloco. Isto é feito por análise exaustiva de todos os possíveis arranjos de *gap* dentro do bloco ou por alinhamento com algoritmo genético (*Local Alignment Genetic Algorithm - LAGA*) em uma pequena região pré-fixada.

Se for necessário examinar menos de 2000 combinações, a busca é realizada por análise exaustiva. Caso contrário, LAGA é utilizado, o qual é uma versão pura do algoritmo genético proposto Goldberg (1989). O LAGA usa apenas o cruzamento pontual (subseção 3.5.1.1) e operador de movimentação de blocos (subseção 3.5.1.2.2). Normalmente é executado por várias gerações com a população de 20 indivíduos.

Capítulo 4

O Algoritmo Genético Baseado em Tipos Abstratos de Dados

4.1 Introdução

Algoritmos genéticos são métodos computacionais adotados em problemas em que o espaço de busca cresce de forma exponencial, como é o caso do problema do alinhamento múltiplo de proteínas, que pertence a classe de problemas NP-completos.

A forma mais comum de utilizar algoritmos genéticos é tanto como uma técnica de aprendizagem de máquina, quanto uma busca heurística (ou ambas), de tal forma que se enquadra na área conhecida como inteligência artificial.

No capítulo anterior foi apresentada uma técnica de alinhamento múltiplo baseada em algoritmos genéticos. A estrutura da população e os operadores genéticos aplicados não apresentam uma estrutura de dados organizada. Assim, pode-se observar que a única semelhança do algoritmo de SAGA aos conceitos de evolução genética diz respeito a denominação população e operadores genéticos. Além disso, comumente a modelagem de algoritmos genéticos é específica para o problema que se deseja resolver. Para preencher essa lacuna, este capítulo apresenta uma abordagem de algoritmo genéticos com tipos abstratos de dados definidos pelo usuário para um dado problema.

O trabalho de Vieira (2003) foi pioneiro com respeito a técnica de algoritmos genéticos baseados em tipos abstratos de dados, do inglês *Genetic Algorithm based on Abstract Data Types* - GAADT, onde são apresentadas definições de como os dados devem ser tratados e manipulados para encontrar soluções. Este algoritmo serve de modelo para utilização de algoritmos genéticos, de tal forma que não auxilia apenas na estruturação, mas também facilita delimitar as propriedades do problema nas fases de requisito e modelagem.

O presente capítulo objetiva exemplificar a utilização do algoritmo genético descrito por Vieira (2003) com a modelagem de mapas conceituais, um problema com formulação mais simples que o alinhamento múltiplo de proteínas, mas que permite ilustrar de forma adequada os aspectos importantes de uma instanciação do algoritmo genético baseado em tipos abstratos de dados.

4.2 Avaliação da Aprendizagem com Mapas Conceituais

Uma questão em aberto na área de aprendizagem guiada por computador é a avaliação da aprendizagem por meio de Mapas Conceituais (MCs). Mapa Conceitual é um processo de construção do conhecimento (Rocha et al. 2004), que resulta em combinações significativas na forma de proposições.

Um conceito é uma regularidade percebida em eventos, objetos ou percepções a respeito de eventos ou objetos, onde o conceito aprendido é indicado por um rótulo (Novak 1998). Uma proposição é uma combinação entre dois conceitos conectados por uma frase ou palavra de enlace (Novak 1998). Uma frase de enlace indica uma relação binária que existe entre dois conceitos de uma proposição (Rocha et al. 2004). Frases de enlace são consideradas metadados que possuem uma hierarquia de tipos (Fischer 2001).

Exemplo 7 *Conceitos e Frases de enlace.*

- *A aprendizagem a respeito do ciclo da água envolve alguns conceitos, dentre os quais pode-se citar: <CICLO DA ÁGUA>, <CONDENSAÇÃO>, <EVAPORAÇÃO> e <PRECIPITAÇÃO>.*
- *Duas possíveis proposições são: <CICLO DA ÁGUA tem como fase CONDENSAÇÃO> e <CONDENSAÇÃO é estágio de CICLO DA ÁGUA>.*
- *Os conectivos <tem como fase> e <é estágio de> são frases de enlace que podem indicar valores dos supertipos temporal.*

Usar mapas conceituais para avaliar a aprendizagem é uma tarefa importante, pois sua estrutura é formada por elementos cognitivos que representam a evidência da aprendizagem (Novak & Gowin 1999), havendo interação entre o novo conhecimento e o conhecimento previamente adquirido. A interação entre os conhecimento faz com que ambos se modifiquem ganhando novas percepções, com mais clareza, estabilidade e diferenciação.

Assim, durante a construção do conhecimento, as estruturas cognitivas vão se reestruturando através da diferenciação progressiva e reconciliação integrativa. Na diferenciação progressiva, as idéias mais inclusivas são previamente expostas e o estudante aumenta o grau de elaboração destas idéias conforme aumenta seu aprendizado sobre elas, tornando-as progressivamente diferenciadas com a inclusão de detalhes mais específicos.

Enquanto isso, a reconciliação integrativa estabelece relações entre elementos existentes na estrutura cognitiva, chamando atenção para diferenças e similaridades inicialmente não percebidas, reconciliando inconsistências reais ou aparentes fazendo assim uma reorganização na estrutura cognitiva (Moreira 1980).

A figura 4.1 apresenta um exemplo de mapa conceitual fornecido pelo estudante, o qual será utilizado no decorrer do capítulo. Neste mapa, o conceito EVAPORAÇÃO foi considerado mais inclusivo. As linhas contínuas representam a diferenciação progressiva de conceitos, enquanto a linha pontilhada sugere a reconciliação integrativa.

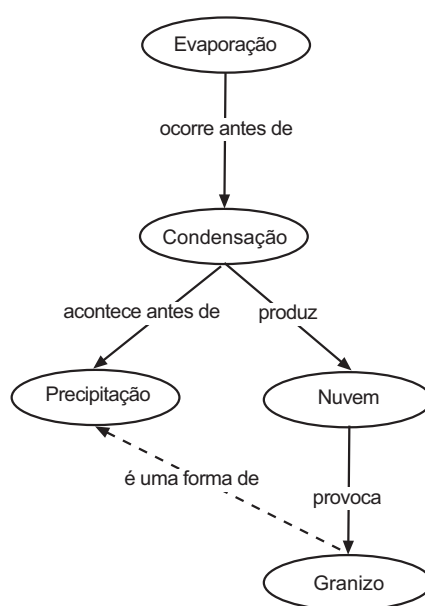


Figura 4.1: Exemplo de Mapa Fornecido Pelo Estudante

Dados os conceitos do domínio CICLO DA ÁGUA eles podem ser representados de diferentes maneiras na construção de mapas conceituais. Dessa forma, um único mapa conceitual é incapaz de representar todas as situações de aprendizagem envolvidas.

Segundo (Rocha et al. 2004), uma alternativa é oferecer ao professor a possibilidade de criar ricas ontologias de domínio para armazenar os conceitos e suas relações no contexto de determinada tarefa de aprendizagem. Então

optou-se por usar algoritmo genético para simular todas as possibilidades de aprendizagem e produção das coleções de MCs. Assim, o professor avalia o espaço de busca gerado pelo algoritmo genético e a ontologia para detectar evidências de aprendizagem no mapa conceitual de um estudante ou grupos de estudantes.

Seja o grafo direcionado $G = (V, A)$ que especifica a ontologia de domínio CICLO DA ÁGUA ¹. O conjunto de vértices do grafo representa todos os conceitos de possíveis MCs e as arestas rotuladas representam as proposições entre dois conceitos, tal que:

- Conceitos: $V(G)$.
 $V(G) = \{ \text{Ciclo da Água, Condensação, Evaporação, Precipitação, Vapor de Água, Neblina, Geadas, Orvalho, Nuvem, Chuva, Neve, Granizo} \}$.
- Relações Binárias: $A(G)$.
 A Tabela 4.1 representa a matriz de adjacência do grafo, onde $A[i, j]$ possui uma relação r_x quando há aresta entre os conceito i e j . Caso contrário $A[i, j] = \emptyset$.

| | Ciclo da Água | Condensação | Evaporação | Precipitação | Vapor de Água | Neblina | Geadas | Orvalho | Nuvem | Chuva | Neve | Granizo |
|---------------|---------------|-------------|------------|--------------|---------------|---------|--------|---------|----------|----------|----------|----------|
| Ciclo da Água | | r_1 | r_1 | r_1 | | | | | | | | |
| Condensação | r_3 | | r_4 | r_2 | | r_5 | r_5 | r_5 | r_5 | | | |
| Evaporação | r_3 | r_2 | | | r_6 | | | | | | | |
| Precipitação | r_3 | r_4 | | | | | | | | r_{13} | r_{13} | r_{13} |
| Vapor de Água | | | r_{10} | | | r_7 | r_7 | r_7 | r_7 | | | |
| Neblina | | r_9 | | | r_{11} | | | | | | | |
| Geadas | | r_9 | | | r_{11} | | | | | | | |
| Orvalho | | r_9 | | | r_{11} | | | | | | | |
| Nuvem | | r_9 | | | r_{11} | | | | | r_8 | r_8 | r_8 |
| Chuva | | | | r_{14} | | | | | r_{12} | | | |
| Neve | | | | r_{14} | | | | | r_{12} | | | |
| Granizo | | | | r_{14} | | | | | r_{12} | | | |

Tabela 4.1: Matriz de Adjacência

¹Fonte da Ontologia: Rocha et al. (2004)

- Frases de enlace: arestas ponderadas.

Seja $G = (V, A)$ ponderado, em qualquer mapa conceitual construído a partir da ontologia as relações binárias têm um valor associado a cada aresta, cujos rótulo e tipo podem ser encontrados na tabela 4.2.

| Arestas do Tipo TEMPORAL | |
|---------------------------------------|----------------------------------------|
| r_1 : | {tem como fase, tem como estágio} |
| r_2 : | {precede, ocorre antes de} |
| r_3 : | {é fase de, é estágio de} |
| r_4 : | {sucede, ocorre depois de} |
| Arestas do Tipo AÇÃO | |
| r_5 : | {forma, produz} |
| r_6 : | {produz, gera} |
| r_7 : | {é convertido em, transforma-se em} |
| r_8 : | {provoca, causa} |
| r_9 : | {resulta de, é produzido por} |
| r_{10} : | {é produzido por, é resultado de} |
| r_{11} : | {é transformação de, é modificação de} |
| r_{12} : | {é provocada por, é causada por} |
| Arestas do Tipo CARACTERÍSTICA | |
| r_{13} : | {pode ser, pode aparecer como} |
| r_{14} : | {é um tipo de, é uma forma de} |

Tabela 4.2: Valores das Relações Binárias da Ontologia

Dado o mapa do estudante e a ontologia de domínio, a seguir são apresentadas as características do GAADT. O enfoque desta instanciação está em comparar o espaço de busca gerado pelo GAADT com o MC do estudante, averiguando todas as formas alternativas de construção do conhecimento permitidas pela ontologia.

4.3 Tipos Básicos

A representação do resultado com GAADT requer uma codificação em três níveis de percepção denominadas de base, gene e cromossomo. A solução de um problema com GAADT requer que o tipo do resultado seja mapeado para uma estrutura denominada cromossomo, onde cada cromossomo possui genes, cujas unidades elementares de formação são as bases.

Definição 4.1 (Base) *Uma base B é o conjunto de todas as unidades genéticas elementares que podem ser usadas na formação do material genético dos cromossomos de uma população.*

Por exemplo, para a construção de MCs, as unidades são formadas por uma relação binária entre os conceitos, onde a base é dada por:

$$B = c \cup e,$$

onde: c é o conjunto dos conceitos existentes numa ontologia de domínio; e é o conjuntos de relações caracterizadas por uma 3-upla (e_1, e_2, e_3) , que representam respectivamente o TIPO DA RELAÇÃO, VALOR DA RELAÇÃO, APRENDIZAGEM DA RELAÇÃO.

Numa instanciação em relação aos dados da ontologia Ciclo da Água (tabela 4.1), possíveis valores são: para o tipo da relação $e_1 = \{\text{temporal, ação, característica}\}$; o valor da relação $e_2 = \{r_1, \dots, r_{14}\}$ (tabela 4.2); e a aprendizagem da relação é dada por $e_3 = \{d, r\}$ sendo d para diferenciação progressiva e r para reconciliação integrativa.

As características do cromossomo são agrupadas em genes, que são formados por uma seqüência de elementos base. Contudo, não é qualquer seqüência de bases que caracteriza um cromossomo. Para tal, deve ser definida uma *lei de formação* para enunciar o modo pelo qual as bases devem se interligar para que o gene apresente uma característica admissível.

O GAADT propõe uma *lei de formação* de características por um conjunto de *Axiomas de Formação de Genes (AFG)*, o qual deverá ser definido para cada caso de acordo com a semântica atribuída ao gene.

Definição 4.2 (Gene) *Um gene G é uma seqüência formada pelos elementos da base B que pertence ao conjunto AFG .*

A minuciosidade dos AFG para a formação do gene só é definida no momento de instanciação do GAADT a um problema específico. Por exemplo, o AFG para o problema de construção de mapas conceituais assegura que as relações expressas pelos elementos do gene fazem parte da ontologia considerada.

Assim, o conjunto de possíveis genes de um MC é formado por todas as relações binárias que satisfazem o AFG. A seqüência de bases que forma um gene é dada por:

$$g_k = \langle c_{k,i}, e_k, c_{k,j} \rangle,$$

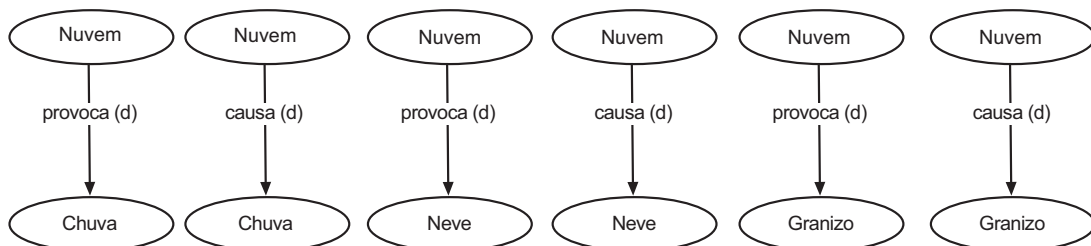
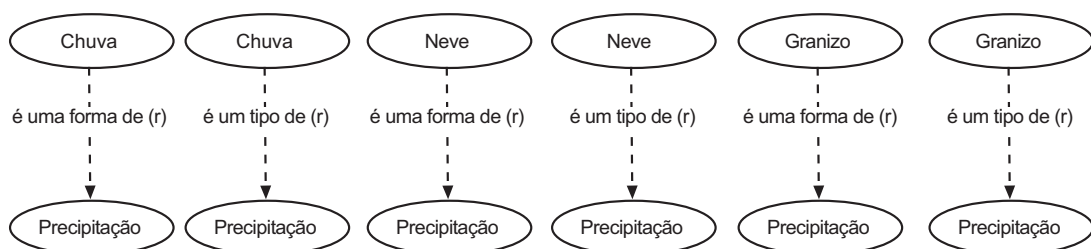
onde k é a característica de dado gene, $c_{k,i}$ forma uma proposição com o conceito $c_{k,j}$ e a frase de enlace é dada pela relação $e_k = (e_{k,1}, e_{k,2}, e_{k,3})$.

O AFG é formado por um conjunto de axiomas que indicam as conformações válidas para os genes. Para a ontologia de domínio ciclo da água mostrada anteriormente, dois possíveis axiomas são descritos na tabela 4.3:

| | |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Ex1: | $afg_1 \equiv (\forall g_k \in g \ c_{k,i} = \text{NUVEM} \wedge c_{k,j} \in \{\text{CHUVA, NEVE, GRANIZO}\} \rightarrow e_{k,1} = \text{AÇÃO} \wedge e_{k,2} \in r_8 \wedge e_{k,3} \in \{d, r\})$ <p>Para todo gene g_k, cujo conceito $c_{k,i}$ é a NUVEM e o conceito $c_{k,j}$ pertencendo ao conjunto de conceitos {CHUVA, NEVE, GRANIZO}, então a relação só poderá ser do tipo AÇÃO, com os valores pertencentes ao subconjunto de frases de enlace r_8 contidos na relação do tipo ação definido na ontologia. A relação pode ser compreendida por diferenciação progressiva (d) ou reconciliação integrativa (r).</p> |
| Ex2: | $afg_2 \equiv (\forall g_k \in g \ c_{k,i} \in \{\text{CHUVA, NEVE, GRANIZO}\} \wedge c_{k,j} = \text{PRECIPITAÇÃO} \rightarrow e_{k,1} = \text{CARACTERÍSTICA} \wedge e_{k,2} \in r_{14} \wedge e_{k,3} \in \{d, r\})$ <p>Para todo gene g_k, cujo conceito $c_{k,i}$ pertencendo ao conjunto de conceitos {CHUVA, NEVE, GRANIZO} e o conceito $c_{k,j}$ é a PRECIPITAÇÃO, então a relação só poderá ser do tipo CARACTERÍSTICA, com os valores pertencentes ao subconjunto de frases de enlace r_{14} contidos na relação do tipo característica definido na ontologia. A relação pode ser compreendida por diferenciação progressiva (d) ou reconciliação integrativa (r).</p> |

Tabela 4.3: Exemplos de AFG para Mapas Conceituais

Abaixo são mostrados alguns genes obtidos pela aplicação dos axiomas afg_1 e afg_2 , respectivamente na figura 4.2 e figura 4.3, onde a aprendizagem dos genes produzidos pelo afg_1 é dada por diferenciação progressiva e para o afg_2 por reconciliação integrativa.

Figura 4.2: Alguns Genes Produzidos pelo afg_1 Figura 4.3: Alguns Genes Produzidos pelo afg_2

Com buscas na ontologia do domínio, outros axiomas podem ser formados para as demais relações binárias. Assim, para que o GAADT possa realizar as operações em busca da solução primeiramente o tipo gene deve ser definido, bem como todas as restrições imposta pelo AFG.

Para representar um resultado o GAADT agrupa os genes em seqüências formando o tipo abstrato cromossomo. Portanto, um cromossomo da população é formado por um conjunto de genes $\{g_1, g_2, \dots, g_n\}$, onde este conjunto fornece a informação necessária para identificá-lo dentro da população.

Não é permitida duplicação de cromossomos na mesma população e a identidade de cada cromossomo é chave desta regra. O GAADT opõe-se em deixar coexistir cópias de um mesmo cromossomo em qualquer tempo durante a busca por um cromossomo mais adaptado, apesar desta comparação ser uma busca exaustiva.

Da mesma forma, não é permitido que cromossomos extintos possam retornar ao conjunto de soluções, pois exclusões de características implica que elas não satisfazem o problema. A única exceção para isto é quando o ambiente sofre alteração.

As características dos cromossomos de uma população auxiliam também a classificá-los em grupos taxonômicos (espécies e famílias) em função do grau atribuído às características compartilhadas pelos mesmos. O caráter do GAADT em gerar cromossomos também é estabelecido por uma *lei de formação* gerando um conjunto de *Axiomas de Formação de Cromossomos (AFC)*.

Definição 4.3 (Cromossomo) *Um cromossomo C é um conjunto de genes que obedece às condições estabelecidas pelo AFC.*

As definições axiomáticas no que tange a associação dos genes só podem ser descritas quando a instanciação de um problema é projetada. Volta-se então ao exemplo de produção de MCs. O AFC que determina mapas conceituais corretos de acordo com os princípios teóricos de estrutura hierárquica, diferenciação progressiva e reconciliação integrativa tem duas leis de formação (Rocha et al. 2004), apresentadas na tabela 4.4.

| | |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| afc_1 | $afc_1 \equiv (\forall c \in C \forall g_1, g_k \in c (\exists g_2, \dots, g_{k-1} \in c \exists i \in \{1, 2, \dots, k-1\} c_{i,1} = c_{i+1,1} \vee c_{i,1} = c_{i+1,2} \vee c_{i,2} = c_{i+1,1} \vee c_{i,2} = c_{i+1,2}))$ <p>Para todo cromossomo c, cujo conjunto de genes tem cardinalidade maior ou igual a k, existe pelo menos um caminho entre todos os seus pares de genes g_i, g_k percorrendo os genes de c no sentido g_i para g_{i+1}, ou vice-verso, que começa no conceito $c_{i,1}$ e termina no conceito $c_{k,2}$. Isto é, o mapa conceitual representado pelo cromossomo c é conexo;</p> |
| afc_2 | $afc_2 \equiv (\forall c \in C \exists x \in C (x = \{g_i \mid \forall g_i \in c e_{i,3} = d\}) \wedge (\forall g_1, g_k \in x \exists_1 g_2, \dots, g_{k-1} \in c \exists i \in \{1, 2, \dots, k-1\} (c_{i,1} = c_{i+1,1} \vee c_{i,1} = c_{i+1,2} \vee c_{i,2} = c_{i+1,1} \vee c_{i,2} = c_{i+1,2})))$ <p>Para todo cromossomo c existe um cromossomo x, cujo conjunto de genes é um subconjunto próprio de genes de c composto somente por proposições compreendidas por diferenciação progressiva que forma uma árvore. Isto é, a aprendizagem por diferenciação progressiva de um mapa conceitual é um grafo conexo acíclico.</p> |

Tabela 4.4: Exemplos de AFC para Mapas Conceituais

O *AFC* é definido como um conjunto de conjuntos das possíveis conformações do tipo abstrato gene, que por sua vez é definido pelo *AFG*. Cada elemento desse superconjunto tem que obedecer a esta propriedade.

No contexto de mapas conceituais o mapa fornecido pelo estudante tem sua correspondência cromossômica, onde cada relação binária é mapeada em um tipo abstrato gene. Dado o mapa do estudante mostrado na figura 4.1, a tabela 4.5 apresenta o mapeamento do mesmo para o tipo abstrato cromossomo.

| Cromossomo Aluno |
|--------------------------------------------------------------------------------------------|
| $g_1 = \langle \text{EVAPORAÇÃO, (TEMPORAL, ocorre antes de, d), CONDENSAÇÃO} \rangle$ |
| $g_2 = \langle \text{CONDENSAÇÃO, (TEMPORAL, acontece antes de, d), PRECIPITAÇÃO} \rangle$ |
| $g_3 = \langle \text{CONDENSAÇÃO, (AÇÃO, produz, d), NUVEM} \rangle$ |
| $g_4 = \langle \text{NUVEM, (AÇÃO, provoca, d), GRANIZO} \rangle$ |
| $g_5 = \langle \text{GRANIZO, (CARACTERÍSTICA, é uma forma de, r), PRECIPITAÇÃO} \rangle$ |

Tabela 4.5: Mapa do Aluno - Correspondência Cromossômica

População é o último tipo básico a ser definido. Uma população guarda um conjunto de indivíduos, representados pelos cromossomos, que em determinado momento são considerados proveitosos para resolver o problema.

