THAÍS HELENA SAMED E SOUSA

**COMPUTAÇÃO INTELIGENTE NO ESTUDO DE
VARIANTES DE HEMOGLOBINA**

Dissertação apresentada ao Programa de Pós-Graduação Interunidades em Bioengenharia - Escola de Engenharia de São Carlos/Faculdade de Medicina de Ribeirão Preto/ Instituto de Química de São Carlos da Universidade de São Paulo, para obtenção do título de Mestre em Bioengenharia.

ORIENTADOR: Prof. Dr. Alexandre C. B. Delbem

São Carlos
2004

Lista de assinaturas

Dedico este trabalho ao meu Marido, Thales pela constante confiança, pelo necessário incentivo e pelo grande amor.

À minha carinhosa mãe pelo amor e imensa compreensão.

AGRADECIMENTOS

Ao professor Dr. Alexandre Delbem, meu orientador, pela amizade, pela orientação paciente e dedicação constante, fundamental para a conclusão deste trabalho.

Ao professor Otávio Thiemann, do IFSC, pelo incentivo e sábias observações a respeito da idéia original do trabalho.

Ao professor Richard Charles Garratt, do IFSC, pelas constantes explicações, idéias e sugestões muito valiosas no decorrer de todo o trabalho.

Ao doutorando do ICMC, Cláudio Policastro por sua indispensável ajuda e dedicação a este trabalho.

À Janete Ferreira Rodrigues dos Santos e a Melaine Priscila Fidelix, não somente pelos trabalhos prestados da secretária da pós-graduação, mas principalmente pela sincera amizade.

Aos colegas da Bioengenharia e em especial a Elizete, pelas horas de companheirismo e de paciência.

Às amigas com quem convivi: Kátia, Soraia e Vália Billota que sempre fizeram muito por mim.

Ao meu irmão Fuad e minha cunhada Márcia pelo animado apoio.

Às minhas tias Leila e Irani por seus pensamentos positivos e suas orações.

Aos alunos da graduação do ICMC, Vinicius Ferreira Negrison e Rodrigo Nishihara Adão, pela grande ajuda com a implementação do algoritmo genético.

À CAPES pelo apoio financeiro.

Por fim, agradeço a todos que de forma direta ou indireta contribuíram para o desenvolvimento deste trabalho.

“Só existem dois dias no ano que nada pode ser feito. Um se chama ontem e o outro se chama amanhã, portanto, hoje é o dia certo para amar, acreditar, fazer e principalmente viver”.

Dalai Lama.

RESUMO

SOUSA, T. H. S. (2004). Computação Inteligente no Estudo de Variantes de Hemoglobina. Dissertação de Mestrado - Escola de Engenharia de São Carlos/ Faculdade de Medicina de Ribeirão Preto/ Instituto de Química de São Carlos, Universidade de São Paulo.

A evolução *in vitro* é um método laboratorial criado para a evolução de moléculas, principalmente de proteínas. Por meio de mutações, o método busca novas propriedades de moléculas, objetivando criar novas proteínas e, com isso, intensificar o estudo e a cura de doenças, pelo desenvolvimento de novos fármacos. O grande desafio na evolução *in vitro* é criar o maior número possível de moléculas de proteínas que atinjam propriedades desejadas, uma vez que apenas uma fração infinitesimal das diversidades geradas utilizando-se seqüências de DNA é aproveitada. Para se obter moléculas com funcionalidade adequada por meio dessa técnica, é requerido muito tempo e aporte financeiro. Com o objetivo de avaliar computacionalmente a funcionalidade de proteínas variantes a partir das seqüências de aminoácidos buscando reduzir o custo e o tempo despendido em laboratório, este trabalho propõe o uso de técnicas de computação inteligentes (evolução *in silicio*¹), baseadas em aprendizado de máquina e computação evolutiva. Para o emprego de técnicas de AM, bancos de dados com elevado número de informações são fundamentais. Neste sentido, escolheu-se investigar as moléculas mutantes de hemoglobina, uma vez que a quantidade de informações disponíveis sobre a mesma é bastante extensa na literatura. Os resultados obtidos mostram que é possível desenvolver algoritmos eficientes para determinar a funcionalidade de variantes de hemoglobina. Com esses resultados, busca-se contribuir no desenvolvimento de técnicas de evolução dirigida com suporte computacional.