Definição 4.4 (População) *Uma população P é um conjunto de cromossomos construídos conforme descrito na Definição 4.3.*

A figura 4.4 apresenta um exemplo de população de mapas conceituais, onde $P = \{c_1, c_2, c_3\}$.

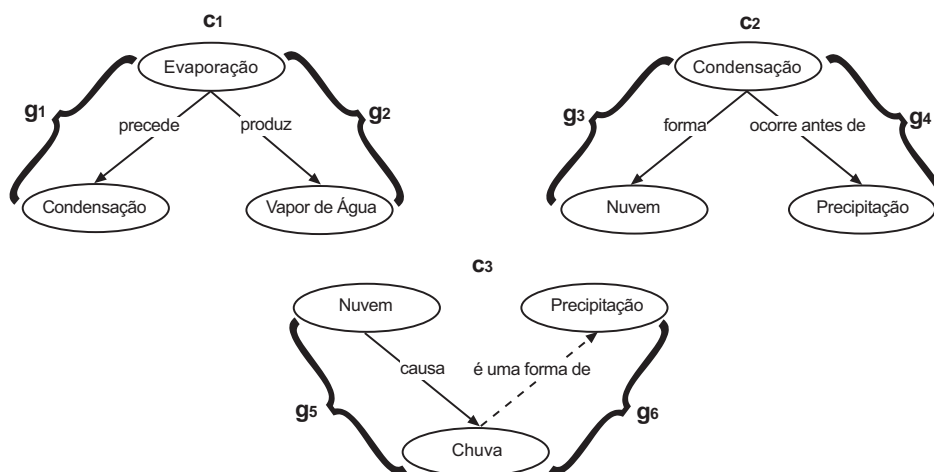


Figura 4.4: Exemplo de Cromossomos Fornecidos pelo GAADT

O tipo população pode ser alterado para aproximar seus cromossomos da solução esperada para o problema. Alterações genéticas são então realizadas para diversificar as características das gerações futuras. A seguir são apresentados os operadores genéticos propostos pelo GAADT.

4.4 Operadores Genéticos

O GAADT trabalha com dois tipos de operadores genéticos. O primeiro deles é o cruzamento ou reprodução, cuja finalidade é mesclar os genes de dois cromossomos. Dois cromossomos, chamados cromossomos pai, são selecionados e suas características são combinadas para formar novos cromossomos, denominados cromossomos filhos.

O segundo é chamado operador de mutação que tem o objetivo de alterar a identidade de um cromossomo para formar um novo, denominado cromossomo mutante.

A informação que é passada de pai para filho é representada pelo gene que melhor satisfaz as restrições do problema, denominado gene dominante.

Dada uma característica específica, tal que em um cromossomo ela é expressa pelo gene g_x e no segundo cromossomo pelo gene g_y . Diz-se que g_x melhor satisfaz os requisitos do problema do que g_y , se e somente se, o grau de adaptação de g_x for maior ou igual ao grau de adaptação de g_y . O GAADT define o grau de adaptação de um gene pela função *grau*.

Definição 4.5 (Grau) *O grau de adaptação de um gene é uma função grau do seguinte tipo: $grau : G \rightarrow K$ tal que, a cada gene g , $g \in G$, é associado um único número k , $k \in K$ (K é um corpo ordenado²), chamado de $grau(g)$ e que reflete, segundo a interpretação adotada para o problema, uma estratificação comparativa entre a adaptação dos genes.*

Para o exemplo de MCs o grau de adaptação de um gene é uma função $grau : G \rightarrow \mathbb{R}$, tal que:

$$grau(g_k) = \begin{cases} 2 & \text{se } g_k \in \text{exatamente à ontologia} \\ 1 & \text{se } g_k \in \text{aproximadamente à ontologia} \\ 0 & \text{se c.c.} \end{cases}$$

O termo *pertencer exatamente à ontologia* necessita de duas verificações:

- o gene expressa uma relação binária que existe;

²Estrutura algébrica, com duas operações, sem divisores próprios de zero e munido de uma ordem. Ex: $(\mathbb{R}, \leq, +, \times, 0, 1)$.

- o valor e o supertipo da frase de enlace coincide exatamente com uma das instâncias da taxonomia que une supertipos e frases de enlace (Rocha et al. 2004).

Já o termo *pertencer aproximadamente à ontologia* ocorre quando só o supertipo da frase de enlace é descrito na ontologia.

A tabela 4.6 apresenta os graus dos genes do mapa do aluno (figura 4.1). Observa-se que a frase de enlace do gene g_2 não pertence exatamente à ontologia, porém seu tipo relação é possível. O grau desse gene não é totalmente penalizado, pois os valores das relações podem ter palavras semelhantes que poderiam ser pesquisadas, por exemplo em um dicionário de sinônimos.

| Genes | $grau(g_i)$ |
|--------------------------------------------------------------------------------------------|-------------|
| $g_1 = \langle \text{EVAPORAÇÃO, (TEMPORAL, ocorre antes de, d), CONDENSAÇÃO} \rangle$ | 2 |
| $g_2 = \langle \text{CONDENSAÇÃO, (TEMPORAL, acontece antes de, d), PRECIPITAÇÃO} \rangle$ | 1 |
| $g_3 = \langle \text{CONDENSAÇÃO, (AÇÃO, produz, d), NUVEM} \rangle$ | 2 |
| $g_4 = \langle \text{NUVEM, (AÇÃO, provoca, d), GRANIZO} \rangle$ | 2 |
| $g_5 = \langle \text{GRANIZO, (CARACTERÍSTICA, é uma forma de, r), PRECIPITAÇÃO} \rangle$ | 2 |

Tabela 4.6: Mapa do Aluno - Grau dos Genes

Quando um gene não expressa uma característica aceitável ele é denominado gene-inócuo, cujo símbolo é representado por g_λ . A presença ou ausência de g_λ no cromossomo não altera sua identidade, pois este gene também satisfaz as regras do *AFG*. Dessa forma, o gene inócuo será uma constante do sistema, cujo valor deve ser definido no momento da instanciação do algoritmo.

Ao grau de adaptação do gene inócuo é atribuído o menor grau possível dentre qualquer outro grau atribuído a genes que expressam características. O grau de g_λ deve ser igual ao elemento neutro do corpo K para a operação de adição, neste caso o grau de g_λ é igual a zero.

Após a definição de grau de adaptação do gene, há a possibilidade de comparar pares de genes que representam a mesma característica. Diz-se que dois genes expressam a mesma característica se eles contém algum atributo relevante para o problema em questão.

O GAADT especifica os atributos relevantes pelo conjunto *atributoRelevante*, que é uma constante do sistema e seu conteúdo depende da interpretação dada ao problema. Para verificar se um par de genes expressam a mesma característica foi definida a relação *mesma*, que é especificada pelos seguintes lemas:

Lema 4.4.1 $\forall g : G | (g, g) \in mesma.$

Lema 4.4.2 $\forall g_1, g_2 : G | (g_1, g_2) \in mesma \rightarrow (g_2, g_1) \in mesma.$

Lema 4.4.3 $\forall g_1, g_2, g_3 : G \mid (g_1, g_2) \in mesma \wedge (g_2, g_3) \in mesma \rightarrow (g_1, g_3) \in mesma.$

Por exemplo, no problema de construção de MCs o atributo relevante é que haja caminho entre os grafos dos genes g_x e g_y , conseqüentemente a relação *mesma* é dada por:

$$(g_x, g_y) \in mesma \rightarrow ((c_{x,1} = c_{y,1}) \vee (c_{x,1} = c_{y,2}) \vee (c_{x,2} = c_{y,1}) \vee (c_{x,2} = c_{y,2}))$$

onde $c_{i,j}$ representa o conceito j do gene g_i .

O GAADT defini a função *domi* denominada gene dominante, que recebe dois genes de cromossomos pai diferentes. Se os genes fornecidos dizem respeito a uma mesma característica, então a função retorna o gene com maior grau de adaptação. Caso os genes não representem a mesma característica, então a função *domi* retorna g_λ apenas para tornar a função um algoritmo válido, pois neste caso o gene-inócuo não é considerado um dominante propriamente dito.

Definição 4.6 (Dominante) *O gene dominante é uma função domi do seguinte tipo:*

$$domi : G \times G \rightarrow G$$

$$domi(g_1, g_2) = \begin{cases} g_\lambda & \text{se } (g_1, g_2) \notin mesma, \\ g_1 & \text{se } (g_1, g_2) \in mesma \wedge grau(g_1) \geq grau(g_2), \\ g_2 & \text{se } (g_1, g_2) \in mesma \wedge grau(g_1) < grau(g_2). \end{cases}$$

Para o exemplo de construção de MCs, gene dominante é uma função *domi* : $G \times G \rightarrow G$, tal que:

$$domi(g_x, g_y) = \begin{cases} g_\lambda & \text{se } ((c_{x,1} \neq c_{y,1}) \wedge (c_{x,1} \neq c_{y,2}) \wedge (c_{x,2} \neq c_{y,1}) \wedge (c_{x,2} \neq c_{y,2})); \\ g_x & \text{se } ((c_{x,1} = c_{y,1}) \vee (c_{x,1} = c_{y,2}) \vee (c_{x,2} = c_{y,1}) \vee (c_{x,2} = c_{y,2})) \wedge \\ & (grau(g_x) \geq grau(g_y)); \\ g_y & \text{se } ((c_{x,1} = c_{y,1}) \vee (c_{x,1} = c_{y,2}) \vee (c_{x,2} = c_{y,1}) \vee (c_{x,2} = c_{y,2})) \wedge \\ & (grau(g_x) < grau(g_y)); \end{cases}$$

onde $c_{i,j}$ representa o conceito j do gene g_i .

O fato de os indivíduos mais adaptados da população terem mais chances de transmitir suas características às gerações futuras, faz com que haja análise da adaptação do cromossomo como todo, que é definida pela função *adapt*.

Definição 4.7 (Adaptação) *A adaptação de um cromossomo é uma função adapt do seguinte tipo:*

$$adapt : C \rightarrow K$$

$$adapt(c) = \sum_{g \in c} \Theta_{c,g} \times grau(g)$$

onde $\Theta_{c,g}$ é o peso com o qual o gene g contribui para a adaptação do cromossomo c .

O valor de Θ usado para calcular a adaptação do cromossomo é mais um parâmetro que deve ser definido quando um problema é instanciado, examinando a presença ou ausência de uma dada característica gênica no cromossomo.

Como exemplo, para a construção de uma população de MCs seu objetivo é comparar o MC do estudante com todas as estruturas alternativas de construção do conhecimento permitidas pela ontologia. Para isto, os MCs gerados pelo GAADT devem ser comparáveis ao MC do estudante. Além disso, um requisito específico deste problema é que os conceitos expostos pelo estudante devem ter prioridade sobre os conceitos contidos nos genes do GAADT.

Dessa forma, os cromossomos que apresentarem maior quantidade de conceitos contidos no mapa do estudante terão maiores valores de adaptação. Assim, os valores de Θ de cada gene g do cromossomo c é dado por:

$$\Theta = \begin{cases} 2 & \text{se } g \text{ contiver os dois conceitos do mapa do estudante} \\ 1 & \text{se } g \text{ apresentar somente um conceito igual ao mapa do estudante} \\ 0 & \text{se c.c} \end{cases}$$

Exemplo 8 Seja o mapa do estudante da figura 4.1. Dado que foram obtidos do GAADT três possíveis cromossomos (figura 4.4), onde $c_1 = \{g_1, g_2\}$, $c_2 = \{g_3, g_4\}$ e $c_3 = \{g_5, g_6\}$.

A tabela 4.7 mostra o cálculo da adaptação de cada um desses cromossomos. Pode-se observar que o cromossomo c_2 é mais próximo semanticamente do cromossomo do estudante do que c_1 e c_3 .

| | Gene | Θ | $grau(g_i)$ |
|--------------|------------------------------------------------------------------------------------------|----------|-------------|
| c_1 | $g_1 = \langle \text{EVAPORAÇÃO, (TEMPORAL, precede, d), CONDENSAÇÃO} \rangle$ | 2 | 2 |
| | $g_2 = \langle \text{EVAPORAÇÃO, (AÇÃO, produz, d), VAPOR DE ÁGUA} \rangle$ | 1 | 2 |
| $adapt(c_1)$ | $\sum_{g \in c} \Theta_{c,g} \times grau(g) = 6$ | | |
| c_2 | $g_3 = \langle \text{CONDENSAÇÃO, (AÇÃO, forma, d), NUVEM} \rangle$ | 2 | 2 |
| | $g_4 = \langle \text{CONDENSAÇÃO, (TEMPORAL, ocorre antes de, d), PRECIPITAÇÃO} \rangle$ | 2 | 2 |
| $adapt(c_2)$ | $\sum_{g \in c} \Theta_{c,g} \times grau(g) = 8$ | | |
| c_3 | $g_5 = \langle \text{NUVEM, (AÇÃO, causa, d), CHUVA} \rangle$ | 1 | 2 |
| | $g_6 = \langle \text{CHUVA, (CARACTERÍSTICA, é uma forma de, r), PRECIPITAÇÃO} \rangle$ | 1 | 2 |
| $adapt(c_3)$ | $\sum_{g \in c} \Theta_{c,g} \times grau(g) = 4$ | | |

Tabela 4.7: Exemplo de Adaptação de Cromossomos em MCs

Antes de definir o operador de cruzamento propriamente dito, ainda é necessário definir duas funções que precedem o operador. A primeira função

diz respeito a seleção de uma subpopulação que satisfaz os requisitos r previamente definidos pelo problema. Estes requisitos indicam quando um dado cromossomo é considerado apto a reproduzir novos cromossomos.

Definição 4.8 (Seleção) *A seleção dos cromossomos que satisfazem um predicado r é uma função sel do seguinte tipo:*

$$sel : \mathbb{P}(P) \times \mathbb{P}(P) \rightarrow \mathbb{P}(P)$$

$$sel(P_1, r) = P_1 \cap r.$$

Por exemplo, o predicado que define o conjunto r para o problema de construção de MCs pode ter a seguinte exigência para selecionar os cromossomos mais adaptados: a adaptação do cromossomo selecionado tem que ser maior que a adaptação média da geração corrente, ou seja, a função $sel : \mathbb{P}(P) \times \mathbb{P}(P) \rightarrow \mathbb{P}(P)$, tal que: $sel_x(c_1, \dots, c_n) = \{c_i \mid adapt(c_i) \geq media(P)\}$, onde $media(P)$ é a soma da adaptação de todos os cromossomos de P dividida pelo quantidade de cromossomos de P .

Tomando como exemplo uma população $P = \{c_1, c_2, c_3\}$ formada pelos cromossomos da tabela 4.7, então a média da adaptação dos cromossomos é dada por $media(P_1) = 6$. Logo $sel(P_1) = \{c_1, c_2\}$, pois $adapt(c_1) \geq media(P_1)$ e $adapt(c_2) \geq media(P_1)$.

Selecionar os cromossomos que possuem uma boa adaptação significa que eles contém características que podem ser repassadas, auxiliando na criação de outros cromossomos mais adaptados.

Dados dois cromossomos pai que satisfazem os requisitos de seleção, torna-se indispensável que o GAADT encontre quais os genes dominantes para todas as características existentes. Dessa forma, o GAADT define uma outra função que auxilia o cruzamento denominada fecundação que, a partir de dois cromossomos, retorna todos os genes dominantes.

Definição 4.9 (Fecundação) *A fecundação é uma função fec do seguinte tipo:*

$$fec : C \times C \rightarrow \mathbb{P}(G)$$

$$fec(c_1, c_2) = \{g \mid \forall g_1 \in c_1 \forall g_2 \in c_2 (g = domi(g_1, g_2))\}$$

Lema 4.4.4 $\forall c : C \mid (fec(c, c), c) = C.$

A definição da fecundação não é peculiar a instanciação do problema, por isso para MCs ela permanece a mesma definida por Vieira (2003). Então para o mapa do estudante da figura 4.1 e cromossomos da tabela 4.7, tem-se:

$$fec(c_1, c_2) = \{domi(g_1, g_3), domi(g_1, g_4), domi(g_2, g_3), domi(g_2, g_4)\} = \{g_1\};$$

$$fec(c_2, c_1) = \{domi(g_3, g_1), domi(g_3, g_2), domi(g_4, g_1), domi(g_4, g_2)\} = \{g_3, g_4\};$$

A tabela 4.8 mostra os detalhes da operação $fec(c_1, c_2)$ e $fec(c_2, c_1)$, indicando todas as possibilidades de combinações entre os genes dos cromossomos fornecidos, para que um deles seja classificado como gene dominante.

A possibilidade de comparar características em MCs ocorre quando um dado conceito faz parte dos dois genes (conceitos representado em negrito na tabela 4.8). Conforme a definição 4.6, o gene é considerado dominante se possui grau superior a outro com mesma característica e, havendo igualdade de graus a função prioriza o primeiro gene fornecido.

| $fec(c_1, c_2)$ | $G \times G$ | $grau(g_i)$ | $\mathbb{P}(G)$ |
|------------------|-------------------------------------------------------------------------------------------|-------------|-----------------|
| $domi(g_1, g_3)$ | $g_1 = \langle \text{EVAPORAÇÃO, (TEMPORAL, precede, d), CONDENSARÇÃO} \rangle$ | 2 | g_1 |
| | $g_3 = \langle \text{CONDENSARÇÃO, (AÇÃO, forma, d), NUVEM} \rangle$ | 2 | |
| $domi(g_1, g_4)$ | $g_1 = \langle \text{EVAPORAÇÃO, (TEMPORAL, precede, d), CONDENSARÇÃO} \rangle$ | 2 | g_1 |
| | $g_4 = \langle \text{CONDENSARÇÃO, (TEMPORAL, ocorre antes de, d), PRECIPITAÇÃO} \rangle$ | 2 | |
| $domi(g_2, g_3)$ | $g_2 = \langle \text{EVAPORAÇÃO, (AÇÃO, produz, d), VAPOR DE ÁGUA} \rangle$ | 2 | g_λ |
| | $g_3 = \langle \text{CONDENSARÇÃO, (AÇÃO, forma, d), NUVEM} \rangle$ | 2 | |
| $domi(g_2, g_4)$ | $g_2 = \langle \text{EVAPORAÇÃO, (AÇÃO, produz, d), VAPOR DE ÁGUA} \rangle$ | 2 | g_λ |
| | $g_4 = \langle \text{CONDENSARÇÃO, (TEMPORAL, ocorre antes de, d), PRECIPITAÇÃO} \rangle$ | 2 | |
| $fec(c_2, c_1)$ | $G \times G$ | $grau(g_i)$ | $\mathbb{P}(G)$ |
| $domi(g_3, g_1)$ | $g_3 = \langle \text{CONDENSARÇÃO, (AÇÃO, forma, d), NUVEM} \rangle$ | 2 | g_3 |
| | $g_1 = \langle \text{EVAPORAÇÃO, (TEMPORAL, precede, d), CONDENSARÇÃO} \rangle$ | 2 | |
| $domi(g_3, g_2)$ | $g_3 = \langle \text{CONDENSARÇÃO, (AÇÃO, forma, d), NUVEM} \rangle$ | 2 | g_λ |
| | $g_2 = \langle \text{EVAPORAÇÃO, (AÇÃO, produz, d), VAPOR DE ÁGUA} \rangle$ | 2 | |
| $domi(g_4, g_1)$ | $g_4 = \langle \text{CONDENSARÇÃO, (TEMPORAL, ocorre antes de, d), PRECIPITAÇÃO} \rangle$ | 2 | g_4 |
| | $g_1 = \langle \text{EVAPORAÇÃO, (TEMPORAL, precede, d), CONDENSARÇÃO} \rangle$ | 2 | |
| $domi(g_4, g_2)$ | $g_4 = \langle \text{CONDENSARÇÃO, (TEMPORAL, ocorre antes de, d), PRECIPITAÇÃO} \rangle$ | 2 | g_λ |
| | $g_2 = \langle \text{EVAPORAÇÃO, (AÇÃO, produz, d), VAPOR DE ÁGUA} \rangle$ | 2 | |

Tabela 4.8: Exemplo de Fecundação de Cromossomos em MCs

Com a análise de quais genes são dominantes chegou-se no seguinte resultado: $fec(c_1, c_2) = \{g_1\}$ e $fec(c_2, c_1) = \{g_3, g_4\}$

No cruzamento os cromossomos aptos são representados pelo conjunto *MACHO* e *FEMEA*, tal que $MACHO = sel(P_1, M)$ e $FEMEA = sel(P_1, F)$, onde:

- os cromossomos adaptados ao ambiente são agrupados na população P_1 ;
- M e F são dois predicados sobre o tipo população que obedecem o conjunto de requisitos do ambiente Rq , escritos em uma linguagem de primeira ordem;
- se $M \cap F = \emptyset$ a reprodução é dita sexuada;
- se $M = F$ a reprodução é dita assexuada;
- se $M \cap F \neq \emptyset$ e $M \neq F$ a reprodução é dita mista.

Definição 4.10 (Cruzamento) *O cruzamento é uma função cruz do seguinte tipo:*

$$cruz : MACHO \times FEMEA \rightarrow P$$

$$cruz(c_1, c_2) = \{c | c \subseteq fec(c_1, c_2)\}$$

Considerando as informações referentes a adaptação e fecundação, apresentadas respectivamente nas tabelas 4.7 e 4.8, a seguir é mostrado um exemplo da representação do cruzamento para mapas conceituais.

- Uma subpopulação é selecionada, tal que $P_1 = \{c_1, c_2\}$;
- Os predicados são definidos com uma reprodução assexuada: $M = F$. Dessa forma, o GAADT realiza a fecundação de todos com todos, verificando o conjunto de genes dominantes entre todos os genes dos cromossomos fornecidos;
- Observa-se na tabela 4.9 que $fec(c_1, c_2) \neq fec(c_2, c_1)$, isso faz valer a escolha de reprodução assexuada para geração de mapas conceituais;

| | | |
|------------------------------------|-------------------------------------------------------------------------------------------------------|--------------------------------|
| <i>MACHO</i> | $sel(P_1, M) = \{c_1\}$ | $sel(P_1, M) = \{c_2\}$ |
| <i>FEMEA</i> | $sel(P_1, F) = \{c_2\}$ | $sel(P_1, F) = \{c_1\}$ |
| $(fec(MACHO, FEMEA))$ | $fec(c_1, c_2) = \{g_1\}$ | $fec(c_2, c_1) = \{g_3, g_4\}$ |
| $fec(c_1, c_2) \neq fec(c_2, c_1)$ | $cruz(c_1, c_2) = \{c \subseteq (fec(c_1, c_2) \wedge fec(c_2, c_1)) \mid adapt(c) \geq adapt_m(P)\}$ | |

Tabela 4.9: Exemplo Cruzamento

- $\therefore cruz(c_1, c_2) = \{c = \{g_1\}, c = \{g_3\}, c = \{g_4\}, c_4 = \{g_1, g_3\}, c = \{g_1, g_4\}, c = \{g_3, g_4\}, c = \{g_1, g_3, g_4\}\}$;
- O cruzamento pode obter um cromossomo cuja adaptação é inferior as já existentes, porém somente cromossomos mais adaptados são realmente gerados. A figura 4.5 mostra a população resultante da operação $cruz(c_1, c_2)$.

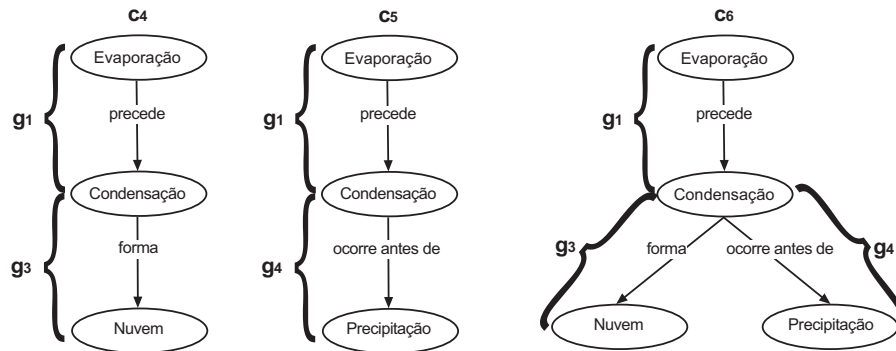


Figura 4.5: Resultado do Cruzamento entre c_1 e c_2

- É importante enfatizar que os cromossomos resultantes do cruzamento são validados pelo *AFC*. Dentre os axiomas há a determinação que o MC deve ser conexo e que caso, as frases de enlace sejam derivadas somente de uma diferenciação progressiva, então o MC deve ser acíclico.

O próximo operador definido pelo GAADT é a mutação, a qual é composta de três funções: inserção (*ins*), supressão (*del*) e troca (*troc*), onde o cromossomo final apresenta parte dos genes do cromossomo original. A seguir a definição de cada função e o exemplo com MCs.

Definição 4.11 (Inserção) A inserção é uma função *ins* do seguinte tipo:

$$ins : C \times \mathbb{P}(G) \rightarrow C$$

$$ins(c, G_1) = \begin{cases} c \cup G_1 & \text{se } (c \cup G_1) \in AFC, \\ c & \text{c.c.} \end{cases}$$

Para realizar a mutação de genes, o GAADT seleciona aqueles que se encontram abaixo da média. Para o exemplo da tabela 4.7, tem-se apenas um gene abaixo da média formando a subpopulação $P_2 = \{c_3\}$. A figura 4.6 mostra um aplicação de inserções de genes G_i no cromossomo c_3 .

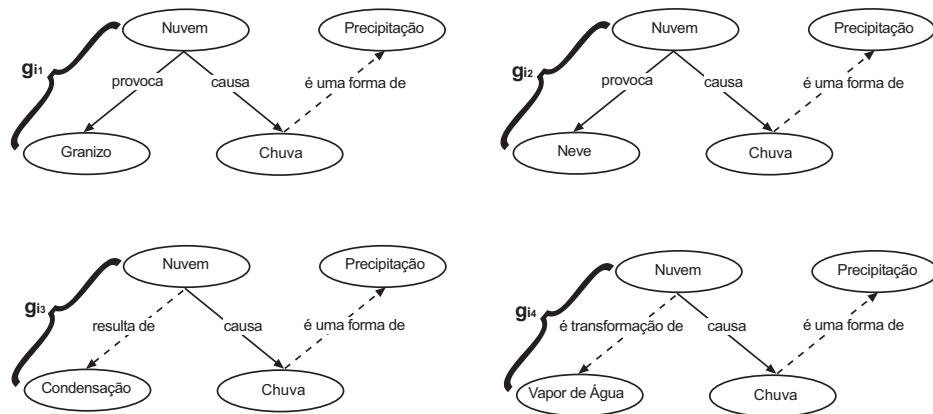


Figura 4.6: Exemplo Inserção de Gene

A operação de supressão *del* remove um conjunto de genes do cromossomo de origem.

Definição 4.12 (Supressão) A supressão é uma função *del* do seguinte tipo:

$$del : C \times \mathbb{P}(G) \rightarrow C$$

$$del(c, G_1) = \begin{cases} c - G_1 & \text{se } (c - G_1) \in AFC, \\ c & \text{c.c.} \end{cases}$$

Alguns genes podem apresentar características indesejadas. No caso de comparação de mapas conceituais um exemplo seria a criação de um gene que contenha um conceito que não se apresente no mapa do aluno, ou seja, a presença de uma característica que não condiz com a realidade.

Para aproximar o resultado a característica indesejada pode ser suprimida, como no caso de $g_6 = \langle \text{CHUVA}, (\text{CARACTERÍSTICA}, \text{é uma forma de}, r), \text{PRECIPITAÇÃO} \rangle$ (figura 4.7), tentando assim eliminar uma opção de gene que contenha o conceito chuva, o qual não está presente na figura 4.1.

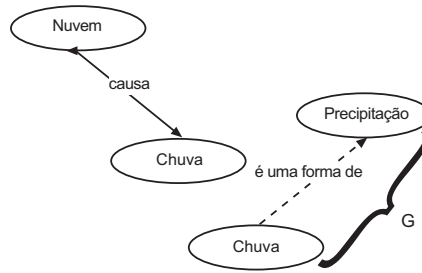


Figura 4.7: Exemplo Supressão de Gene

A operação de troca *troc* remove um conjunto de genes do cromossomo de origem e lhe adiciona outro conjunto de genes.

Definição 4.13 (Troca) A troca é uma função *troc* do seguinte tipo:

$$troc : C \times \mathbb{P}(G) \times \mathbb{P}(G) \rightarrow C$$

$$troc(c, G_1, G_2) = \begin{cases} (c \cup G_1) - G_2 & \text{se } (c \cup G_1) \in AFC \wedge ((c \cup G_1) - G_2) \in AFC, \\ c \cup G_1 & \text{se } (c \cup G_1) \in AFC \wedge ((c \cup G_1) - G_2) \notin AFC, \\ c - G_2 & \text{se } (c \cup G_1) \notin AFC \wedge (c - G_2) \in AFC, \\ c & \text{se } (c \cup G_1) \notin AFC \wedge (c - G_2) \notin AFC. \end{cases}$$

Ações da função de inserção e supressão podem ser vistas como casos particulares da ação da função de troca. Conforme estabelece o seguinte corolário:

Corolário 4.4.1 $\forall c : C; G_1, G_2 : \mathbb{P}(G) \mid troc(c, G_1, G_2) = del(ins(c, G_1), G_2)$

A troca é uma forma de adaptação do resultado, sendo uma mesclagem dos operadores supressão e inserção, respectivamente. Dadas as possíveis inserções em c_3 mostradas acima, a figura 4.8 mostra exemplos de troca cujos cromossomos resultantes têm adaptação superior ao cromossomo ancestral.

Definição 4.14 (Mutaç o) A muta o   um predicado $mut \subseteq \mathbb{P}(P)$, tal que:

$$mut(c_1) = \{c_2 \mid \exists G_1, G_2 : \mathbb{P}(G) ((\#G_1 \leq \#c_1 \div 2) \wedge (\#G_2 \leq \#c_1 \div 2) \wedge (troc(c_1, G_1, G_2) = c_2) \wedge (adapt(c_2) > adapt(c_1)))\}$$

A restrição que c_2 seja maior que c_1 garante que todo cromossomo mutante é mais adaptado do que o cromossomo original. Os cromossomos da figura 4.8 são exemplos válidos de mutação em MCs, pois $adapt(c_7) = 6$ e $adapt(c_8) = 6$, indicando que houve adaptação destes cromossomos ao ambiente frente às características do cromossomo original c_3 .

O GAADT define que a alteração do número de genes do cromossomo no cromossomo mutante é limitada em cinquenta por cento do tamanho inicial, para que as mutações ocorridas em um cromossomo não sejam bruscas ao ponto de repelir os demais cromossomos da sua espécie.

4.5 Ambiente

O GAADT opera sobre populações de cromossomos que evoluem de acordo com as características de um ambiente A . Um ambiente A é uma 8-tupla $\langle P, \mathbb{P}(P), Rq, AFG, AGC, Tx, \Sigma, P_0 \rangle$, onde:

- P é a população,
- $\mathbb{P}(P)$ é o conjunto potência de P ,
- Rq é o conjunto dos requisitos (características expressas através de fórmulas numa linguagem de primeira ordem) do problema que influenciam a genealogia da população P ,
- AFG é o conjunto de axiomas de formação dos genes dos cromossomos da população P ,
- AFC é o conjunto de axiomas de formação dos cromossomos da população P e
- Tx é o conjunto de pares de cromossomos (x, y) , onde x é um cromossomo construído a partir do cromossomo y , pela ação da operação de cruzamento ou mutação, registrando desta forma a genealogia dos cromossomos pertencentes às populações geradas pelo GAADT durante a sua execução,
- Σ é o conjunto de operadores genealógicos que atuam sobre a população P ,
- P_0 é uma sub-população pertencente a $\mathbb{P}(P)$, chamada de população inicial, com no mínimo um cromossomo.

O GAADT submete os cromossomos de uma população à ação dos requisitos do problema Rq . A seguir é apresentado um exemplo do funcionamento do algoritmo para gerar novos cromossomos a partir dos pré-existentes.

4.6 Algoritmo

O GAADT é uma função $GAADT$ que recebe a população P_0 e, depois de submetê-la à simulação de um processo evolutivo, devolve uma população P_t .

Os cromossomos da população P_t são os cromossomos das populações P_0, P_1, \dots, P_{t-1} , cujas características predominaram por satisfazer os requisitos do problema Rq , ou então são novos cromossomos resultantes da ação das operações de cruzamento e mutação sobre os cromossomos da população anterior P_{t-1} . Diz-se então que a população P_t evoluiu da população P_0 .

Os cromossomos das gerações anteriores, cujas características não atenderam os requisitos do problema Rq , podendo ser agrupados numa população de cromossomos "mortos".

Antes da definição do GAADT propriamente dita é necessário definir o critério de preservação sobre a população atual P_t , denominado p_{corte} . Este predicado está entre o conjunto de requisitos Rq e seu objetivo principal é selecionar os cromossomos que são automaticamente repassados para a próxima geração. Para o exemplo de mapas conceituais considera-se como ponto de corte a adaptação média da população atual.

Existem dois critérios de parada para a função GAADT. O primeiro é utilizado quando é de conhecimento prévio o valor de adaptação a ser encontrado, tal que os cromossomos da população são considerados satisfatórios. O segundo critério diz respeito ao número máximo de iterações desejadas. Ambos os critérios são requisitos do problema e devem ser definidos durante a instanciação do ambiente.

Definição 4.15 (GAADT) *O GAADT é uma função GAADT do seguinte tipo:*

$$GAADT : A \rightarrow A$$

$$GAADT(P_t) = \begin{cases} P_{otm} & \text{se } P_{otm} = \{c | \forall c : P_t(adapt(c) \geq k)\} \neq \emptyset, \\ P_{t+1} & \text{se } t + 2 = T, \\ GAADT(P_{t+1}) & \text{caso contrário.} \end{cases}$$

onde $P_{t+1} = cruz(a, b) \cup mut(c) \cup p_{corte}(P_t)$ com $a, b, c \in P_t$, P_0 é a população inicial considerada, $k \in K$ é um valor imposto pelo ambiente A , como critério de aceitação de cromossomos em P_t que satisfazem o problema e $T \in \mathbb{N}$ é um número dado como critério de satisfação do número de iterações.

As duas primeiras opções de saída da função acima ocorrem em decorrência das condições de parada impostas pelo ambiente. Enquanto as condições de parada não são alcançadas a função é chamada recursivamente criando novas gerações de cromossomos mais adaptados.

Para o exemplo de MCs foi considerada a população inicial $P_0 = \{c_1, c_2, c_3\}$ (figura 4.4). Para melhor entendimento a tabela 4.10 mostra todos os genes que são utilizados nas gerações de novos cromossomos. Optou-se por evitar a inserção de genes que possuam algum conceito que não esteja no MC do estudante.

| | |
|-----------|--------------------------------------------------------------------|
| g_1 | = <EVAPORAÇÃO, (TEMPORAL, precede, d), CONDENSAÇÃO> |
| g_2 | = <EVAPORAÇÃO, (AÇÃO, produz, d), VAPOR DE ÁGUA> |
| g_3 | = <CONDENSAÇÃO, (AÇÃO, forma, d), NUVEM> |
| g_4 | = <CONDENSAÇÃO, (TEMPORAL, ocorre antes de, d), PRECIPITAÇÃO> |
| g_5 | = <NUVEM, (AÇÃO, causa, d), CHUVA> |
| g_6 | = <CHUVA, (CARACTERÍSTICA, é uma forma de, r), PRECIPITAÇÃO> |
| g_{i_1} | = <NUVEM, (AÇÃO, provoca, d), GRANIZO> |
| g_{i_3} | = <NUVEM, (AÇÃO, resulta de, r), CONDENSAÇÃO> |
| g_{i_5} | = <GRANIZO, (AÇÃO, é causado por, r), NUVEM> |
| g_{i_6} | = <GRANIZO, (CARACTERÍSTICA, é uma forma de, r), PRECIPITAÇÃO> |
| g_{i_7} | = <PRECIPITAÇÃO, (CARACTERÍSTICA, pode aparecer como, r), GRANIZO> |

Tabela 4.10: Genes Utilizados no Algoritmo

Por ser um exemplo textual, o número máximo de iterações será dado por $T = 5$. Conforme o questionamento de não haver um mapa conceitual ideal, então será definido $K = 0$. Isto só ocorrerá quando a busca encontrar todos cromossomos com nenhum dos seus conceitos pertencentes ao conjunto de conceitos do mapa fornecido pelo estudante, significando que a população não poderá convergir a uma solução adequada devido ao mapa do estudante não se assemelhar à ontologia.

Dessa forma, se o mapa de entrada for adequado o único critério de parada alcançado será o número máximo de iterações.

O estado do sistema é composto pelas seguintes propriedades:

- P_{atual} : população de cromossomos vivos;
- P_{cruz} : população de cromossomos gerados por cruzamento;
- P_{mut} : população de cromossomos gerados por mutação;
- T_x : relação taxonômica dos cromossomos da população atual;
- t : contador do número de iterações.
- $MACHO$ e $FEMEA$: propriedades invariantes do sistema, sendo subconjuntos de P_{atual} .

Quando o sistema é inicializado tem-se: $P_0 = P_{atual} = \{c_1, c_2, c_3\}$; $T_x = \emptyset$; $t = 0$ para indicar que nenhuma iteração da função GAADT foi realizada; $P_{cruz} = \emptyset$ e $P_{mut} = \emptyset$ indicando que chamadas das funções *cruz* e *mut* ainda não foram executadas. Contudo, já é feita uma pré-seleção de $MACHO = sel(p_{corte}(P_0), M)$ e $FEMEA = sel(p_{corte}(P_0), F)$.

A tabela 4.11 mostra iterações do GAADT para MCs, onde das quatro colunas a leitura deve ser orientada duas a duas na vertical. O primeiro conjunto de células indica a iteração do GAADT sobre P_0 .

A chamada da função *cruz* especifica a construção de novos cromossomos a partir dos cromossomos da população atual que satisfazem p_{corte} . Estes cromossomos pai são selecionados dos conjuntos *MACHO* e *FEMEA* de tal forma que a reprodução seja assexuada. Por sua vez, os cromossomos que não satisfazem p_{corte} são submetidos a função *mut* antes de serem transferidos para a população de cromossomos mortos.

Como $t + 2 \neq T$ há uma chamada recursiva da função GAADT, onde a população atual é atualizada e $t = 1$ para indicar que já houve uma iteração do algoritmo.

Está implícito no decorrer das gerações da tabela 4.11 algumas fecundações e mutações que não vingam, pois os cromossomos descendentes são equivalentes a outros pré-existentes.

A tabela 4.12 é uma continuação da tabela 4.11, onde são mostrados a população de cromossomos gerados por cruzamento na primeira coluna e a relação taxonômica na segunda, onde os conjuntos P_{cruz} e T_x são referentes a última iteração.

Deve ser destacado que a partir da segunda iteração algumas mutação nos cromossomos $\{c_2, c_4, c_5, c_{12}, c_{13}\}$ e $\{c_9, c_{10}\}$, os quais não satisfazem mais os requisitos, são obtidas somente pela aplicação da função de inserção. A inserção de um gene tornaria a mutação válida, porém a busca por genes a serem suprimidos não retornou uma opção válida já que resultaria num cromossomo mutante cuja adaptação não superaria ao do cromossomo que o originou. Por isso a mutação foi empregada apenas com a função *ins*.

Finalmente, chega-se a condição de parada $t + 2 = 5$ e o sistema retorna a população de cromossomos vivos $P_{atual} = \{\{c_6, c_{11}, c_{14}, c_{15}, \dots, c_{25}\} \cup \{c_{26}, c_{27}, \dots, c_{77}\} \cup \{c_{78}, c_{79}, \dots, c_{82}\}\}$.

Pode-se observar que o GAADT encontrou cromossomos próximos ao mapa fornecido, como c_{30}, c_{31}, c_{50} e c_{51} . Contudo, mesmo evitando a inserção de certos conceitos, o mapa do estudante não apresenta conceitos importantes sobre o domínio ciclo da água, cabendo ao professor fazer uma avaliação mais crítica.

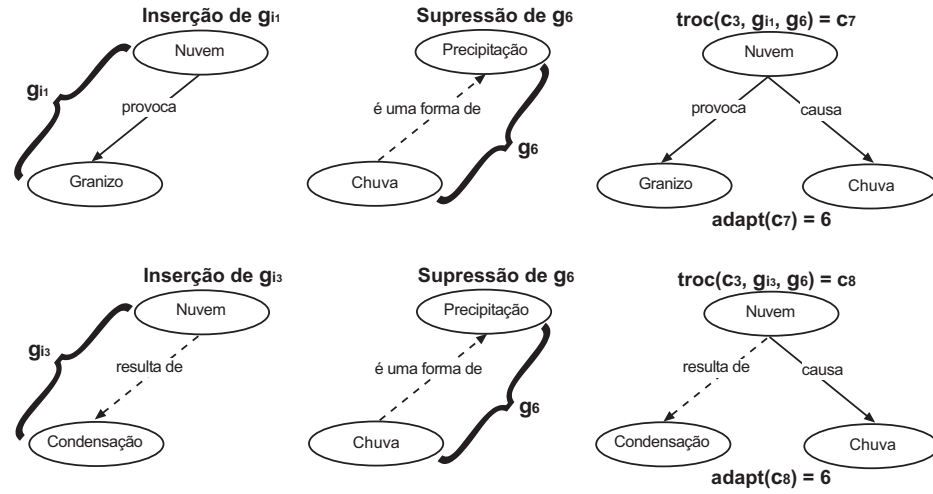


Figura 4.8: Exemplo Troca de Genes

| | | | |
|---------------------------|--------------------------------------------------------------------------------------------|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Patual | $P_0 = \{c_1, c_2, c_3\}$ | Patual | $\{c_2, c_4, c_5, c_6, c_9, c_{10}, c_{11}, c_{12}, c_{13}\}$ |
| esvazia() | $P_{cruz} = \emptyset; P_{mut} = \emptyset$ | esvazia() | $P_{cruz} = \emptyset; P_{mut} = \emptyset$ |
| t | 0 | t | 2 |
| MACHO | $\{c_1, c_2\}$ | MACHO | $\{c_6, c_9, c_{10}, c_{11}\}$ |
| FEMEA | $\{c_1, c_2\}$ | FEMEA | $\{c_6, c_9, c_{10}, c_{11}\}$ |
| $\neg p_{corte}$ | $\{c_3\}$ | $\neg p_{corte}$ | $\{c_2, c_4, c_5, c_{12}, c_{13}\}$ |
| P_{cruz} | $\{c_4 = \{g_1, g_3\}, c_5 = \{g_1, g_4\}, c_6 = \{g_1, g_3, g_4\}\}$ | P_{cruz} | \emptyset |
| T_x | $T_x \cup \{(c_4, \{c_1, c_2\}), (c_5, \{c_1, c_2\}), (c_8, \{c_1, c_2\})\}$ | T_x | T_x |
| P_{mut} | $\{c_7 = \{g_{i_1}, g_6\}, c_8 = \{g_{i_3}, g_6\}\}$ | P_{mut} | $\{c_{14} = \{g_3, g_4, g_{i_1}\}, c_{15} = \{g_3, g_4, g_{i_5}\}, c_{16} = \{g_3, g_4, g_{i_6}\}, c_{17} = \{g_3, g_4, g_{i_7}\}, c_{18} = \{g_1, g_3, g_{i_1}\}, c_{19} = \{g_1, g_4, g_{i_6}\}, c_{20} = \{g_1, g_4, g_{i_7}\}, c_{21} = \{g_3, g_{i_1}, g_{i_6}\}, c_{22} = \{g_3, g_{i_1}, g_{i_7}\}, c_{23} = \{g_{i_1}, g_{i_3}, g_1\}, c_{24} = \{g_{i_1}, g_{i_3}, g_4\}, c_{25} = \{g_{i_1}, g_{i_3}, g_{i_6}\}\}$ |
| T_x | $T_x \cup \{(c_7, c_3), (c_8, c_3)\}$ | T_x | $T_x \cup \{(c_{14}, c_2), (c_{15}, c_2), (c_{16}, c_2), (c_{17}, c_2), (c_{18}, c_4), (c_{19}, c_5), (c_{20}, c_5), (c_{21}, c_{12}), (c_{22}, c_{12}), (c_{23}, c_{13}), (c_{24}, c_{13}), (c_{25}, c_{13})\}$ |
| P_{mortos} | $P_{mortos} \cup \{c_3\}$ | P_{mortos} | $P_{mortos} \cup \{c_2, c_4, c_5, c_{12}, c_{13}\}$ |
| Patual | $\{c_1, c_2, c_4, c_5, c_6, c_7, c_8\}$ | Patual | $\{c_6, c_9, c_{10}, c_{11}, c_{14}, c_{15}, c_{16}, c_{17}, c_{18}, c_{19}, c_{20}, c_{21}, c_{22}, c_{23}, c_{24}, c_{25}\}$ |
| esvazia() | $P_{cruz} = \emptyset; P_{mut} = \emptyset$ | esvazia() | $P_{cruz} = \emptyset; P_{mut} = \emptyset$ |
| t | 1 | t | 3 |
| MACHO | $\{c_2, c_4, c_5, c_6\}$ | MACHO | $\{c_6, c_{11}, c_{14}, c_{15}, c_{16}, c_{17}, c_{18}, c_{19}, c_{20}, c_{21}, c_{22}, c_{23}, c_{24}, c_{25}\}$ |
| FEMEA | $\{c_2, c_4, c_5, c_6\}$ | FEMEA | $\{c_6, c_{11}, c_{14}, c_{15}, c_{16}, c_{17}, c_{18}, c_{19}, c_{20}, c_{21}, c_{22}, c_{23}, c_{24}, c_{25}\}$ |
| $\neg p_{corte}$ | $\{c_1, c_7, c_8\}$ | $\neg p_{corte}$ | $\{c_9, c_{10}\}$ |
| P_{cruz} | $\{c_9 = \{g_1, g_2, g_3\}, c_{10} = \{g_1, g_2, g_4\}, c_{11} = \{g_1, g_2, g_3, g_4\}\}$ | P_{cruz} | (Ver Tabela 4.12) |
| T_x | $T_x \cup \{(c_9, \{c_2, c_4\}), (c_{10}, \{c_2, c_5\}), (c_{11}, \{c_2, c_6\})\}$ | T_x | (Ver Tabela 4.12) |
| P_{mut} | $\{c_{12} = \{g_{i_1}, g_3\}, c_{13} = \{g_{i_3}, g_{i_1}\}\}$ | P_{mut} | $\{c_{78} = \{g_1, g_2, g_3, g_{i_1}\}, c_{79} = \{g_1, g_2, g_3, g_{i_5}\}, c_{80} = \{g_1, g_3, g_{i_5}\}, c_{81} = \{g_1, g_2, g_4, g_{i_6}\}, c_{82} = \{g_1, g_2, g_4, g_{i_7}\}\}$ |
| T_x | $T_x \cup \{(c_{12}, c_7), (c_{13}, c_8)\}$ | T_x | $T_x \cup \{(c_{78}, c_9), (c_{79}, c_9), (c_{80}, c_9), (c_{81}, c_{10}), (c_{82}, c_{10})\}$ |
| P_{mortos} | $P_{mortos} \cup \{c_1, c_7, c_8\}$ | P_{mortos} | $P_{mortos} \cup \{c_9, c_{10}\}$ |