Palavras-chave: Aprendizado de máquina. Hemoglobina. Seqüências mutantes.

¹ Processo de evolução de moléculas simulado em computador.

ABSTRACT

SOUSA, T. H. S. (2004). Intelligent Computation applied to the study of hemoglobin variants. M.Sc. - Escola de Engenharia de São Carlos/ Faculdade de Medicina de Ribeirão Preto/ Instituto de Química de São Carlos, Universidade de São Paulo.

In vitro evolution is a laboratorial method developed to molecule evolution mainly proteins. By producing mutations, this method looks for new molecule properties, aiming achieve new proteins for the development of drugs for diseases. The great challenge of *in vitro* evolution is the development of the highest possible number of molecules that reaches desired properties. This objective is a great challenge to be transposed, since only one infinitesimal fraction of generated proteins using DNA sequences is usefull to obtain molecules with the desired function. Besides high financial support and time are required to apply this technique. With the objective of evaluating computacionaly and functionality of proteins mutants starting from aminoacids sequences looking for to reduce the cost and the time loosened at laboratory, this work proposes the use of intelligent computation techniques based on learning of it conspires and evolutionary computation. On the other hand, when machine learning techniques are used, it is fundamental to access data mining with high number of information. In order to reduce these difficulties, this work proposes a machine learning (ML) based on approach to evaluate computationally hemoglobin variants. ML techniques require, in general, large data base. In order to supply this requirement, hemoglobin variants were used because there is a large number of hemoglobin variants available in the literature. The obtained results shown that is possible to develop efficient algorithms to determine hemoglobin variant function. These results can contribute for development of molecule evolution techniques.

Key words: Machine Learning. Hemoglobin. Mutant Sequences.

LISTA DE FIGURAS

Figura 2.1 – Esquema representativo da estrutura de um aminoácido.....	19
Figura 2.2 - Esquema representativo de uma ligação peptídica.	19
Figura 2.3 - Uma proteína fibrosa. Molécula de colágeno	21
Figura 2.4 - Proteínas Globulares com estruturas em α -hélice e folha β	22
Figura 2.5 - (a) Proteína Fibrinogênio. (b) Proteína Miosina.....	23
Figura 2.6 - Esquema representativo da estrutura primária de proteína	24
Figura 2.7 - (a) Estrutura em folha β pregueada. (b) Estrutura em α -hélice	25
Figura 2.8 - Esquema representativo da estrutura terciária de proteína.....	26
Figura 2.9 – Esquema representativo da estrutura quaternária de proteína	27
Figura 2.10 – Forças que estabilizam a estrutura terciária das proteínas.....	32
Figura 3.1 - Figura representativa de uma célula sanguínea.	36
Figura 3.2 - Região de contato entre $\alpha_2\beta_2$ envolvendo as hélices B, o dobramento GH e as hélices G	38
Figura 3.3 - Interface de contato entre as subunidades $\alpha_1\beta_2$ entre os dobramentos FG e as Hélices C	38
Figura 3.4 - Esquema representativo da molécula de mioglobina	39
Figura 3.5 - Esquema representativo do grupo heme	40
Figura 3.6 - Representação esquemática dos estados tenso e relaxado	41
Figura 3.7 - Deso-xihemoglobina	41
Figura 3.8 - Oxihemoglobina.....	42
Figura 3.9 - Curva de saturação da mioglobina e da hemoglobina	43
Figura 3.10 - Representação de uma célula sanguínea com anemia falciforme	46
Figura 4.1 - Árvore de decisões	53
Figura 4.2 - Regras de conhecimento utilizadas.....	53
Figura 4.3 - Representação esquemática da Tabela 4.2.....	56
Figura 4.4 - Representação esquemática da Tabela 4.3.....	58
Figura 4.5 - Representação esquemática da Tabela 4.4. Passo 3.	59
Figura 5.1 - Apresentação do programa.....	67
Figura 5.2 - Seqüência de aminoácidos da hemoglobina	68
Figura 5.3 - Definição dos atributos no arquivo de entrada.....	69
Figura 5.4 - Campo de dados do arquivo de entrada.....	70
Figura 5.5 - Exemplo do final de linhas no campo de dados, mostrando a classe de cada variante	70
Figura 5.6 - (a) Tela de inicialização do programa (b) Busca do arquivo.	71
Figura 5.7 - Figura (a) Escolha do arquivo a ser aberto Figura (b) Arquivo aberto	72
Figura 5.8 - (a) Tela para escolha do teste estatístico para análise dos resultados e (b) Tela para escolha do algoritmo de AM	72
Figura 5.9 - (a) Resultados do algoritmo C4.5 mostrando a AD formada (b) Análise estatística produzida pelo C4.5.....	73
Figura 6.1 - Seqüências de hemoglobina mutante (denominada Niigata)	75
Figura 6.2 - Exemplo de seqüência de aminoácido alterada pelo Filtro 1	82
Figura 6.3 - Processo de alteração das seqüências de acordo com o Filtro 5	85
Figura 6.4 - A alteração das seqüências produzidas pelo Filtro 6.....	86
Figura 6.5 - A alteração do arquivo de entrada produzido pelo Filtro 7	87
Figura 6.6 - Processo de alteração das seqüências de acordo com o Filtro 8	88