Tabela 4.11: Iterações do GAADT

| P_{cruz} | T_x |
|--------------------------------------------------------------|--------------------------------|
| { | $T_x \cup \{$ |
| $c_{26} = \{g_1, g_3, g_4, g_{i_1}\}$ | $(c_{26}, \{c_6, c_{14}\})$ |
| $c_{27} = \{g_1, g_3, g_4, g_{i_5}\}$ | $(c_{27}, \{c_6, c_{15}\})$ |
| $c_{28} = \{g_1, g_3, g_4, g_{i_6}\}$ | $(c_{28}, \{c_6, c_{16}\})$ |
| $c_{29} = \{g_1, g_3, g_4, g_{i_7}\}$ | $(c_{29}, \{c_6, c_{17}\})$ |
| $c_{30} = \{g_1, g_3, g_4, g_{i_1}, g_{i_6}\}$ | $(c_{30}, \{c_6, c_{21}\})$ |
| $c_{31} = \{g_1, g_3, g_4, g_{i_1}, g_{i_7}\}$ | $(c_{31}, \{c_6, c_{22}\})$ |
| $c_{32} = \{g_1, g_3, g_4, g_{i_1}, g_{i_3}\}$ | $(c_{32}, \{c_6, c_{23}\})$ |
| $c_{33} = \{g_1, g_3, g_4, g_{i_1}, g_{i_3}, g_{i_6}\}$ | $(c_{33}, \{c_6, c_{25}\})$ |
| $c_{34} = \{g_1, g_2, g_3, g_4, g_{i_1}\}$ | $(c_{34}, \{c_{11}, c_{14}\})$ |
| $c_{35} = \{g_1, g_2, g_3, g_4, g_{i_5}\}$ | $(c_{35}, \{c_{11}, c_{15}\})$ |
| $c_{36} = \{g_1, g_2, g_3, g_4, g_{i_6}\}$ | $(c_{36}, \{c_{11}, c_{16}\})$ |
| $c_{37} = \{g_1, g_2, g_3, g_4, g_{i_7}\}$ | $(c_{37}, \{c_{11}, c_{17}\})$ |
| $c_{38} = \{g_1, g_2, g_3, g_4, g_{i_1}, g_{i_6}\}$ | $(c_{38}, \{c_{11}, c_{21}\})$ |
| $c_{39} = \{g_1, g_2, g_3, g_4, g_{i_1}, g_{i_7}\}$ | $(c_{39}, \{c_{11}, c_{22}\})$ |
| $c_{40} = \{g_1, g_2, g_3, g_4, g_{i_1}, g_{i_3}\}$ | $(c_{40}, \{c_{11}, c_{23}\})$ |
| $c_{41} = \{g_1, g_2, g_3, g_4, g_{i_1}, g_{i_3}, g_{i_6}\}$ | $(c_{41}, \{c_{11}, c_{25}\})$ |
| $c_{42} = \{g_3, g_4, g_{i_1}, g_{i_5}\}$ | $(c_{42}, \{c_{14}, c_{15}\})$ |
| $c_{43} = \{g_3, g_4, g_{i_1}, g_{i_6}\}$ | $(c_{43}, \{c_{14}, c_{16}\})$ |
| $c_{44} = \{g_3, g_4, g_{i_1}, g_{i_7}\}$ | $(c_{44}, \{c_{14}, c_{17}\})$ |
| $c_{45} = \{g_3, g_4, g_{i_1}, g_{i_3}\}$ | $(c_{45}, \{c_{14}, c_{24}\})$ |
| $c_{46} = \{g_3, g_4, g_{i_1}, g_{i_3}, g_{i_6}\}$ | $(c_{46}, \{c_{14}, c_{25}\})$ |
| $c_{47} = \{g_3, g_4, g_{i_5}, g_{i_6}\}$ | $(c_{47}, \{c_{15}, c_{16}\})$ |
| $c_{48} = \{g_3, g_4, g_{i_5}, g_{i_7}\}$ | $(c_{48}, \{c_{15}, c_{17}\})$ |
| $c_{49} = \{g_1, g_3, g_4, g_{i_1}, g_{i_5}\}$ | $(c_{49}, \{c_{15}, c_{18}\})$ |
| $c_{50} = \{g_1, g_3, g_4, g_{i_5}, g_{i_6}\}$ | $(c_{50}, \{c_{15}, c_{19}\})$ |
| $c_{51} = \{g_1, g_3, g_4, g_{i_5}, g_{i_7}\}$ | $(c_{51}, \{c_{15}, c_{20}\})$ |
| $c_{52} = \{g_3, g_4, g_{i_1}, g_{i_5}, g_{i_6}\}$ | $(c_{52}, \{c_{15}, c_{21}\})$ |
| $c_{53} = \{g_3, g_4, g_{i_1}, g_{i_5}, g_{i_7}\}$ | $(c_{53}, \{c_{15}, c_{22}\})$ |
| $c_{54} = \{g_1, g_3, g_4, g_{i_1}, g_{i_3}, g_{i_5}\}$ | $(c_{54}, \{c_{15}, c_{23}\})$ |
| $c_{55} = \{g_3, g_4, g_{i_1}, g_{i_3}, g_{i_5}\}$ | $(c_{55}, \{c_{15}, c_{24}\})$ |
| $c_{56} = \{g_3, g_4, g_{i_1}, g_{i_3}, g_{i_5}, g_{i_6}\}$ | $(c_{56}, \{c_{15}, c_{25}\})$ |
| $c_{57} = \{g_3, g_4, g_{i_6}, g_{i_7}\}$ | $(c_{57}, \{c_{16}, c_{17}\})$ |
| $c_{58} = \{g_1, g_3, g_4, g_{i_6}, g_{i_7}\}$ | $(c_{58}, \{c_{16}, c_{20}\})$ |
| $c_{59} = \{g_3, g_4, g_{i_1}, g_{i_6}, g_{i_7}\}$ | $(c_{59}, \{c_{16}, c_{22}\})$ |
| $c_{60} = \{g_1, g_3, g_4, g_{i_1}, g_{i_3}, g_{i_7}\}$ | $(c_{60}, \{c_{17}, c_{23}\})$ |
| $c_{61} = \{g_3, g_4, g_{i_1}, g_{i_3}, g_{i_7}\}$ | $(c_{61}, \{c_{17}, c_{24}\})$ |
| $c_{62} = \{g_3, g_4, g_{i_1}, g_{i_6}, g_{i_7}\}$ | $(c_{62}, \{c_{17}, c_{25}\})$ |
| $c_{63} = \{g_1, g_3, g_{i_1}, g_{i_6}\}$ | $(c_{63}, \{c_{18}, c_{21}\})$ |
| $c_{64} = \{g_1, g_3, g_{i_1}, g_{i_7}\}$ | $(c_{64}, \{c_{18}, c_{22}\})$ |
| $c_{65} = \{g_1, g_3, g_{i_1}, g_{i_3}\}$ | $(c_{65}, \{c_{18}, c_{23}\})$ |
| $c_{66} = \{g_1, g_3, g_{i_1}, g_{i_3}, g_{i_6}\}$ | $(c_{66}, \{c_{18}, c_{25}\})$ |
| $c_{67} = \{g_1, g_4, g_{i_6}, g_{i_7}\}$ | $(c_{67}, \{c_{19}, c_{20}\})$ |
| $c_{68} = \{g_1, g_4, g_{i_1}, g_{i_3}, g_{i_6}\}$ | $(c_{68}, \{c_{19}, c_{23}\})$ |
| $c_{69} = \{g_1, g_3, g_4, g_{i_1}, g_{i_6}, g_{i_7}\}$ | $(c_{69}, \{c_{20}, c_{21}\})$ |
| $c_{70} = \{g_1, g_4, g_{i_1}, g_{i_3}, g_{i_7}\}$ | $(c_{70}, \{c_{20}, c_{23}\})$ |
| $c_{71} = \{g_3, g_{i_1}, g_{i_6}, g_{i_7}\}$ | $(c_{71}, \{c_{21}, c_{22}\})$ |
| $c_{72} = \{g_3, g_{i_1}, g_{i_3}, g_{i_6}\}$ | $(c_{72}, \{c_{21}, c_{25}\})$ |
| $c_{73} = \{g_1, g_3, g_{i_1}, g_{i_3}, g_{i_7}\}$ | $(c_{73}, \{c_{22}, c_{23}\})$ |
| $c_{74} = \{g_3, g_{i_1}, g_{i_3}, g_{i_6}, g_{i_7}\}$ | $(c_{74}, \{c_{22}, c_{25}\})$ |
| $c_{75} = \{g_1, g_4, g_{i_1}, g_{i_3}\}$ | $(c_{75}, \{c_{23}, c_{24}\})$ |
| $c_{76} = \{g_1, g_{i_1}, g_{i_3}, g_{i_6}\}$ | $(c_{76}, \{c_{23}, c_{25}\})$ |
| $c_{77} = \{g_4, g_{i_1}, g_{i_3}, g_{i_6}\}$ | $(c_{77}, \{c_{24}, c_{25}\})$ |
| } | } |

Tabela 4.12: População gerada pelo cruzamento

Capítulo 5

GAADT para o Problema de Alinhamento Múltiplo de Proteínas

5.1 Introdução

Com a apresentação do GAADT no capítulo anterior pode-se chegar as seguintes observações:

- o GAADT trabalha sobre um ambiente o qual possui uma estrutura denominada população;
- para todo problema existe uma função que mapeia os possíveis resultados do problema numa estrutura de dados denominada cromossomo;
- as características do ambiente são vistas como propriedades iniciais do período evolutivo, no qual os cromossomos da população atual vão sofrer a ação dos operadores genéticos. Estas características devem ser obedecidas no decorrer das gerações;
- os cromossomos filhos oriundos do cruzamento contém apenas as características responsáveis pela adaptação dos cromossomos pais ao ambiente, as quais são chamadas genes dominantes;
- os cromossomos que não fazem mais parte dos requisitos ambientais são transferidos para uma população de cromossomos mortos;
- antes de desaparecerem da população, os cromossomos não adaptados ao ambiente atual são submetidos à ação do operador genético de mutação como uma forma de garantir a parte adaptada destes cromossomos nas próximas gerações;

- o resultado do problema é obtido pelo cromossomo mais adaptado ao ambiente no ponto onde há estagnação do GAADT.

Com o intuito de resolver o problema de alinhamento múltiplo de proteína, este capítulo propõe uma descrição formal do problema como uma instanciação do GAADT. Uma versão preliminar desta instanciação pode ser vista no artigo de (Santos et al. 2006).

A seção a seguir apresenta uma breve descrição de como seqüências de proteínas são dispostas para formar o alinhamento. Em seguida os detalhes da instanciação do GAADT são apresentados.

5.2 Descrição Formal do Problema: Alinhamento Múltiplo de Proteína

Para tratar o problema de alinhamento múltiplo de proteínas, m seqüências de aminoácidos são agrupadas no conjunto $S_0 = \{s_1, s_2, \dots, s_i, \dots\}$, com $1 \leq i \leq m$. A quantidade de aminoácidos de cada seqüência é dada por $length(s_i)$, com $1 \leq length(s_i) \leq n$, onde n é o tamanho da maior seqüência.

O alinhamento múltiplo de S_0 é o conjunto $R_0 = \{r_1, r_2, \dots, r_i, \dots\}$ com as seguintes propriedades:

- a disposição de S_0 ao alinhamento múltiplo é tal que $\forall i (s_i \cup \{-\}) = r_i$, com $1 \leq i \leq m$ e o símbolo $-$ é denominado *gap*;
- $L = length(r_i)$ é o mesmo em todas as seqüências de R_0 , com $1 \leq j \leq L$, onde j representa cada coluna ou posição no alinhamento;
- $score(R_0)$ tem que ser maximizado.

Pode-se perceber que em cada seqüência são inseridos *gaps*, uma tarefa imprescindível no alinhamento múltiplo. Porém, qualquer que seja a operação realizada em S_0 para sobrepor aminoácidos semelhantes, quando aplicada a operação *rem* que remove *gaps*, esta deve produzir as seqüências de aminoácidos originais conforme a ordem apresentada no conjunto S_0 .

Definição 5.1 (Remoção de Gaps) *Remoção de gaps é uma função recursiva rem que remove todos os gaps:*

$$rem : (A_{prot} \cup \{-\})^* \rightarrow A_{prot}^*$$

$$rem(\langle a_1, \dots, a_\omega, \dots, a_L \rangle) = \begin{cases} rem(\langle a_1, \dots, a_{\omega-1}, a_{\omega+1}, \dots, a_L \rangle) & \text{se } \omega \neq 0 \\ \langle a_1, a_2, \dots, a_n \rangle & \text{c.c.} \end{cases} \quad (5.1)$$

onde a_ω é o primeiro gap encontrado na seqüência de aminoácidos $\langle a_1, a_2, \dots, a_n \rangle$.

Definição 5.2 (Alinhamento) Alinhamento múltiplo de proteína é uma função *align* do seguinte tipo: $align : \wp(A_{prot}) \rightarrow \wp((A_{prot} \cup \{-\})^*)$ tal que:

$$align(S_0 = \{s_1, s_2, \dots, s_m\}) = \{R_0 = \{r_1, r_2, \dots, r_m\} \mid \max(score(R_0))\} \quad (5.2)$$

onde $\wp(X^*)$ é o conjunto potência do conjunto de seqüências de tamanhos quaisquer, construído com elementos de X .

5.3 Uma Instanciação do GAADT para Alinhamento Múltiplo

5.3.1 Elementos Básicos

No GAADT os indivíduos são representados por cromossomos que indicam um possível resultado do problema. Para o problema aqui analisado, as soluções têm os aminoácidos como unidades atômicas.

Definição 5.3 (Base) Base é o conjunto B , tal que:

$$B = (A_{prot} \cup \{-\}) \times \mathbb{N}$$

onde $A_{prot} = \{D, E, A, R, N, C, F, G, Q, H, I, L, K, M, P, S, Y, T, W, V\}$ e o símbolo $\langle -, 0 \rangle$ é chamado de base inócua.

As bases são agrupadas para formar as características (genes) dos indivíduos. Dependendo destas características é possível classificar dois indivíduos de acordo com o grupo taxonômico (espécies, famílias). Isto provém do fato de existir uma "lei de formação" para indicar como o gene pode ser instanciado para representar uma característica. Assim, um conjunto de axiomas de formação de gene (AFG) é definido para determinar como um tipo gene pode se comportar.

Definição 5.4 (Gene) O conjunto gene é definido por $G = \{g_j = \langle b_{1,j}, b_{2,j}, \dots, b_{i,j} \rangle \mid b_{1,j}, b_{2,j}, \dots, b_{i,j} \in B\}$ que satisfaz os seguintes axiomas:

$$AFG_1 = \{\forall i \in \{1, \dots, m\} \forall j \in \{1, \dots, L\} \text{primeiro}(b_{i,j}) \in \{A_{prot} \cup \{-\}\}\} \quad (5.3)$$

$$AFG_2 = \{\forall i \in \{1, \dots, m\} \forall j \in \{1, \dots, L\} \text{segundo}(b_{i,j}) \in \mathbb{N}\} \quad (5.4)$$

$$AFG_3 = \{\forall i \in \{1, \dots, m\} \forall j = 1 (\text{segundo}(b_{i,j}) = 0) \vee (\text{segundo}(b_{i,j}) = 1)\} \quad (5.5)$$

$$AFG_4 = \{\forall i \in \{1, \dots, m\} \forall j \in \{2, \dots, L\} \forall b_{i,j} \neq \text{'-'} \text{segundo}(b_{i,j}) = \max(\{\text{segundo}(b_{i,k}) \mid k \in \{1, \dots, j-1\}\}) + 1\} \quad (5.6)$$

$$AFG_5 = \{\forall i \in \{1, \dots, m\} \forall j \in \{1, \dots, L\} \text{primeiro}(b_{i,j}) \neq \text{'-'} \rightarrow \text{segundo}(b_{i,j}) \neq 0\} \quad (5.7)$$

$$AFG_6 = \{\forall i \in \{1, \dots, m\} \forall j \in \{1, \dots, L\} \text{primeiro}(b_{i,j}) = \text{'-'} \rightarrow \text{segundo}(b_{i,j}) = 0\} \quad (5.8)$$

onde $A_{prot} = \{D, E, A, R, N, C, F, G, Q, H, I, L, K, M, P, S, Y, T, W, V\}$; $\text{primeiro}(b_{i,j})$ e $\text{segundo}(b_{i,j})$ são predicados que informam, respectivamente, o primeiro e o segundo elemento do par ordenado que formam a base $b_{i,j}$.

O gene $g_\lambda = \langle b_{1,j}, b_{2,j}, \dots, b_{i,j} \rangle$ com $b_{1,j} = b_{2,j} = \dots = b_{i,j} = \langle -, 0 \rangle$ é chamado de gene inócuo.

Na definição 5.4 as variáveis i e j são utilizadas para representar a i -ésima linha e a j -ésima coluna do alinhamento múltiplo, respectivamente. Logo, cada gene g_j é uma coluna do alinhamento e sua formação deve obedecer a todos os axiomas de formação de genes. A tabela 5.1 mostra uma explicação textual de cada um destes axiomas.

| | |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AFG_1 | Para toda linha i e para toda coluna j , o primeiro elemento da base $b_{i,j}$ pode ser um símbolo que representa um aminoácido ou um <i>gap</i> . |
| AFG_2 | Para toda linha i e para toda coluna j , o segundo elemento da base $b_{i,j}$ é formado por um número natural. |
| AFG_3 | Para toda linha i e para toda primeira coluna, o valor do segundo elemento da base $b_{i,j}$ é igual a zero ou é igual a um. |
| AFG_4 | Para toda linha i , para as colunas de 2 até L e para toda base $b_{i,j}$ diferente de <i>gap</i> , o valor do segundo elemento da base $b_{i,j}$ é o valor máximo entre os segundos elementos das bases anteriores na mesma linha i acrescido de um. |
| AFG_5 | Para toda linha i e para toda coluna j , se o primeiro elemento for diferente de <i>gap</i> então o segundo elemento é diferente de zero. |
| AFG_6 | Para toda linha i e para toda coluna j , se o primeiro elemento for igual a <i>gap</i> então o segundo elemento é igual a zero. |

Tabela 5.1: Axiomas de formação de genes

A importância do subconjunto formado pelos números naturais (AFG_2) de maneira ordenada (AFG_3) deve-se ao fato de que as operações genéticas que ocorrem para melhorar o alinhamento não devem alterar a ordem que os aminoácidos aparecem na cadeia protéica. Numerando cada aminoácido, é possível perceber, por exemplo, que para seqüências equivalentes, a Leucina de índice 8 em um determinado cromossomo é diferente da Leucina 12 em outro, tornando inviável troca de material genético nestes pontos.

Outro fator que se deve mencionar diz respeito a g_λ . Apesar de uma coluna contendo apenas *gaps* não ser o mais recomendado, algumas operações genéticas podem resultar em g_λ , porém, como se trata de um gene inócuo, a sua supressão não altera a representação do alinhamento resultante.

Exemplo 9 (Gene) Considere o conjunto $S_0 = \{s_1, s_2, s_3\}$. Um possível conjunto R_0 é formado por um elemento de $\wp((A_{prot} \cup \{-\})^*)$ (tabela 5.2).

| | | | |
|-------|-------------------|-------|---------------------------|
| S_0 | $\{A_{prot}\}$ | R_0 | $\{A_{prot}\} \cup \{-\}$ |
| s_1 | N L V N S E H R M | r_1 | N L - V N S E H R M |
| s_2 | N L Y V P S E M I | r_2 | N L Y V P S E M I - |
| s_3 | Y V N E H M | s_3 | - - Y V - N E H M - |

Tabela 5.2: Exemplo de Seqüências e um Possível Alinhamento Múltiplo

Cada gene é uma coluna do alinhamento múltiplo de proteína. A tabela 5.3 mostra a formação de genes conforme os AFGs.

| g_1 | g_2 | g_3 | g_4 | g_5 | g_6 | g_7 | g_8 | g_9 | g_{10} |
|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|
| $\langle N, 1 \rangle$ | $\langle L, 2 \rangle$ | $\langle -, 0 \rangle$ | $\langle V, 3 \rangle$ | $\langle N, 4 \rangle$ | $\langle S, 5 \rangle$ | $\langle E, 6 \rangle$ | $\langle H, 7 \rangle$ | $\langle R, 8 \rangle$ | $\langle M, 9 \rangle$ |
| $\langle N, 1 \rangle$ | $\langle L, 2 \rangle$ | $\langle Y, 3 \rangle$ | $\langle V, 4 \rangle$ | $\langle P, 5 \rangle$ | $\langle S, 6 \rangle$ | $\langle E, 7 \rangle$ | $\langle M, 8 \rangle$ | $\langle I, 9 \rangle$ | $\langle -, 0 \rangle$ |
| $\langle -, 0 \rangle$ | $\langle -, 0 \rangle$ | $\langle Y, 1 \rangle$ | $\langle V, 2 \rangle$ | $\langle -, 0 \rangle$ | $\langle N, 3 \rangle$ | $\langle E, 4 \rangle$ | $\langle H, 5 \rangle$ | $\langle M, 6 \rangle$ | $\langle -, 0 \rangle$ |

Tabela 5.3: Exemplo de Tipo Gene

Algumas operações que são realizadas no GAADT, especificamente para o problema de alinhamento múltiplo, trabalham com um conjunto de genes. Assim, foi necessário introduzir a definição 5.5, onde genes consecutivos são agrupados formando um bloco gênico.

Definição 5.5 (Bloco Gênico) O bloco gênico é definido por $(G)^* = \langle g, \dots, g_j \rangle$, onde g_j é o último gene do bloco.

Exemplo 10 (Bloco Gênico) Considere os genes da tabela 5.3. Um exemplo de bloco é apresentado na tabela 5.4, formado pelo agrupamento de genes $\langle g_5, g_6, g_7 \rangle$.

No GAADT, genes são agrupados em cadeias para formar cromossomos. O conjunto de genes $\{g_1, g_2, \dots, g_L\}$ que compõe um cromossomo c é usado para indicar as características de c . Para saber quais grupamentos de genes podem

| g_5 | g_6 | g_7 |
|------------------------|------------------------|------------------------|
| $\langle N, 4 \rangle$ | $\langle S, 5 \rangle$ | $\langle E, 6 \rangle$ |
| $\langle P, 5 \rangle$ | $\langle S, 6 \rangle$ | $\langle E, 7 \rangle$ |
| $\langle -, 0 \rangle$ | $\langle N, 3 \rangle$ | $\langle E, 4 \rangle$ |

Tabela 5.4: Exemplo do Tipo Bloco Gênico

formar um cromossomo, um conjunto de axiomas de formação de cromossomos (AFC) é apresentado para determinar como o tipo cromossomo pode se comportar.

Definição 5.6 (Cromossomo) *O conjunto cromossomo é definido por $C = \{c = \{g_1, g_2, \dots, g_L\} \mid g_1, g_2, \dots, g_L \in G\}$ que satisfaz o seguinte axioma:*

$$AFC_1 = \{c \mid (\forall i \in \{1, 2, \dots, m\} \exists s_i \mid (\forall j \in \{1, 2, \dots, L\} (b_{i,j} \in g_j) \\ \exists r_i = \text{concat}(\text{primeiro}(b_{i,j}))) \wedge (s_i = \text{rem}(r_i))))\}$$

onde $\text{concat}(\text{primeiro}(b_{i,j}))$ é um predicado que concatena os símbolos dos primeiros elementos das bases $b_{i,j}$.

A definição 5.6 pressupõe que a configuração do cromossomo deve ser tal que a concatenação dos valores das bases que formam cada gene, aplicada a operação de remoção de *gaps* produz exatamente a seqüência original.

Exemplo 11 (Cromossomo) *Considere o alinhamento $R_0 = \{r_1, r_2, r_3\}$ da tabela 5.2. O Cromossomo c_1 é formado pelos genes $\{g_1, \dots, g_{10}\}$ a partir de R_0 . Abaixo é apresentado o conjunto c_1 :*

$$c_1 = \left\{ \begin{array}{l} g_1 = \{\langle N, 1 \rangle, \langle N, 1 \rangle, \langle -, 0 \rangle\}, \\ g_2 = \{\langle L, 2 \rangle, \langle L, 2 \rangle, \langle -, 0 \rangle\}, \\ g_3 = \{\langle -, 0 \rangle, \langle Y, 3 \rangle, \langle Y, 1 \rangle\}, \\ g_4 = \{\langle V, 3 \rangle, \langle V, 4 \rangle, \langle V, 2 \rangle\}, \\ g_5 = \{\langle N, 4 \rangle, \langle P, 5 \rangle, \langle -, 0 \rangle\}, \\ g_6 = \{\langle S, 5 \rangle, \langle S, 6 \rangle, \langle N, 3 \rangle\}, \\ g_7 = \{\langle E, 6 \rangle, \langle E, 7 \rangle, \langle E, 4 \rangle\}, \\ g_8 = \{\langle H, 7 \rangle, \langle M, 8 \rangle, \langle H, 5 \rangle\}, \\ g_9 = \{\langle R, 8 \rangle, \langle I, 9 \rangle, \langle M, 6 \rangle\}, \\ g_{10} = \{\langle M, 9 \rangle, \langle -, 0 \rangle, \langle -, 0 \rangle\}. \end{array} \right\}. \quad (5.9)$$

Um mapeamento da representação do alinhamento de c_1 é apresentado na tabela 5.5 para uma melhor visualização.

| | | | | | |
|-------|------------------------|------------------------|------------------------|------------------------|------------------------|
| c_1 | g_1 | g_2 | g_3 | g_4 | g_5 |
| | $\langle N, 1 \rangle$ | $\langle L, 2 \rangle$ | $\langle -, 0 \rangle$ | $\langle V, 3 \rangle$ | $\langle N, 4 \rangle$ |
| | $\langle N, 1 \rangle$ | $\langle L, 2 \rangle$ | $\langle Y, 3 \rangle$ | $\langle V, 4 \rangle$ | $\langle P, 5 \rangle$ |
| | $\langle -, 0 \rangle$ | $\langle -, 0 \rangle$ | $\langle Y, 1 \rangle$ | $\langle V, 2 \rangle$ | $\langle -, 0 \rangle$ |
| | g_6 | g_7 | g_8 | g_9 | g_{10} |
| | $\langle S, 5 \rangle$ | $\langle E, 6 \rangle$ | $\langle H, 7 \rangle$ | $\langle R, 8 \rangle$ | $\langle M, 9 \rangle$ |
| | $\langle S, 6 \rangle$ | $\langle E, 7 \rangle$ | $\langle M, 8 \rangle$ | $\langle I, 9 \rangle$ | $\langle -, 0 \rangle$ |
| | $\langle N, 3 \rangle$ | $\langle E, 4 \rangle$ | $\langle H, 5 \rangle$ | $\langle M, 6 \rangle$ | $\langle -, 0 \rangle$ |

Tabela 5.5: Exemplo de Tipo Cromossomo

Um cromossomo é uma representação do indivíduo. Para agrupar um conjunto de indivíduos, que em certo período são classificados como possíveis soluções, o GAADT propõe o tipo população que é formado por um conjunto representado por P .

Definição 5.7 (População) *O conjunto população é dado por:*

$$P = \{p = \{c_1, c_2, \dots, c_q\} \mid \{c_1, c_2, \dots, c_q\} \in C\} \quad (5.10)$$

Exemplo 12 (População) *Um exemplo de população para o conjunto de seqüências S_0 é apresentado na tabela 5.6, onde cada cromossomo representa um alinhamento múltiplo de proteína.*

5.3.2 Operadores Genéticos

Os operadores genéticos são aplicados em uma população P para modificar o material genético de seus cromossomos tentando encontrar novos indivíduos que melhor satisfazem os requisitos R_q de um ambiente A .

Como foi mostrado no capítulo anterior, o GAADT trabalha com dois tipos de operadores genéticos: cruzamento e mutação. Aplicar operadores genéticos implica em uma pré-avaliação de quão bom é um gene. Entretanto, uma especificidade do problema de alinhamento múltiplo dá-se por analisar um conjunto de genes.

Dados os genes $(G_x)^* \in c_1$ e $(G_y)^* \in c_2$, diz-se que o conjunto de genes $(G_x)^*$ melhor satisfaz um requisito r que os genes $(G_y)^*$, se ambos satisfazem o mesmo requisito $r \in R_q$ e se o grau de adaptação de $(G_x)^*$ é superior ao grau de adaptação dos genes $(G_y)^*$.

Por sua vez, o grau de adaptação de um gene abrange a análise de todos os elementos de $\text{primeiro}(b_{i,j})$, mas devem ser separados os casos em que o tipo

| | | | | | | |
|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|
| p_1 | c_1 | g_1 | g_2 | g_3 | g_4 | g_5 |
| | | $\langle N, 1 \rangle$ | $\langle L, 2 \rangle$ | $\langle -, 0 \rangle$ | $\langle V, 3 \rangle$ | $\langle N, 4 \rangle$ |
| | | $\langle N, 1 \rangle$ | $\langle L, 2 \rangle$ | $\langle Y, 3 \rangle$ | $\langle V, 4 \rangle$ | $\langle P, 5 \rangle$ |
| | | $\langle -, 0 \rangle$ | $\langle -, 0 \rangle$ | $\langle Y, 1 \rangle$ | $\langle V, 2 \rangle$ | $\langle -, 0 \rangle$ |
| | | g_6 | g_7 | g_8 | g_9 | g_{10} |
| | | $\langle S, 5 \rangle$ | $\langle E, 6 \rangle$ | $\langle H, 7 \rangle$ | $\langle R, 8 \rangle$ | $\langle M, 9 \rangle$ |
| | $\langle S, 6 \rangle$ | $\langle E, 7 \rangle$ | $\langle M, 8 \rangle$ | $\langle I, 9 \rangle$ | $\langle -, 0 \rangle$ | |
| | $\langle N, 3 \rangle$ | $\langle E, 4 \rangle$ | $\langle H, 5 \rangle$ | $\langle M, 6 \rangle$ | $\langle -, 0 \rangle$ | |
| | c_2 | g_1 | g_2 | g_3 | g_4 | g_5 |
| | | $\langle N, 1 \rangle$ | $\langle L, 2 \rangle$ | $\langle -, 0 \rangle$ | $\langle V, 3 \rangle$ | $\langle N, 4 \rangle$ |
| | | $\langle N, 1 \rangle$ | $\langle L, 2 \rangle$ | $\langle Y, 3 \rangle$ | $\langle V, 4 \rangle$ | $\langle P, 5 \rangle$ |
| | | $\langle -, 0 \rangle$ | $\langle Y, 1 \rangle$ | $\langle -, 0 \rangle$ | $\langle V, 2 \rangle$ | $\langle -, 0 \rangle$ |
| | | g_6 | g_7 | g_8 | g_9 | g_{10} |
| | | $\langle S, 5 \rangle$ | $\langle E, 6 \rangle$ | $\langle H, 7 \rangle$ | $\langle R, 8 \rangle$ | $\langle M, 9 \rangle$ |
| | $\langle S, 6 \rangle$ | $\langle E, 7 \rangle$ | $\langle M, 8 \rangle$ | $\langle I, 9 \rangle$ | $\langle -, 0 \rangle$ | |
| $\langle N, 3 \rangle$ | $\langle E, 4 \rangle$ | $\langle H, 5 \rangle$ | $\langle -, 0 \rangle$ | $\langle M, 6 \rangle$ | | |
| c_3 | g_1 | g_2 | g_3 | g_4 | g_5 | |
| | $\langle N, 1 \rangle$ | $\langle L, 2 \rangle$ | $\langle V, 3 \rangle$ | $\langle N, 4 \rangle$ | $\langle -, 0 \rangle$ | |
| | $\langle N, 1 \rangle$ | $\langle L, 2 \rangle$ | $\langle Y, 3 \rangle$ | $\langle V, 4 \rangle$ | $\langle P, 5 \rangle$ | |
| | $\langle -, 0 \rangle$ | $\langle Y, 1 \rangle$ | $\langle V, 2 \rangle$ | $\langle -, 0 \rangle$ | $\langle N, 3 \rangle$ | |
| | g_6 | g_7 | g_8 | g_9 | g_{10} | |
| | $\langle S, 5 \rangle$ | $\langle E, 6 \rangle$ | $\langle H, 7 \rangle$ | $\langle R, 8 \rangle$ | $\langle M, 9 \rangle$ | |
| $\langle S, 6 \rangle$ | $\langle E, 7 \rangle$ | $\langle -, 0 \rangle$ | $\langle M, 8 \rangle$ | $\langle I, 9 \rangle$ | | |
| $\langle -, 0 \rangle$ | $\langle E, 4 \rangle$ | $\langle -, 0 \rangle$ | $\langle H, 5 \rangle$ | $\langle M, 6 \rangle$ | | |

Tabela 5.6: Exemplo de Tipo População

base pertence aos conjuntos $\{A_{prot}\}$ e $\{-\}$, dando flexibilidade para escolha de matrizes de substituição e penalidade de *gaps*, respectivamente. Assim, dois predicados foram definidos para permitir esta separação: *getGaps* e *getPairs*.

Definição 5.8 (Busca Gaps) *O predicado que, dado um gene g , busca todas as bases com gap é denominado *getGaps*, tal que:*

$$getGaps : G \rightarrow \wp(B)$$

$$getGaps(g_j) = \{b_{i,j} \mid \forall i \ 1 \leq i \leq m \ (primeiro(b_{i,j}) = \{-\})\}$$

Exemplo 13 (Busca Gaps) *Dadas as características expressas na população da tabela 5.6, ao aplicar o predicado *getGaps* em cada gene dos cromossomos c_1 , c_2 e c_3 obtém-se as seguintes bases com gaps:*

$$c_1 = \left\{ \begin{array}{l} getGaps(g_1) = \{b_{3,1} = \langle -, 0 \rangle\} \\ getGaps(g_2) = \{b_{3,2} = \langle -, 0 \rangle\} \\ getGaps(g_3) = \{b_{1,3} = \langle -, 0 \rangle\} \\ getGaps(g_4) = \emptyset \\ getGaps(g_5) = \{b_{3,5} = \langle -, 0 \rangle\} \\ getGaps(g_6) = \emptyset \\ getGaps(g_7) = \emptyset \\ getGaps(g_8) = \emptyset \\ getGaps(g_9) = \emptyset \\ getGaps(g_{10}) = \{b_{2,10} = \langle -, 0 \rangle, b_{3,10} = \langle -, 0 \rangle\} \end{array} \right.$$

$$c_2 = \left\{ \begin{array}{l} getGaps(g_1) = \{b_{3,1} = \langle -, 0 \rangle\} \\ getGaps(g_2) = \emptyset \\ getGaps(g_3) = \{b_{1,3} = \langle -, 0 \rangle, b_{3,3} = \langle -, 0 \rangle\} \\ getGaps(g_4) = \emptyset \\ getGaps(g_5) = \{b_{3,5} = \langle -, 0 \rangle\} \\ getGaps(g_6) = \emptyset \\ getGaps(g_7) = \emptyset \\ getGaps(g_8) = \emptyset \\ getGaps(g_9) = \{b_{3,9} = \langle -, 0 \rangle\} \\ getGaps(g_{10}) = \{b_{2,10} = \langle -, 0 \rangle\} \end{array} \right.$$

$$c_3 = \begin{cases} \text{getGaps}(g_1) = \{b_{3,1} = \langle -, 0 \rangle\} \\ \text{getGaps}(g_2) = \emptyset \\ \text{getGaps}(g_3) = \emptyset \\ \text{getGaps}(g_4) = \{b_{3,4} = \langle -, 0 \rangle\} \\ \text{getGaps}(g_5) = \{b_{1,5} = \langle -, 0 \rangle\} \\ \text{getGaps}(g_6) = \{b_{3,6} = \langle -, 0 \rangle\} \\ \text{getGaps}(g_7) = \emptyset \\ \text{getGaps}(g_8) = \{b_{2,8} = \langle -, 0 \rangle, b_{3,8} = \langle -, 0 \rangle\} \\ \text{getGaps}(g_9) = \emptyset \\ \text{getGaps}(g_{10}) = \emptyset \end{cases}$$

Definição 5.9 (Busca Pares) O predicado que dado um gene g busca todas as possíveis combinações de pares de bases sem gap é denominado getPairs , tal que:

$$\text{getPairs} : G \rightarrow \wp(B, B)$$

$$\begin{aligned} \text{getPairs}(g_j) = \{ & (a_u, a_v) \mid \forall u \ 1 \leq u \leq m-1, \forall v \ u \leq v \leq m \\ & (a_u \in \{\text{primeiro}(b_{1,j}), \dots, \text{primeiro}(b_{m-1,j})\}) \wedge \\ & (a_v \in \{\text{primeiro}(b_{2,j}), \dots, \text{primeiro}(b_{m,j})\}) \} \end{aligned}$$

onde $a_u \in A_{\text{prot}}$ e $a_v \in A_{\text{prot}}$.

Exemplo 14 (Busca Pares) Dadas as características expressas na população da tabela 5.6, ao aplicar o predicado getPairs em cada gene dos cromossomos c_1 , c_2 e c_3 obtém-se os seguintes pares de bases:

$$c_1 = \begin{cases} \text{getPairs}(g_1) = \{(N, N)\} \\ \text{getPairs}(g_2) = \{(L, L)\} \\ \text{getPairs}(g_3) = \{(Y, Y)\} \\ \text{getPairs}(g_4) = \{(V, V), (V, V), (V, V), \} \\ \text{getPairs}(g_5) = \{(N, P)\} \\ \text{getPairs}(g_6) = \{(S, S), (S, N), (S, N)\} \\ \text{getPairs}(g_7) = \{(E, E), (E, E), (E, E)\} \\ \text{getPairs}(g_8) = \{(H, M), (H, H), (M, H)\} \\ \text{getPairs}(g_9) = \{(R, I), (R, M), (I, M)\} \\ \text{getPairs}(g_{10}) = \emptyset \end{cases}$$

$$c_2 = \left\{ \begin{array}{l}
 \text{getPairs}(g_1) = \{(N, N)\} \\
 \text{getPairs}(g_2) = \{(L, L), (L, Y), (L, Y)\} \\
 \text{getPairs}(g_3) = \emptyset \\
 \text{getPairs}(g_4) = \{(V, V), (V, V), (V, V)\} \\
 \text{getPairs}(g_5) = \{(N, P)\} \\
 \text{getPairs}(g_6) = \{(S, S), (S, N), (S, N)\} \\
 \text{getPairs}(g_7) = \{(E, E), (E, E), (E, E)\} \\
 \text{getPairs}(g_8) = \{(H, M), (H, H), (M, H)\} \\
 \text{getPairs}(g_9) = \{(R, I)\} \\
 \text{getPairs}(g_{10}) = \{(M, M)\}
 \end{array} \right.$$

$$c_3 = \left\{ \begin{array}{l}
 \text{getPairs}(g_1) = \{(N, N)\} \\
 \text{getPairs}(g_2) = \{(L, L), (L, Y), (L, Y)\} \\
 \text{getPairs}(g_3) = \{(V, Y), (V, V), (Y, V)\} \\
 \text{getPairs}(g_4) = \{(N, V)\} \\
 \text{getPairs}(g_5) = \{(P, N)\} \\
 \text{getPairs}(g_6) = \{(S, S)\} \\
 \text{getPairs}(g_7) = \{(E, E), (E, E), (E, E)\} \\
 \text{getPairs}(g_8) = \emptyset \\
 \text{getPairs}(g_9) = \{(R, M), (R, H), (M, H)\} \\
 \text{getPairs}(g_{10}) = \{(M, I), (M, M), (I, M)\}
 \end{array} \right.$$

Finalmente, o grau de adaptação do gene foi baseado no método da soma dos pares, com um certo ajuste para permitir penalizar a presença de *gaps* no mesmo gene.

A penalidade de cada *gap* ocorre independente das demais bases presentes no gene, as quais por sua vez são comparadas aos pares. Esta independência pode ser contornada caso seja de conhecimento prévio que a ausência de determinada base possa prejudicar a característica protéica. A seguir, é definido com mais detalhes como o grau de adaptação de um gene é calculado.

Definição 5.10 (Grau) *O grau de adaptação de um gene é uma função grau, tal que:*

$$\text{grau} : G \rightarrow \mathbb{R}$$

$$\text{grau}(g_j = \langle b_{1,j}, b_{2,j}, \dots, b_{m,j} \rangle) = \sum_{x=1}^{x'} p(\text{getPairs}(g_j)_x) + \sum_{y=1}^{y'} \mathbf{G}_{\text{pen}}(\text{getGaps}(g_j)_y) \quad (5.11)$$

onde:

$p(a_u, a_v)$ é a função que determina a pontuação entre os aminoácidos a_u e a_v

de acordo com os requisitos R_a ; e G_{pen} é o modelo de penalidade de gaps que obedece os requisitos R_p ; x' é a quantidade de pares obtida utilizando a função $cardinality(getPairs(g_j))$; y' é a quantidade de gaps obtida utilizando a função $cardinality(getGaps(g_j))$;

Lema 5.3.1 $\forall g : G \mid g \neq g_\lambda (grau(g) > grau(g_\lambda))$.

Pode-se observar que a definição 5.10 precisa de dois requisitos para indicar o sistema de pontuação a ser utilizado no alinhamento. O requisito R_a determina a matriz de substituição ou modelo para penalizar comparações entre aminoácidos, enquanto o requisito R_p indica o modelo de penalidade de gaps pré-estabelecido. Quanto ao valor do gene inócuo, no presente problema considera-se $grau(g_\lambda) = 0$.

Exemplo 15 (Grau) Considere os seguintes requisitos do ambiente:

- $R_a \in R_q \mid ((a_1 = a_2) \rightarrow (p(a_1, a_2) = +1)) \vee ((a_1 \neq a_2) \rightarrow (p(a_1, a_2) = -1))$
- $R_p \in R_q \mid (G_{pen} = -2)$;

A tabela 5.7 mostra o função grau para os genes dos cromossomos c_1 , c_2 e c_3 .

Por sua vez, o grau de adaptação do cromossomo é dada pela soma dos graus de todos os seus genes, conforme mostrado na definição a seguir.

Definição 5.11 (Adaptação) O grau de adaptação do cromossomo é uma função $adapt : C \rightarrow \mathbb{R}$, tal que:

$$adapt(c) = \sum_{j=1}^L grau(g_j) \quad (5.12)$$

onde L é o comprimento do cromossomo, ou seja, a quantidade de genes que o cromossomo contém.

Exemplo 16 (Adaptação) Considerando os graus dos genes da tabela 5.7. A tabela 5.8 mostra o valor de adaptação dos cromossomos c_1 e c_2 .

A adaptação média de uma população $adapt_m$ é a soma do grau de adaptação de todos os cromossomos dividida pelo número de indivíduos da população.

Definição 5.12 (Adaptação Média) A adaptação média de uma população é uma função $adapt_m : P \rightarrow \mathbb{R}$, tal que:

$$adapt_m(P) = \frac{\sum_{c \in P} adapt(c)}{cardinality(P)} \quad (5.13)$$

| c_1 | $\sum_{x=1}^{x'} p(\text{getPairs}(g_j)_x) + \sum_{y=1}^{y'} G_{pen}(\text{getGaps}(g_j)_y)$ | $grau(g_j)$ |
|----------|----------------------------------------------------------------------------------------------|-------------|
| g_1 | $p(N, N) + G_{pen}(b_{3,1})$ | -1 |
| g_2 | $p(L, L) + G_{pen}(b_{3,2})$ | -1 |
| g_3 | $p(Y, Y) + G_{pen}(b_{1,3})$ | -1 |
| g_4 | $p(V, V) + p(V, V) + p(V, V)$ | +3 |
| g_5 | $p(N, P) + G_{pen}(b_{3,5})$ | -3 |
| g_6 | $p(S, S) + p(S, N) + p(S, N)$ | -1 |
| g_7 | $p(E, E) + p(E, E) + p(E, E)$ | +3 |
| g_8 | $p(H, M) + p(H, H) + p(M, H)$ | -1 |
| g_9 | $p(R, I) + p(R, M) + p(I, M)$ | -3 |
| g_{10} | $G_{pen}(b_{2,10}) + G_{pen}(b_{3,10})$ | -4 |
| c_2 | $\sum_{x=1}^{x'} p(\text{getPairs}(g_j)_x) + \sum_{y=1}^{y'} G_{pen}(\text{getGaps}(g_j)_y)$ | $grau(g_j)$ |
| g_1 | $p(N, N) + G_{pen}(b_{3,1})$ | -1 |
| g_2 | $p(L, L) + p(L, Y) + p(L, Y)$ | -1 |
| g_3 | $G_{pen}(b_{1,3}) + G_{pen}(b_{3,3})$ | -4 |
| g_4 | $p(V, V) + p(V, V) + p(V, V)$ | +3 |
| g_5 | $p(N, P) + G_{pen}(b_{3,5})$ | -3 |
| g_6 | $p(S, S) + p(S, N) + p(S, N)$ | -1 |
| g_7 | $p(E, E) + p(E, E) + p(E, E)$ | +3 |
| g_8 | $p(H, M) + p(H, H) + p(M, H)$ | -1 |
| g_9 | $p(R, I) + G_{pen}(b_{3,9})$ | -3 |
| g_{10} | $p(M, M) + G_{pen}(b_{2,10})$ | -1 |
| c_3 | $\sum_{x=1}^{x'} p(\text{getPairs}(g_j)_x) + \sum_{y=1}^{y'} G_{pen}(\text{getGaps}(g_j)_y)$ | $grau(g_j)$ |
| g_1 | $p(N, N) + G_{pen}(b_{3,1})$ | -1 |
| g_2 | $p(L, L) + p(L, Y) + p(L, Y)$ | -1 |
| g_3 | $p(V, Y) + p(V, V) + p(Y, V)$ | -1 |
| g_4 | $p(N, V) + G_{pen}(b_{3,4})$ | -3 |
| g_5 | $p(P, N) + G_{pen}(b_{1,5})$ | -3 |
| g_6 | $p(S, S) + G_{pen}(b_{3,6})$ | -1 |
| g_7 | $p(E, E) + p(E, E) + p(E, E)$ | +3 |
| g_8 | $G_{pen}(b_{2,8}) + G_{pen}(b_{3,8})$ | -4 |
| g_9 | $p(R, M) + p(R, H) + p(M, H)$ | -3 |
| g_{10} | $p(M, I) + p(M, M) + p(I, M)$ | -1 |

Tabela 5.7: Exemplo de Grau de Adaptação do Gene

| | | |
|-------|-------------------------------------------------------|--------------------|
| c_1 | $adapt(c_1) = -1 - 1 - 1 + 3 - 3 - 1 + 3 - 1 - 3 - 4$ | $adapt(c_1) = -9$ |
| c_2 | $adapt(c_2) = -1 - 1 - 4 + 3 - 3 - 1 + 3 - 1 - 3 - 1$ | $adapt(c_2) = -9$ |
| c_2 | $adapt(c_2) = -1 - 1 - 1 - 3 - 3 - 1 + 3 - 4 - 3 - 1$ | $adapt(c_2) = -15$ |

Tabela 5.8: Exemplo de Grau de Adaptação do Cromossomo

Exemplo 17 (Adaptação Média) Dada a população da tabela 5.6, o valor da adaptação média dessa população é $adapt_m(P) = -11$.

Para aplicar os operadores, com a finalidade de criar novos cromossomos para as futuras gerações é necessário realizar a seleção daqueles que se submeterão a cada operador. Além disso, também é feito um ponto de corte, para permitir que alguns cromossomos da população atual sejam automaticamente repassados à próxima geração segundo algum critério.

Definição 5.13 (Ponto de Corte) O ponto de corte é um predicado $pCorte \subseteq Rq$,

$$pCorte = \forall c \text{ adapt}(c) \geq adapt_m \quad (5.14)$$

Tanto o requisito para selecionar cromossomos quanto o ponto de corte são especificados pelos requisitos do ambiente. Tem-se então a seleção, a qual é uma função que escolhe os cromossomos que melhor satisfazem os requisitos.

Definição 5.14 (Seleção) A seleção é uma função $sel \subseteq \wp(P_{atual})$,

$$sel(P_s, r_s) = \{c \in P_s \mid r_s\} \quad (5.15)$$

onde P_s é a subpopulação da qual devem ser selecionados cromossomos conforme os requisitos de seleção $r_s \in Rq$.

A seleção retorna um conjunto de cromossomos formando uma outra subpopulação. Uma aplicação da função seleção pode ser observada na divisão feita para definir quais os cromossomos que sofrem a ação de cada operador.

Exemplo 18 (Seleção) Seja a população atual formada pelos cromossomos c_1 , c_2 e c_3 , como mostrado anteriormente na tabela 5.6. Toda iteração do GAADT envolve a separação de P_{atual} em duas subpopulações:

- Subpopulação selecionada para cruzamento: $P_{sel_{cruz}} = sel(P_{atual}, pCorte)$
- Subpopulação selecionada para mutação: $P_{sel_{mut}} = sel(P_{atual}, \neg pCorte)$.

Seja $adapt(c_1) = -9$, $adapt(c_2) = -9$ e $adapt(c_3) = -15$. Assim temos:

- $P_{sel_{cruz}} = \{c_1, c_2\}$;
- $P_{sel_{mut}} = \{c_3\}$.

5.3.2.1 Cruzamento

Em Vieira (2003), o cromossomo resultante do cruzamento é formado somente pelos genes que são ditos dominantes nos cromossomos originais, havendo para isto comparações gene a gene em cromossomos distintos.

No entanto, foi necessária uma adaptação para tornar possível um operador que analise conjuntos de genes, denominados nesta dissertação por blocos gênicos (definição 5.5), tal que os blocos representam a mesma característica.

Uma proposta para diagnosticar a mesma característica gênica é pela análise das bases presentes nas seqüências correspondentes, ou seja, removendo-se a base inócua, as bases presentes na seqüência de um bloco devem ser exatamente as mesmas no outro bloco gênico.

Lema 5.3.2 $((G_x)^*, (G_y)^*) \in mesma \rightarrow \forall r_x \in (G_x)^*, \forall r_y \in (G_y)^* \exists s_i = rem(r_x) = rem(r_y)$

Exemplo 19 (Mesma) *Sejam os cromossomos c_1 e c_2 (tabela 5.6) aptos a serem submetidos ao cruzamento. Comparando-se os blocos gênicos tem-se que $(G_x)^* \in c_1$ e $(G_y)^* \in c_2$ possuem a mesma característica, pois para cada seqüência as bases são as mesmas (tabela 5.9).*

| | | | | |
|-------|-------|------------------------|------------------------|------------------------|
| c_1 | G_x | g_1 | g_2 | g_3 |
| | | $\langle N, 1 \rangle$ | $\langle L, 2 \rangle$ | $\langle -, 0 \rangle$ |
| | | $\langle N, 1 \rangle$ | $\langle L, 2 \rangle$ | $\langle Y, 3 \rangle$ |
| | | $\langle -, 0 \rangle$ | $\langle -, 0 \rangle$ | $\langle Y, 1 \rangle$ |
| c_2 | G_y | g_1 | g_2 | g_3 |
| | | $\langle N, 1 \rangle$ | $\langle L, 2 \rangle$ | $\langle -, 0 \rangle$ |
| | | $\langle N, 1 \rangle$ | $\langle L, 2 \rangle$ | $\langle Y, 3 \rangle$ |
| | | $\langle -, 0 \rangle$ | $\langle Y, 1 \rangle$ | $\langle -, 0 \rangle$ |

Tabela 5.9: Exemplo Mesma Característica

Caso dois blocos gênicos sejam equivalentes, possuindo a mesma característica com atributos relevantes ao sistema, então o bloco que melhor satisfaz os requisitos é selecionado. O bloco gênico escolhido é denominado dominante.

Definição 5.15 (Dominante) *O gene dominante é uma função $domi$, tal que:*

$$domi : G \times G \rightarrow G$$

$$domi(g_1, g_2) = \begin{cases} g_\lambda & \text{se } ((G_x)^* \supset g_1, (G_y)^* \supset g_2) \notin \text{mesma}, \\ g_1 & \text{se } ((G_x)^* \supset g_1, (G_y)^* \supset g_2) \in \text{mesma} \wedge \\ & \text{alinhamento}((G_x)^* \supset g_1) \geq \text{alinhamento}((G_y)^* \supset g_2), \\ g_2 & \text{se } ((G_x)^* \supset g_1, (G_y)^* \supset g_2) \in \text{mesma} \wedge \\ & \text{alinhamento}((G_x)^* \supset g_1) < \text{alinhamento}((G_y)^* \supset g_2) \end{cases}$$

onde $\text{alinhamento}((G)^*)$ é o predicado que retorna a soma dos graus do bloco $(G)^*$.

A função $domi$ é utilizada durante a fecundação de dois cromossomos, na qual são obtidos todos os genes considerados dominantes para serem preservados no alinhamento.

Definição 5.16 (Fecundação) A fecundação é uma função fec do seguinte tipo:

$$fec: C \times C \rightarrow \wp(G)$$

$$fec(c_1, c_2) = \{g \mid \forall g_x \in c_1 \forall g_y \in c_2 (g = domi(g_x, g_y))\}$$

Os blocos gênicos são encontrados por busca exaustiva analisando quais blocos representam a mesma característica. Ao ser realizado o cruzamento entre c_1 e c_2 , o cromossomo fecundado é formado por blocos com genes dominantes.

A fecundação é uma função usada quando se deseja gerar um novo cromossomo através do operador genético de cruzamento.

Primeiramente são selecionados pela função $sel(pCorte(P_{atual}), r_{cruz})$ os cromossomos que obedecem os requisitos r_{cruz} impostos pelo ambiente. A seleção retorna um conjunto de cromossomos aptos a passarem suas características gênicas para a próxima geração, mas desses apenas dois são suficientes para realizar cada operação de cruzamento. Então, desta subpopulação são escolhidos dois conjuntos, denominados cromossomos *MACHO* e cromossomos *FEMEA*.

Assim, a escolha desses cromossomos obedece os seguintes requisitos:

- os cromossomos obtidos da função $sel(P_{atual}, pCorte)$ são agrupados na subpopulação P_1 , formada por cromossomos adaptados ao ambiente;
- $MACHO = sel(P_1, M)$ e $FEMEA = sel(P_1, F)$;
- M e F são dois predicados sobre o tipo população pertencentes ao conjunto de requisitos do ambiente Rq ;
- se $M = F$ a reprodução é dita assexuada, onde todos os cromossomos podem pertencer ao conjunto *MACHO* e ao conjunto *FEMEA*;

- se $M \cap F = \emptyset$ a reprodução é dita sexuada, onde os cromossomos são particionados, metade são classificados como *MACHO* e a outra metade como *FEMEA*;

Definição 5.17 (Cruzamento) *O cruzamento é uma função cruz, tal que:*

$$cruz : MACHO \times FEMEA \rightarrow P$$

$$P_{cruz} = cruz(c_1, c_2) = \{c \mid c \supset fec(c_1, c_2) \wedge (c \in AFC) \wedge (adapt(c) \geq adapt_m(P_{atual}))\}$$

onde $c_1 \in MACHO$ e $c_2 \in FEMEA$.

O cromossomo resultante só é inserido na subpopulação de cromossomos oriundos do cruzamento, se for mais adaptado que a média da população que o criou.

5.3.2.2 Mutação

Para aumentar a variabilidade do alinhamento há a possibilidade de inserção de um gene inócuo no cromossomo, pois a presença deste gene não altera o material genético, apenas permite que futuras operações genéticas possam mais *gaps* a reposicionar, encontrando assim novas combinações de bases.

Definição 5.18 (Inserção) *A inserção de um gene inócuo é dada pela função ins, tal que:*

$$ins : C \times g_\lambda \rightarrow C$$

$$ins(c, g_\lambda) = c \cup g_\lambda$$

onde a posição de inserção de g_λ é escolhida aleatoriamente.

Com várias iterações do algoritmo pode ser que ao finalizar ele possua um gene inócuo, cujas bases não foram permutadas com as bases dos demais genes ou chegaram a esta conformação final. Então pode-se suprimir o gene inócuo do alinhamento resultante.

Definição 5.19 (Supressão) *A supressão do gene inócuo é dada pela função del, tal que:*

$$del : C \times g_\lambda \rightarrow C$$

$$del(c, g_\lambda) = c - g_\lambda$$

O operador genético de mutação propriamente dito é definido como o predicado que constrói uma população de cromossomos aplicando uma função que realiza a troca de um conjunto de genes.

Dado um cromossomo, os genes que devem sofrer mutação são aqueles cuja função *grau* tenha um valor abaixo da média dos graus dos genes que compõe este cromossomo. A busca exaustiva por posições de mutação é definida a seguir.

Definição 5.20 (Busca Gene) O predicado que dado um cromossomo c busca todos os genes que devem ser mutados é denominado *getGenes*, tal que:

$$getGenes : C \rightarrow \wp(G)$$

$$getGenes(c) = \{g_j \mid \forall j \ 1 \leq j \leq cardinality(c), (b_\lambda \in g_j) \wedge (grau(g_j) < (\sum grau(g_j)/cardinality(c)))\}$$

A cada gene encontrado pela função *buscaGenes(c)*, genes adjacentes são adicionados formando um bloco gênico. Durante a mutação, este bloco gênico sofre alteração mediante o uso da função *trocaGaps*, onde uma nova posição para a base inócua é escolhida ao acaso enquanto as bases adjacentes são movidas de uma posição para dar espaço para que o *gap* seja inserido, tentando melhorar a adaptação do cromossomo.

Definição 5.21 (Troca Gaps) Troca Gaps é um predicado *trocaGaps*, tal que:

$$trocaGaps : \wp((G)^*) \rightarrow \wp((G)^*)$$

$$trocaGaps((G_x)^*) = \{ (G_y)^* = \langle g_1, \dots, g_n \rangle \mid (\exists j \ 1 \leq j \leq n) (\exists i \ 1 \leq i \leq m)$$

$$(primeiro(b_{i,j}) = -)$$

$$((k > j) \rightarrow (\forall j', j \leq j' < k) (b_{i,j'} = b_{i,j'+1}) \wedge (b_{i,k} = \langle -, 0 \rangle))$$

$$((k < j) \rightarrow (\forall j', j \geq j' > k) (b_{i,j'} = b_{i,j'-1}) \wedge (b_{i,k} = \langle -, 0 \rangle)) \}$$

onde: n é a quantidade de genes do conjunto $(G_x)^*$, com $n = cardinality((G_x)^*)$; m é a quantidade de seqüências submetidas; e k é uma posição escolhida ao acaso para ser ocupada pela base inócua.

A definição 5.21, faz uma pesquisa por todas as bases com *gaps*. Então, há uma movimentação das bases a direita ou a esquerda, dependendo da posição k , resultando em um subconjunto de genes mutados.

A função *troca* é um caso especial das funções supressão e inserção, porém operando com um subconjunto de genes, os quais são submetidos a operação *trocaGaps*.

Então a função *troca* seleciona aleatoriamente um subconjunto de genes, os quais são suprimidos, prevalecendo a característica expressa no novo subconjunto que alterou as posições dos *gaps* encontrados.

Definição 5.22 (Troca) A troca é uma função *troc*, tal que:

$$troc : C \times \wp(G) \times \wp(G) \rightarrow C$$

$$troc(c, g_1, g_2) = \begin{cases} (c - ((G_x)^* \supset g_1)) \cup ((G_y)^* \supset g_2) & \text{if } ((c - (G_x)^*) \cup (G_y)^*) \in AFC \\ c & \text{c.c.} \end{cases} \quad (5.16)$$

onde $cardinality((G_x)^*) = cardinality((G_y)^*)$ e $(G_y)^*$ é determinado pelo predicado *trocaGaps* a partir de $(G_x)^*$.

O operador de mutação só permite nascer na nova população aqueles cromossomos que apresentem melhoria no alinhamento, ou seja, sua adaptação seja superior ao cromossomo que o originou.

Definição 5.23 (Mutação) A mutação é um predicado *mut*, tal que:

$$mut : C \rightarrow \wp(P)$$

$$P_{mut} = mut(c_1) = \{c_2 \mid \exists G_x \in \wp(G), \exists G_y \in \wp(G), (c_2 = troca(c_1, G_x, G_y)) \wedge (adapt(c_2) > adapt(c_1))\} \quad (5.17)$$

onde $(cardinality(G_x) \leq (cardinality(c_1) \text{ div } 2))$ e $(cardinality(G_y) = cardinality(G_x))$.

A quantidade de genes no cromossomo fornecido que podem ser mutados é limitada em no máximo cinquenta por cento, para evitar que as mutações ocorridas em um cromossomo sejam muito grandes.

5.4 Ambiente

O GAADT opera sobre populações de cromossomos que evoluem de acordo com as características de um ambiente A (Vieira 2003). Define-se um ambiente A como uma 8-tupla $\langle P, \mathbb{P}(P), Rq, AFG, AGC, Tx, \Sigma, P_0 \rangle$, onde:

- P é a população de alinhamentos;
- $\mathbb{P}(P)$ é o conjunto potência de P ;
- Rq é o conjunto de requisitos que devem ser obedecidos durante a criação das várias gerações;
- AFG é o conjunto de axiomas de formação dos genes dos cromossomos da população P (definição 5.4);
- AFC é o conjunto de axiomas de formação dos cromossomos da população P (definição 5.6);

- Tx é o conjunto de pares de cromossomos (x, y) , onde x é um cromossomo construído a partir do cromossomo y , pela ação da operação de cruzamento ou mutação, registrando desta forma a genealogia dos cromossomos pertencentes às populações geradas pelo GAADT durante a sua execução,
- Σ é o conjunto de operadores genealógicos que atuam sobre a população P ,
- P_0 é uma sub-população pertencente a $\mathbb{P}(P)$, chamada de população inicial, com no mínimo um cromossomo.

Um conjunto de seqüências são fornecidas para a construção dos tipos abstrato de dados, até a criação da população inicial P_0 . A criação da população inicial consiste nas seqüências de entrada S_0 , que ao serem adicionados os *gaps* geram o conjunto R_0 , formando um cromossomo $c \in P_0$.

O tamanho da maior seqüência s_m é de grande importância no processo de criação da população inicial. Primeiramente, são adicionados aleatoriamente *gaps* em s_m em uma quantidade determinada empiricamente como $q_{gaps} = \text{cardinality}(s_m)/25$. Isto permite que as bases da maior seqüência também possam variar de posição quando o GAADT aplicar o operador genético de mutação.

O comprimento de todas as seqüências de R_0 será aquele obtido pela adição de *gaps* na maior seqüência, conforme mencionado no parágrafo anterior. Dessa forma, *gaps* são adicionados nas demais seqüências de R_0 de forma aleatória, de modo que todas atinjam o mesmo comprimento da maior seqüência.

Esse processo se repete para a criação de vários cromossomos, sendo diferenciados pelos *gaps* aleatórios que são adicionados em S_0 para criar cromossomos de R_0 . O tamanho da população inicial ainda precisa ser melhor trabalhado. No presente trabalho, convencionou-se adotar uma população inicial de tamanho 100.

Os cromossomos de cada população sofrem a ação dos operadores genéticos, criando novos cromossomos através das sucessivas iterações do algoritmo. Os detalhes do funcionamento do algoritmo são apresentados na seção seguinte.

5.5 Algoritmo

O GAADT foi interpretado para a função $GAADT_A$ específica ao problema de alinhamento múltiplo de proteína. Como o proposto em (Vieira 2003), o

$GAADT_A$ recebe uma população P_0 e, após submetê-la ao algoritmo que se assemelha ao processo evolutivo, devolve uma população P_t .

O processo evolutivo começa determinando o critério de preservação sobre a população atual, neste caso é usado o predicado $pCorte$. Assim, todos os cromossomos que obedecem o ponto de corte são preservados e permanecer na próxima geração.

$$P_g = sel(P_{atual}, p_{corte})$$

Antes de aplicar o algoritmo sobre a população atual, precisa ser determinado o requisito que diz respeito ao critério de parada. O critério de parada no $GAADT_A$ é empírico, pois não se sabe qual o valor de adaptação para um alinhamento ótimo. A única informação que se pode ter é referente ao número de gerações, limitando-se aos requisitos de hardware do computador de teste e tempo hábil de processamento. O número de gerações T será evidenciado na seção de testes e resultados.

Feita a separação dos cromossomos de P_{atual} em $P_{selcruz}$ e P_{selmut} , se cada cromossomo filho apresentar uma adaptação maior que a adaptação média da população que o originou, então ele é adicionado à P_g .

Definição 5.24 (GAADT) *O GAADT é uma função $GAADT_A$ para alinhamento múltiplo de proteínas, tal que:*

$$GAADT_A : A \rightarrow A$$

$$GAADT_A(P_t) = \begin{cases} P_{t+1} & \text{se } t + 2 = T, \\ GAADT(P_{t+1}) & \text{caso contrário.} \end{cases}$$

onde $P_{t+1} = p_{corte}(P_t) \cup cruz(a, b) \cup mut(c)$ com $a, b, c \in P_t$, P_0 é a população inicial considerada e $T \in \mathbb{N}$ é um número dado como critério de satisfação do número de iterações.

Capítulo 6

Resultados

A análise estatística da presente instanciação foi realizada utilizando um banco de dados público de alinhamento de proteínas conhecido como BALIBASE¹ (*benchmark alignment database*) (Thompson et al. 1999a). O propósito da criação do BALIBASE foi servir como base para comparação entre os vários métodos de alinhamento múltiplo existentes, o que permite comparar os resultados da instanciação do GAADT apresentada nos capítulos anteriores com várias outras técnicas analisadas e cujos resultados puderam ser obtidos na literatura.

Uma vantagem da utilização do BALIBASE é que torna a comparação entre os modelos computacionais independente de parâmetros como matrizes de substituição, funções de penalização de *gaps* e funções objetivo. A verificação do quão adequado é o resultado da ferramenta computacional é realizada calculando a similaridade entre o resultado e o alinhamento referência fornecido pelo BALIBASE.

Um grupo de 142 alinhamentos referência são fornecidos pelo BALIBASE. Esses alinhamentos foram gerados manualmente de modo a garantir o alinhamento de resíduos funcionais e outras regiões conservadas na estrutura tridimensional.

Os resultados apresentados na próxima seção comparam o GAADT com uma série de outras técnicas de alinhamento calculadas por Thompson et al. (1999b). Entretanto, os resultados a seguir não possuem a mesma quantidade de detalhes dos apresentados por Thompson et al. (1999b), pois os recursos computacionais disponíveis impediram uma análise com todos os 142 alinhamentos de referência.

O ambiente de execução dos testes era composto por: um processador AMD Athlon XP 2800+ de 2.0 Ghz, 512 MB de memória RAM e sistema operacional

¹<http://bips.u-strasbg.fr/fr/Products/Databases/BAlIBASE/>

Windows XP SP2. O GAADT foi implementado na linguagem de programação JAVA e nos testes usou-se a versão JDK 1.6.

Vale ressaltar que, mesmo não alcançando todo o potencial de especificação proposto pelo GAADT e nem fazendo uma análise estatística mais detalhada dos modelos, é possível concluir que o GAADT consegue resolver a tarefa de alinhamento múltiplo de proteínas com uma certa regularidade quando a escolha das matrizes de pontuação e os parâmetros de penalidade de *gaps* são escolhidos adequadamente.

O presente capítulo está dividido em três seções: a seção 6.1 é uma breve descrição do modo como a análise de qualidade das técnicas de alinhamento é obtida quando se usa o BALIBASE. A seção 6.2 mostra a performance do GAADT com a escolha de alguns parâmetros, um fator que, como será visto, é determinante para um bom alinhamento. A seção 6.3 apresenta uma comparação do GAADT com os resultados obtidos por Thompson et al. (1999b).

6.1 Medida de Qualidade do BALIBASE

Dados um alinhamento de referência R e um alinhamento de teste T , o BALIBASE fornece um cálculo para saber o quanto T se aproxima de R . Seja m o número de seqüências do alinhamento e n o número de colunas no alinhamento de teste T , denota-se os elementos da i -ésima coluna do alinhamento T por $t_{i1}, t_{i2}, \dots, t_{im}$.

Para cada par de resíduos t_{ij} e t_{ik} define-se ρ_{ijk} da seguinte forma:

$$\rho_{ijk} = \begin{cases} 1 & \text{se } t_{ij} \text{ e } t_{ik} \text{ estão na mesma coluna em } R \\ 0 & \text{c.c.} \end{cases}$$

O score da i -ésima coluna é dado por:

$$s_i = \sum_{j=1, j \neq k}^m \sum_{k=1}^m \rho_{ijk}$$

A pontuação do alinhamento de teste T é obtido pelo seguinte cálculo:

$$S_T = \sum_{i=1}^n s_i$$

Para calcular o quanto o alinhamento de teste T está próximo do alinhamento referência R , é preciso também calcular a pontuação que R obteria caso fosse executado como alinhamento de teste, seguindo os passos descritos acima e obtendo o valor S_R , ou seja, S_R é o maior valor que pode ser obtido

por um alinhamento utilizando o cálculo descrito nesta seção.

O *score* final do alinhamento T é dado por:

$$score = S_T/S_R$$

6.2 Parâmetros do Algoritmo

No capítulo anterior foram mostradas as características do algoritmo genético baseado em tipos abstratos de dados. Entretanto, alguns parâmetros típicos de algoritmos genéticos foram deixados em aberto e serão especificados na seção 6.2.2. Vale ressaltar que a escolha dos parâmetros do algoritmo genético foi, na maioria das vezes, realizada de acordo com as restrições computacionais do ambiente de execução.

Um outro parâmetro que ficou em aberto foi a escolha da matriz de substituição. Este parâmetro não está relacionado com as especificações de algoritmos genéticos, mas com o modelo de alinhamento múltiplo de proteínas. Dessa forma, uma análise estatística foi realizada e apresentada na seção 6.2.1.

6.2.1 Matriz de substituição

Matrizes de substituição são adequadas de acordo com as seqüências que se deseja alinhar, dependendo se as seqüências são fortemente relacionadas ou se divergiram no decorrer da evolução. Este parâmetro seria mais preciso se fosse conhecido o antepassado da família protéica.

O objetivo desta instanciação do GAADT é alinhar seqüências de proteínas sem um conhecimento mais profundo sobre elas. Assim, optou-se por utilizar a família protéica **1aab** como estudo de caso para comparar as matrizes de substituição BLOSUM62, *Dayhoff*, PAM40, PAM80, PAM120 e PAM250. Para cada matriz foram realizadas dez execuções do GAADT, sendo coletadas as informações abaixo:

- Adaptação Média de P_0 : a criação da população inicial P_0 segue uma heurística que será apresentada na seção 6.2.2.2. A adaptação média da população inicial foi armazenada para ser possível averiguar o impacto que causa no tempo de execução do algoritmo e no resultado apresentado pelo GAADT. Pode-se observar que, muitas vezes, a adaptação média de P_0 tem valor negativo, o que indica que uma heurística melhor do que

a utilizada e específica para a matriz em questão poderia ser implementada;

- Adaptação Média de P_n : com o decorrer das gerações os cromossomos evoluem gradativamente até resultarem em uma população que não apresenta melhoras. A adaptação média desta última população tem seu valor apresentado nesta coluna;
- Número de gerações: para analisar as matrizes de substituição, o critério de parada do GAADT foi a evolução da adaptação média de uma geração para outra. Caso a adaptação média não apresentasse melhoras no decorrer de 10 gerações a execução do GAADT era finalizada e o número total de gerações armazenado. Assim sendo, esta coluna representa o número de gerações que o GAADT apresentou em cada execução.
- Cromossomo mais adaptado: quando o algoritmo atinge o critério de parada, o cromossomo que apresenta o melhor alinhamento possui o valor de adaptação armazenado nesta coluna. Este valor é calculado pela função *adapt* do GAADT (definição 5.11) e deve ser lido apenas como um *score* interno do algoritmo, representando a função de adaptação adotada.
- *Score* BALIBASE: a saída do algoritmo é comparada com o alinhamento referência do BALIBASE utilizando a medida especificada na seção 6.1. Quanto mais próximo este valor estiver de 1.0, mais próximo o alinhamento está do alinhamento referência.
- Média *Score* BALIBASE: o parâmetro matriz de substituição foi um parâmetro preliminar a ser analisado para determinar o comportamento de uma matriz diante da família protéica **laab**. Para comparar as matrizes entre si utilizou-se a média dos *scores* BALIBASE.

A tabela 6.1 mostra os resultados utilizando a matriz de substituição BLOSUM62. Para esta execução a escala de valores da matriz proporcionou em todos os casos adaptação média de P_0 negativa. Em dois casos o alinhamento apresentou *score* BALIBASE igual a 0.640.

A tabela 6.2 mostra os resultados utilizando a matriz de substituição DAYHOFF. Para esta execução a escala de valores da matriz proporcionou duas execuções com adaptação média de P_0 positiva. Porém, mesmo o teste número 2 começando com uma adaptação média negativa, os cromossomos conseguiram evoluir e atingir o maior *score* BALIBASE desta fase de testes, cujo valor foi igual a 0.768.

| BLOSUM62 | | | | | |
|-----------------------------|--------------------------|--------------------------|--------------------|--------------------------|----------------|
| Teste | Adaptação Média de P_0 | Adaptação Média de P_n | Número de Gerações | Cromossomo Mais Adaptado | Score BALIBASE |
| 1 | -63.0315315315315 | 515.176767676768 | 100 | 531.5 | 0.640 |
| 2 | -73.0786516853932 | 570.085106382979 | 120 | 579 | 0.607 |
| 3 | -96.8488372093023 | 542.693467336683 | 80 | 549 | 0.631 |
| 4 | -123.059139784946 | 485.820707070707 | 70 | 491 | 0.485 |
| 5 | -112.557142857143 | 274.922279792746 | 90 | 287 | 0.259 |
| 6 | -88.2816901408451 | 517.264102564103 | 110 | 526 | 0.640 |
| 7 | -90.3557692307692 | 391.883838383838 | 110 | 400 | 0.372 |
| 8 | -54.0151515151515 | 502.385572139303 | 80 | 523.5 | 0.554 |
| 9 | -58.5509259259259 | 414.758706467662 | 80 | 431 | 0.500 |
| 10 | -73.4734042553192 | 546.13440860215 | 140 | 562.5 | 0.601 |
| Média Score BALIBASE | | | | | 0.5289 |

Tabela 6.1: Teste família **laab** com matriz de substituição BLOSUM62

| DAYHOFF | | | | | |
|-----------------------------|--------------------------|--------------------------|--------------------|--------------------------|----------------|
| Teste | Adaptação Média de P_0 | Adaptação Média de P_n | Número de Gerações | Cromossomo Mais Adaptado | Score BALIBASE |
| 1 | -16.5333333333333 | 681.676470588235 | 130 | 687 | 0.729 |
| 2 | -71.3170731707317 | 689.994871794872 | 120 | 698 | 0.768 |
| 3 | -48.5822784810127 | 655.879888268156 | 200 | 664.5 | 0.61 |
| 4 | -31.993670886076 | 677.497422680412 | 120 | 688.5 | 0.729 |
| 5 | 33.0412844036697 | 681.32320441989 | 70 | 690 | 0.643 |
| 6 | -36.4940476190476 | 714.794871794872 | 230 | 729 | 0.735 |
| 7 | -4.51485148514851 | 666.56 | 100 | 676 | 0.744 |
| 8 | -7.46236559139785 | 685.633165829146 | 130 | 695 | 0.69 |
| 9 | 27.9529914529914 | 695.755263157895 | 90 | 702 | 0.693 |
| 10 | -24.8173076923077 | 614.44414893617 | 90 | 632.5 | 0.497 |
| Média Score BALIBASE | | | | | 0.6838 |

Tabela 6.2: Teste família **laab** com matriz de substituição DAYHOFF

A tabela 6.3 mostra os resultados utilizando a matriz de substituição PAM40. Para esta execução, a escala de valores da matriz proporcionou em todos os casos adaptação média de P_0 negativa. Em dois casos o melhor cromossomo apresentou adaptação média positiva. Para a tabela PAM40 a configuração do melhor alinhamento se aproximou do alinhamento referência da família proteica **laab** em 63.1%. Entretanto, os resultados oscilaram muito entre bons e ruins, o que fez com que a média ficasse muito abaixo do desejado.

A tabela 6.4 mostra os resultados utilizando a matriz de substituição PAM80. Para esta execução a escala de valores da matriz também proporcionou em todos os casos adaptação média de P_0 negativa. Contudo, a escala de valores desta matriz não é tão baixa e a função de avaliação conseguiu atingir valores positivos.

Nos teste com a medida de qualidade do BALIBASE, o melhor resultado obtido (69.90%) foi apenas levemente superior ao melhor resultado utilizando a matriz PAM40 (63.1%). Entretanto, a matriz PAM80 apresentou mais regularidade nos resultados, obtendo uma média 51.25% de proximidade com o alinhamento referência.

| PAM40 | | | | | |
|-----------------------------|--------------------------|--------------------------|--------------------|--------------------------|----------------|
| Teste | Adaptação Média de P_0 | Adaptação Média de P_n | Número de Gerações | Cromossomo Mais Adaptado | Score BALIBASE |
| 1 | -962.665094339623 | -194.125 | 70 | -170 | 0.208 |
| 2 | -1085.52298850575 | -150.484924623116 | 120 | -123 | 0.402 |
| 3 | -1052.45 | -118.972222222222 | 100 | -98 | 0.313 |
| 4 | -985.495327102804 | 198.3825 | 190 | 224.5 | 0.631 |
| 5 | -951.605769230769 | 137.93085106383 | 190 | 144.5 | 0.589 |
| 6 | -967.183760683761 | -106.165816326531 | 120 | -82 | 0.384 |
| 7 | -1041.29896907216 | -162.714285714286 | 60 | -141 | 0.31 |
| 8 | -969.578431372549 | -90.790404040404 | 80 | -64 | 0.396 |
| 9 | -1033.97474747475 | -52.0201005025126 | 90 | -29 | 0.402 |
| 10 | -1032.6170212766 | -88.8179347826087 | 90 | -65 | 0.333 |
| Média Score BALIBASE | | | | | 0.3968 |

Tabela 6.3: Teste família **1aab** com matriz de substituição PAM40

| PAM80 | | | | | |
|-----------------------------|--------------------------|--------------------------|--------------------|--------------------------|----------------|
| Teste | Adaptação Média de P_0 | Adaptação Média de P_n | Número de Gerações | Cromossomo Mais Adaptado | Score BALIBASE |
| 1 | -452.72619047619 | 404.13 | 100 | 418.5 | 0.619 |
| 2 | -568.286458333333 | 142.375634517766 | 80 | 152.5 | 0.399 |
| 3 | -522.329787234043 | 233.723958333333 | 120 | 246 | 0.512 |
| 4 | -487.384615384615 | 401.611675126904 | 80 | 452 | 0.699 |
| 5 | -549.026315789474 | 428.515306122449 | 180 | 448 | 0.634 |
| 6 | -467.384210526316 | 442.234536082474 | 110 | 452 | 0.542 |
| 7 | -496.696808510638 | 172.529255319149 | 80 | 179 | 0.256 |
| 8 | -518.762886597938 | 213.736842105263 | 90 | 238 | 0.44 |
| 9 | -504.005208333333 | 153.409090909091 | 100 | 165.5 | 0.402 |
| 10 | -534.177419354839 | 447.232044198895 | 140 | 455.5 | 0.622 |
| Média Score BALIBASE | | | | | 0.5125 |

Tabela 6.4: Teste família **1aab** com matriz de substituição PAM80

A tabela 6.5 mostra os resultados com a matriz de substituição PAM120. Para esta execução a escala de valores da matriz também proporcionou em todos os casos adaptação média de P_0 negativa. Nesta execução é importante destacar que o cromossomo mais adaptado é o mesmo que obteve maior score BALIBASE. Esta proporcionalidade se manteve em todos os testes e demonstra que a otimização utilizando a matriz PAM120 gera alinhamentos mais próximos do alinhamento referência.

| PAM120 | | | | | |
|-----------------------------|--------------------------|--------------------------|--------------------|--------------------------|----------------|
| Teste | Adaptação Média de P_0 | Adaptação Média de P_n | Número de Gerações | Cromossomo Mais Adaptado | Score BALIBASE |
| 1 | -262.810526315789 | 431.816489361702 | 170 | 451.5 | 0.488 |
| 2 | -288.427884615385 | 558.329145728643 | 130 | 578 | 0.795 |
| 3 | -302.926315789474 | 248.168341708543 | 90 | 261.5 | 0.342 |
| 4 | -274.105882352941 | 417.677664974619 | 90 | 438 | 0.485 |
| 5 | -283.258620689655 | 257.886842105263 | 100 | 268.5 | 0.372 |
| 6 | -255.788888888889 | 328.154228855721 | 100 | 338.5 | 0.455 |
| 7 | -292.761904761905 | 497.616580310881 | 150 | 515.5 | 0.661 |
| 8 | -295.145833333333 | 191.59693877551 | 70 | 199.5 | 0.226 |
| 9 | -244.83908045977 | 483.540609137056 | 120 | 503.5 | 0.667 |
| 10 | -240.379120879121 | 410.597014925373 | 60 | 420 | 0.545 |
| Média Score BALIBASE | | | | | 0.5036 |

Tabela 6.5: Teste família **laab** com matriz de substituição PAM120

A tabela 6.6 mostra os resultados com a matriz de substituição PAM250. Para esta execução a escala de valores da matriz em alguns casos apresentou adaptação média de P_0 positiva. Assim como no teste da matriz PAM120, o cromossomo mais adaptado é o mesmo que obteve maior score BALIBASE. Esta proporcionalidade também se manteve em todos os testes.

| PAM250 | | | | | |
|-----------------------------|--------------------------|--------------------------|--------------------|--------------------------|----------------|
| Teste | Adaptação Média de P_0 | Adaptação Média de P_n | Número de Gerações | Cromossomo Mais Adaptado | Score BALIBASE |
| 1 | -38.2 | 682.444162436548 | 100 | 689.5 | 0.679 |
| 2 | -66.0112359550562 | 410.758883248731 | 90 | 422 | 0.274 |
| 3 | 9.78571428571429 | 521.708955223881 | 110 | 537.5 | 0.494 |
| 4 | -55.4759615384615 | 417.715384615385 | 60 | 424 | 0.295 |
| 5 | -11.2777777777778 | 647.025125628141 | 140 | 663 | 0.616 |
| 6 | 4.05454545454545 | 497.967032967033 | 90 | 510 | 0.458 |
| 7 | -40.5052083333333 | 538.6592039801 | 70 | 551.5 | 0.423 |
| 8 | 26.4857142857143 | 469.614795918367 | 70 | 480 | 0.342 |
| 9 | -44.4190476190476 | 483.211734693878 | 100 | 498 | 0.482 |
| 10 | -0.78217821782178 | 665.59595959596 | 80 | 676.5 | 0.649 |
| Média Score BALIBASE | | | | | 0.4712 |

Tabela 6.6: Teste família **laab** com matriz de substituição PAM250

A conclusão que se pode tirar com os testes sobre matriz de substituição é que em média a matriz DAYHOFF apresentou score BALIBASE mais elevado para a família protéica **laab**. Dessa forma, optou-se por utilizar esta matriz como parâmetro do GAADT para famílias de proteínas cujas características são parecidas com a família **laab**, segundo a classificação do BALIBASE.

6.2.2 Parâmetros Relacionados com Algoritmos Genéticos

O foco do GAADT não é, nesta instanciação, apresentar resultados em um curto intervalo de tempo. Entretanto, foram necessárias algumas limitações para que a execução do GAADT fosse possível no ambiente de execução.

6.2.2.1 Tamanhos da população e do cromossomo

O GAADT tem como característica aumentar o tamanho da população de uma geração para outra no início da execução, devido ao número de filhos gerados com o operador genético de cruzamento. O comportamento do GAADT é que, conforme a solução vai convergindo, o tamanho da população vai diminuindo.

No ambiente de testes não foi possível permitir que o GAADT executasse livremente de acordo com o seu comportamento natural, pois a quantidade de memória não era suficiente para armazenar todos os cromossomos. Além de exigir um grande espaço de armazenamento, uma população muito grande aumenta bastante o tempo para criar a próxima geração.

Os testes realizados com 1000 e 800 cromossomos demoraram muito para criar algumas poucas gerações e a execução foi interrompida antes do algoritmo começar a convergir. Já os testes com a população limitada em 400 cromossomos conseguiu se aproximar em média 60.0% do alinhamento referência, mas o tempo de execução era de 4 horas para cada teste. Limitando-se o tamanho da população em 200 cromossomos foi possível aproximar em média 71.48% do alinhamento referência com um tempo de execução menor. Dessa forma, optou-se então por limitar o tamanho da população em 200 cromossomos nos testes que se seguiram.

O tamanho do cromossomo é um parâmetro que não está relacionado com o tempo de execução do algoritmo, mas com o resultado que é fornecido. Para esta instanciação, o tamanho do cromossomo é limitado conforme o tamanho da maior seqüência de entrada. Quando a maior seqüência tem até 100 aminoácidos são adicionados 4 *gaps*. Nas demais seqüências é inserida a quantidade necessária para que todas as seqüências tenham o mesmo tamanho.

Para seqüência com mais de 100 aminoácidos o número de *gaps* é dado por:

$$Num_{gaps} = tam_{maior}/25.$$

No GAADT original não há limitação para o tamanho do cromossomo. Contudo, pôde-se observar que, mesmo fazendo esta limitação para o problema

de alinhamento múltiplo, os resultados foram satisfatórios, pois a equação acima normalmente gera mais *gaps* do que o necessário e, quando isso ocorre, o GAADT tende a produzir genes inócuos, os quais são suprimidos conforme especificado na definição 5.19.

6.2.2.2 População inicial

Para gerar a população inicial foi utilizada a seguinte heurística, que, em geral, fornece uma população inicial melhor que uma escolha completamente aleatória e mantém uma variabilidade genética suficiente para que o algoritmo não convirja muito rapidamente.

Heurística: são criados 50 cromossomos inserindo *gaps* aleatoriamente nas seqüências. Destes, são selecionados apenas 5 cromossomos que obtiverem melhor adaptação. Para gerar mais cromossomos é utilizada a função *troca* (seção 5.22) verificando se há melhora nos cromossomos iniciais. Esta operação se repete até que sejam encontrados 50 cromossomos para formar a população inicial.

Para vários problemas de otimização a população inicial é tratada antes de ser submetida à execução do algoritmo genético. Isso ocorre porque a população inicial interfere no tempo total de execução. Entretanto, os testes apresentados na seção 6.2.1 não indicam que uma boa população inicial garante uma execução com um menor número de gerações.

A tabela 6.6 é bastante ilustrativa nesse sentido, pois foi a que apresentou populações iniciais com adaptação média mais variadas. Os testes 7 e 8 apresentaram adaptação média da população inicial -40.50 e 26.49 aproximadamente, mas o número de gerações permaneceu o mesmo.

6.2.2.3 Método de escolha dos cromossomos pais

A definição 5.17 mostra que o cromossomo pai c_1 pertence ao conjunto *MACHO* e o cromossomo pai c_2 pertence ao conjunto *FEMEA*. Estes conjuntos são parte da subpopulação que está apta a sofrer a ação do operador de cruzamento.

Houve a necessidade de analisar se a reprodução deveria ser uma combinação de todos os cromossomos dois a dois, ou seja, reprodução assexuada com $M = F$, ou se os conjuntos deveriam ser particionados para uma reprodução sexuada, onde a metade da subpopulação pertenceria ao conjunto *MACHO* e a outra metade pertenceria ao conjunto *FEMEA*.

A tabela 6.7 mostra os testes realizados quando cada cromossomo é cruzado com todos os outros.

| Teste | Score BALI-BASE | Adaptação Média de P_0 | Adaptação Média de P_n | Número de Gerações | Cromossomo Mais Adaptado |
|-----------------------------|-----------------|--------------------------|--------------------------|--------------------|--------------------------|
| 1 | 0.354 | -6.21965317919075 | 445.897905759162 | 119 | 461 |
| 2 | 0.42 | -37.3581560283688 | 548.194736842105 | 88 | 555 |
| 3 | 0.682 | -9.36904761904762 | 706.397905759162 | 89 | 718.5 |
| 4 | 0.369 | -56.4130434782609 | 468.47193877551 | 69 | 473.5 |
| 5 | 0.318 | -79.1037735849057 | 496.141361256544 | 99 | 507 |
| 6 | 0.744 | -10.0939597315436 | 712.540983606557 | 79 | 717 |
| 7 | 0.824 | 12.5732984293194 | 750.192893401015 | 129 | 761.5 |
| 8 | 0.628 | 3.1830985915493 | 631.5 | 89 | 664.5 |
| 9 | 0.292 | -58.485401459854 | 448.942934782609 | 109 | 456.5 |
| 10 | 0.777 | -2.89795918367347 | 765.97193877551 | 119 | 775.5 |
| 11 | 0.461 | -35.9657142857143 | 530.7825 | 109 | 543 |
| 12 | 0.744 | -46.8861788617886 | 689.767857142857 | 99 | 700 |
| 13 | 0.438 | -36.6028368794326 | 506.391959798995 | 99 | 520 |
| 14 | 0.286 | -63.6565217391304 | 449.11282051282 | 69 | 461 |
| 15 | 0.393 | -29.2714285714286 | 501.917948717949 | 59 | 507 |
| 16 | 0.705 | 16.188679245283 | 659.770833333333 | 79 | 670 |
| 17 | 0.732 | -12.17578125 | 679.311855670103 | 49 | 697.5 |
| 18 | 0.717 | -1.60763888888889 | 710.486413043478 | 99 | 730 |
| 19 | 0.545 | -18.884328358209 | 562.638586956522 | 79 | 569.5 |
| 20 | 0.461 | -89.9959016393443 | 499.6775 | 59 | 511.5 |
| Média Score BALIBASE | | | | 0.5445 | |

Tabela 6.7: Teste Cruzamento: $M = F$

O melhor resultado foi obtido no teste 7, onde o alinhamento resultante se aproximou em 82.4% do alinhamento referência. No entanto, esta forma de realizar o cruzamento apresentou-se desvantajosa, pois, em média, os testes acertaram apenas 54.45% e o tempo médio de execução de cada teste foi de 3 horas.

Para melhorar o tempo de espera, a seleção dos cromossomos para cruzamento foi alterado para reprodução sexuada, dividindo os cromossomos aptos em duas partições e permitindo o cruzamento apenas entre cromossomos de partições diferentes. Isso diminuiu o total de comparações, o que diminui o tempo para criar uma nova geração. Além da diminuição do tempo de execução, a tabela 6.8 mostra os resultados usando este tipo de cruzamento.

O melhor resultado foi obtido no teste 11, cujo alinhamento resultante se aproximou em 86.6% do alinhamento referência. Pode-se observar que em média esta alteração no cruzamento obteve 71.485% de aproximação, uma melhora significativa sobre o primeiro método.

A diferença dessa nova reprodução para a assexuada é que a antiga fazia a população resultante ter muitos cromossomos parecidos, o que diminui a diversidade genética da população e acelera a convergência da mesma. Dessa forma, optou-se por prosseguir os testes com reprodução sexuada.

| Teste | Score BALI-BASE | Adaptação Média de P_0 | Adaptação Média de P_n | Número de Gerações | Cromossomo Mais Adaptado |
|-----------------------------|-----------------|--------------------------|--------------------------|--------------------|--------------------------|
| 1 | 0.729 | -16.5333333333333 | 681.676470588235 | 130 | 687 |
| 2 | 0.768 | -71.3170731707317 | 689.994871794872 | 120 | 698 |
| 3 | 0.61 | -48.5822784810127 | 655.879888268156 | 200 | 664.5 |
| 4 | 0.729 | -31.993670886076 | 677.497422680412 | 120 | 688.5 |
| 5 | 0.643 | 33.0412844036697 | 681.32320441989 | 70 | 690 |
| 6 | 0.735 | -36.4940476190476 | 714.794871794872 | 230 | 729 |
| 7 | 0.744 | -4.51485148514851 | 666.56 | 100 | 676 |
| 8 | 0.69 | -7.46236559139785 | 685.633165829146 | 130 | 695 |
| 9 | 0.693 | 27.9529914529914 | 695.755263157895 | 90 | 702 |
| 10 | 0.497 | -24.8173076923077 | 614.44414893617 | 90 | 632.5 |
| 11 | 0.866 | -27.35625 | 792.927860696517 | 789 | 798.5 |
| 12 | 0.759 | -83.1505376344086 | 700.725806451613 | 449 | 707 |
| 13 | 0.682 | -28.1363636363636 | 659.022857142857 | 89 | 669 |
| 14 | 0.753 | -36.5833333333333 | 750.503676470588 | 1229 | 750.503676470588 |
| 15 | 0.75 | -2.60493827160494 | 698.805970149254 | 529 | 704.5 |
| 16 | 0.747 | -64.4393939393939 | 707.485714285714 | 539 | 718 |
| 17 | 0.795 | -48.2012195121951 | 758.73275862069 | 99 | 775 |
| 18 | 0.738 | -43.3943661971831 | 727.817073170732 | 219 | 734.5 |
| 19 | 0.658 | -66.1091954022988 | 700.040880503145 | 869 | 714 |
| 20 | 0.711 | -31.3440860215054 | 698.726804123711 | 609 | 706 |
| Média Score BALIBASE | | | 0.71485 | | |

Tabela 6.8: Teste Cruzamento: $M = 50\%P$ e $F = 50\%P$

6.3 Resultados

A análise dos parâmetros a serem usados, mostrada na seção anterior, foi realizada com o intuito de selecionar os parâmetros que fornecem bons resultados para a família **1aab** e verificar se outras famílias de proteínas com características similares à família **1aab** também podem se beneficiar dessas escolhas.

Com todos os parâmetros fixados, o GAADT alinhou as famílias proteicas **1aab**, **1fjla**, **1hpi**, **1csy** e **1tgxa**. Tais famílias foram selecionadas por causa das características citadas na tabela 6.9. Os resultados com essas famílias é importante para mostrar que não ocorreu o problema de *overfitting* e que a análise realizada na seção anterior pôde ser generalizada.

| Família | Quantidade de Sequências | Tamanho do Alinhamento | Maior Sequência | Menor Sequência | Percentual de Identidade |
|--------------|--------------------------|------------------------|-----------------|-----------------|--------------------------|
| 1aab | 4 | 82 | 79 | 67 | 30 |
| 1fjla | 6 | 76 | 70 | 58 | 28 |
| 1hpi | 4 | 85 | 81 | 70 | 33 |
| 1csy | 5 | 110 | 104 | 100 | 30 |
| 1tgxa | 4 | 69 | 64 | 57 | 31 |

Tabela 6.9: Características das Famílias Protéicas

Para famílias proteicas com características muito diferentes da **1aab** não foram realizadas análises, pois não há indicações na literatura que sugerem que os parâmetros selecionados na seção anterior deveriam apresentar bons

resultados a elas. No caso específico da escolha da matriz de substituição, é conhecido que famílias protéicas com percentual de identidade diferentes devem utilizar diferentes matrizes de substituição.

Os resultados do GAADT e das outras ferramentas são mostrados na tabela 6.10.

| Família | 1aab | 1fj1A | 1hpi | 1csy | 1tgxA |
|------------|-------------|--------------|-------------|-------------|--------------|
| GAADT | 0.881 | 0.814 | 0.708 | 0.703 | 0.692 |
| PRP | 1.000 | 1.000 | 0.918 | 0.949 | 0.895 |
| CLUSTALX | 1.000 | 1.000 | 0.861 | 1.000 | 0.806 |
| SAGA | 0.823 | 0.993 | 0.916 | 0.969 | 0.873 |
| DIALIGN | 1.000 | 1.000 | 0.785 | 0.980 | 0.871 |
| SB_PIMA | 1.000 | 1.000 | 0.909 | 0.976 | 0.782 |
| ML_PIMA | 1.000 | 1.000 | 0.909 | 0.968 | 0.768 |
| MULTIALIGN | 1.000 | 1.000 | 0.890 | 0.981 | 0.819 |
| PILEUPS | 1.000 | 1.000 | 0.859 | 0.932 | 0.718 |
| MULTAL | 1.000 | 1.000 | 0.852 | 0.935 | 0.651 |
| HMMT | 0.214 | 0.436 | 0.962 | 0.779 | 0.556 |

Tabela 6.10: Resultado Final

A análise do resultado permitiu verificar que o GAADT, mesmo para esta instanciamento que não especificou por completo a proposta de Vieira (2003), conseguiu se aproximar de forma significativa do resultado final, apresentando regularidade nas respostas.

Ainda há muitas melhorias a serem feitas para permitir que a convergência do algoritmo seja natural, tais questões serão mostradas na seção de trabalhos futuros (seção 7.2).

Capítulo 7

Conclusão

7.1 Contribuições e Relevância

Neste trabalho foram apresentados um modelo de alinhamento múltiplo de proteínas utilizando algoritmos genéticos baseados em tipos abstratos de dados e a análise estatística deste modelo de alinhamento comparada com outras ferramentas já consolidadas para realizar a mesma tarefa.

Apesar de haver ferramentas consolidadas para realizar a tarefa de alinhamento múltiplo, pesquisas continuam em andamento na busca de encontrar novas heurísticas para resolver esta tarefa, o que é demonstrado pela quantidade de artigos sobre alinhamentos apresentados todos os anos nas principais revistas de bioinformática. Isto faz com que haja sempre a avaliação dos algoritmos propostos verificando as potenciais vantagens e desvantagens, como mencionado por Nuin et al. (2006).

Vale ressaltar, que o alinhamento realizado pela presente instanciação do GAADT não faz uso de conhecimentos adicionais sobre a família das proteínas investigadas, tais como inferência de proximidade entre as seqüências pela construção de árvores guia para a matriz de mérito e inferência filogenética. Isto indica que a instanciação do GAADT pode ser usada para fazer alinhamento múltiplo de famílias de proteínas desconhecidas ou para aquelas que se tem conhecimento incompleto sobre as mesmas.

Embora características adicionais acerca da família protéica não estejam implementadas nesta instanciação do GAADT elas podem ser incorporados em outras instanciações como parâmetros de entrada ou interno do sistema. No entanto, a análise estatística mostra que a forma que o modelo foi implementado é capaz de resolver o problema de alinhamento múltiplo de proteínas, apesar das limitações impostas ao modelo pelo ambiente de testes. Acredita-se que, em ambientes com melhor potencial de recursos computacionais, os

resultados do GAADT seriam superiores ao que foram apresentados na seção 6.3.

As decisões que foram tomadas para que os resultados fossem obtidos em um ambiente computacional de pouca capacidade servirão como ponto de partida nas próximas instanciações do GAADT para outros problemas.

7.2 Sugestões de Trabalhos Futuros

As limitações apresentadas no capítulo 6 afetaram diretamente o percurso do algoritmo, não permitindo, em vários casos, que ele convergisse a uma única solução. Dessa forma, esforços serão necessários para que o modelo se aproxime cada vez mais da especificação original do GAADT. Como trabalhos futuros sugere-se:

- Implementar maneiras para que não seja necessário limitar o tamanho da população, deixando que ela se expanda conforme o comportamento natural do GAADT, assim se manteria a variabilidade genética e possíveis soluções não seriam perdidas;
- Não fixar o tamanho do cromossomo, pois, além de ser uma limitação para o algoritmo, fez com que o operador de mutação fosse aplicado apenas para inserção de genes inócuos (definição 5.18) e de troca das posição que contém *gaps* (definição 5.22). Dois efeitos da liberação do tamanho do cromossomo precisariam ser trabalhados para as próximas instanciações do GAADT:
 - O operador genético de cruzamento deverá ser revisto, pois a implementação utilizada neste trabalho apenas é válida para cromossomos de mesmo tamanho.
 - Implementar o operador de mutação para a inserção de *gaps* em posições arbitrárias nas seqüências;
- A técnica de soma dos pares (Sperschneider 2008) utilizada neste trabalho é uma das formas mais simples de pontuar a coluna do alinhamento múltiplo. Para futuras análises sugere-se melhorar a função de adaptação utilizando técnicas propostas em trabalhos mais recentes, tais como: T-COFFEE (Notredame et al. 2000), M-COFFEE (Wallace et al. 2006), MAFFT (Kato et al. 2005) e DIALIGN (Subramanian et al. 2005) e (Subramanian et al. 2008).

Além disso, sugere-se a criação de uma outra instanciação do GAADT que utilize técnicas de raciocínio baseado em casos para especificar os parâmetros, tais como matrizes de substituição e pesos para penalidade de *gap*, o que facilitaria gerenciar o conhecimento adicional de proximidade entre as seqüências da família.

Um última linha de pesquisa sugerida envolve a utilização de uma arquitetura paralela, onde cada nó da arquitetura poderia receber uma parte dos cromossomos da população atual e ficar responsável por aplicar os operadores genéticos. Isto seria eficaz nas execuções de futuras instanciações, contemplando as limitações computacionais e problemas com o tempo de resposta.

Referências Bibliográficas

- Altschul, S. (1989), 'Gap costs for multiple sequence alignment', *J. Theor. Biol* **138**, 297–309.
<http://www.pnas.org/cgi/content/abstract/86/12/4412>.
- Branden, C. & Tooze, J. (1991), *Introduction to protein structure*, Garland Publishing, New York.
- Brown, T. A. (1999), *Genética Um enfoque Molecular*, Guanabara Koogan.
- Chargaff, E. (1950), 'Chemical specificity of nucleic acids and mechanism of their enzymatic degradation', *Experientia* **6**, 201–209.
- Davis, L. (1991), *Handbook of Genetic Algorithms*, Van Nostrand Reinhold.
- Edgar, R. C. (2004), 'Muscle: mutiple sequence alignment with high accuracy and high throughput', *Nucleic Acids Research* **32**(5), 1792–1799.
- Feng, D. F. & Doolittle, R. F. (1987), 'Progressive sequence alignment as a prerequisite to correct phylogenetic trees', *Journal of Molecular Evolution* **25**, 351–360.
- Fischer, S. (2001), 'Course and exercise sequencing using metadata in adaptive hypermedia learning systems', *Journal on Educational Resources in Computing (JERIC) - ACM* **1**(1), 5.
- Fitch, W. M. & Margoliash, E. (1967), 'Construction of phylogenetic trees.', *Science* **155**, 279–284.
- Garnier, J., Gibrat, J. & B. Robson (1998), 'Gor method for predicting protein secondary structure from amino acid sequence', *Methods Enzymol.* **266**, 540–553.
- Gibas, C. & Jambeck, P. (2001), *Desenvolvendo Bioinformática*, Campus.
- Goldberg, E. D. (1989), *Genetic algorithms in search, optimization and machine learning*, Addison-Wesley Longman Publishing Co.

- Golubchik, T. & M. J. Wise, S. Easteal, L. S. J. (2007), 'Mind the gaps: evidence of bias in estimates of multiple sequence alignments', *Mol. Biol. Evol.* **24**(11), 2433–2475.
- Gotoh, O. (1996), 'Significant improvement in accuracy of multiple protein sequence alignments by iterative refinement as assessed by reference to structural alignments', *Journal of Molecular Biology* **264**, 823–838.
- Griffiths, A. J. F., Gelbart, W. M., Miller, J. H. & Lewontin, R. C. (2001), *Genética Moderna*, Guanabara Koogan.
- Henikoff, S. & Henikoff, J. G. (1991), 'Amino acid substitution matrices from protein blocks', *Nucleic Acids Research* **19**(23), 6565–6572.
- Higgins, D. G. & Sharp, P. M. (1988), 'Clustal: A package for performing multiple sequence alignment on a microcomputer', *Gene* **73**, 237–244.
- Isaev, A. (2004), *Introduction to Mathematical Methods in Bioinformatics*, Universitext, Springer, springeronline.com.
- Katoh, K., Kuma, K., Toh, H. & Miyata, T. (2005), 'Mafft version 5: improvement in accuracy of multiple sequence alignment', *Nucleic Acids Research* **33**(2), 511–519.
- Kimura, M. (1980), 'A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences', *Journal of Molecular Evolution* **16**, 111–120.
- Krogh, A., Brown, M., Mian, I. S., Sjölander, K. & Haussler, D. (1994), 'Hidden markov models in computational biology. applications to protein modeling', *Journal of Molecular Biology* **235**, 1501–1531.
- Larking, M. A., Blackshields, G., Brown, N. P., and P. A. McGettigan, R. C., McWilliam, H., Valentim, F., Wallace, I. M., Wilm, A., Lopez, R., Thompson, J. D., Gibson, T. J. & Higgins, D. G. (2007), 'Clustal w and clustal x version 2.0', *Bioinformatics* **23**(21), 2947–2948.
- Mongenstern, B., French, K., Dress, A. & Werner, T. (1998), 'Dialign: finding local similarities by multiple sequence alignment.', *Bioinformatics* **14**(3), 290–294.
- Moreira, M. A. (1980), 'Mapas conceituais como instrumentos para promover a diferenciação conceitual progressiva e a reconciliação integrativa.', *Ciência e Cultura* **32**(13), 474–479.

- Mount, D. (2001), *Bioinformatics: Sequence and Genome Analysis*, Cold Spring Harbor Laboratory Press.
- Needleman, S. B. & Wunsch, C. D. (1970), 'A general method applicable to the search for similarities in the amino acid sequence of two proteins', *Journal of Molecular Biology* **48**, 443–453.
- Notredame, C. & Higgins, D. (1996), 'Saga: Sequence alignment by genetic algorithm', *Nucleic Acids Res.* **24**, 1515–1524.
- Notredame, C., Higgins, D. G. & Heringa, J. (2000), 'T-coffee: A novel method for fast and accurate multiple sequence alignment', *J. Mol. Biol.* **302**, 205–217.
- Novak, J. D. (1998), *Learning Creating and Using Knowledge: Conceptual Maps as Facilitative Tools in Schools and Corporations*, Lawrence Erlbaum Associates, Inc., New Jersey.
- Novak, J. D. & Gowin, D. (1999), *Aprender a Aprender*, 2 edn, Edições Técnicas, Lisboa.
- Nuin, P. A., Wang, Z. & Tillier, E. R. (2006), 'The accuracy of several multiple sequence alignment programs for proteins', *BMC Bioinformatics* **24**(7), 471–489.
- Pevsner, J. (2003), *Bioinformatics and Functional Genomics*, Wiley-Liss.
- Rocha, F. E. L., Vieira, R. V., Jr, J. V. C. & Favero, E. L. (2004), 'Especificação de um algoritmo genético para auxiliar na avaliação da aprendizagem significativa com mapas conceituais', *Simpósio Brasileiro de Informática na Educação - SBIE* **1**, 3.
- Santos, D. F., Dias, U., Lopes, R. V. V. & Lopes, M. A. (2006), 'A genetic algorithm based on abstract data types to the multiple protein alignment problem', IX Encontro de Modelagem Computacional. Publicado em meio impresso durante o congresso.
- Sen, T., Jernigan, R., Garnier, J. & Kloczkowski, A. (2005), 'Gor v server for protein secondary structure prediction', *Bioinformatics* **21**, 2787–2788.
- Setubal, J. C. & Meidanis, J. (1997), *Introduction to Computational Molecular Biology*, Books/Cole Publishing Company.
- Sperschneider, V. (2008), *Bioinformatics Problem Solving Paradigms*, Springer.

- Subramanian, A. R., Kaufmann, M. & Morgenstern, B. (2008), 'Dialign-tx: greedy and progressive approaches for segment-based multiple sequence alignment', *Algorithms Mol. Biol.* **27**, 3–6.
- Subramanian, A. R., Weyer-Menkhoff, J., Kaufmann, M. & Morgenstern, B. (2005), 'Dialign-t: an improved algorithm for segment-based multiple sequence alignment', *BMC Bioinformatics* **22**, 6–66.
- Thompson, J. D., Higgins, D. G. & Gibson, T. J. (1994a.), 'Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice', *Nucleic Acids Res.* **22**, 4673–4680.
- Thompson, J. D., Plewniak, F. & Poch, O. (1999a), 'Balibase: a benchmark alignment database for the evaluation of mutiple alignment programs', *Bioinformatics* **15**(1), 87–88.
- Thompson, J. D., Plewniak, F. & Poch, O. (1999b), 'A comprehensive comparison of multiple sequence alignment programs', *Nucleic Acids Research* **27**(13), 2682–2690.
- Vieira, R. (2003), Um Algoritmo Genético Baseado em Tipos Abstratos de Dados e sua Especificação em Z, PhD thesis, Universidade Federal de Pernambuco, Recife, Pernambuco, Brasil.
- Wallace, I. M., O'Sullivan, O., Higgins, D. G. & Notredame, C. (2006), 'M-coffee: combining multiple sequence alignment methods with t-coffee', *Nucleic Acids Research* **34**(6), 1692–1699.
- Watson, J. & Crick, F. H. C. (1953), 'Molecular structure of nucleic acids: a structure for deoxyribose nucleic acid', *Nature* **171**, 737–738.

Assinatura do Aluno

Assinatura do Orientador