LISTA DE TABELAS

Tabela 2.1 – Exemplo de proteínas conjugadas	20
Tabela 3.1 – Algumas Hemoglobinas Variantes	44
Tabela 4.1 – Conjunto de Instâncias	55
Tabela 4.2 – Construindo uma AD a partir do conjunto de exemplos. Passo 1	57
Tabela 4.3 – Construindo uma AD a partir do conjunto de exemplos. Passo 2	58
Tabela 4.4 – Podando a AD. Passo 3	59
Tabela 6.1 – Resultados produzidos pelos C4.5, SMO, RF, <i>Kstar</i> e FLR.....	77
Tabela 6.2 – Resultados do Teste 13 considerando ambientes químicos e a cadeia α de indivíduos heterozigotos.	78
Tabela 6.3– Resultados do Teste 14 considerando ambientes químicos e a cadeia β de indivíduos homozigotos	79
Tabela 6.4 –Resultados do Teste 15 considerando ambientes químicos e a cadeia β de indivíduos heterozigotos.....	79
Tabela 6.5 –Resultados do Teste 16 considerando ambientes químicos e a cadeia β de indivíduos heterozigotos e as principais posições de acordo co BASHFORD (1987).....	80
Tabela 6.6 – Resultados do Teste 17 considerando ambientes químicos e a cadeia β de indivíduos homozigotos e as principais posições de acordo co BASHFORD (1987).....	80
Tabela 6.7 – Resultados do Teste 18 considerando ambientes químicos e a cadeia α de indivíduos heterozigotos e as principais posições de acordo co BASHFORD (1987).....	80
Tabela 6.8 – Resultados do Teste 19 considerando ambientes químicos, a cadeia α e a cadeia β de indivíduos homozigotos e heterozigotos com as principais posições de acordo co BASHFORD (1987)..	81
Tabela 6.9 - Resultados do Teste 20 considerando o Filtro 1	82
Tabela 6.10 - Resultados do Teste 21 considerando o Filtro 2	83
Tabela 6.11 - Resultados do Teste 22 considerando o Filtro 3	83
Tabela 6.12 - Resultados do Teste 23 considerando o Filtro 4	84
Tabela 6.13 - Resultados do Teste 24	85
Tabela 6.14 - Resultados do Teste 25	86
Tabela 6.15 - Resultados do Teste 26	87
Tabela 6.16 - Resultados do Teste 27	88

LISTAS DE SIGLAS E ABREVIATURAS

DNA - Ácido Desoxirribonucleico

RNA_m – Ácido Ribonucleico mensageiro

3D – Tridimensional

KD – Kilodaltons

nm – Nanômetro

N-terminal – Nitrogênio terminal

C-terminal – Carbono terminal

mm – Milímetros

CO₂ - Gás Carbônico

O₂ – Oxigênio

Hb – Hemoglobina

pO₂ - Pressão parcial do oxigênio

mmhg – Milímetros de Mercúrio

MLP – *Multi Layer Perceptron*

SVM – *Support Vector Machines*

AM – Aprendizado de Máquina

IA – Inteligência Artificial

ADs – Árvores de Decisão

AGs - Algoritmos Genéticos

pH – Potencial Hidrogeniônico

LISTA DE SÍMBOLOS

α - Alfa

β - Beta

$\text{\AA}$ – Angstrom

δ - Delta

ζ - Zeta

ε - Epsilon

γ - Gama

R – Radical

SUMÁRIO

RESUMO

ABSTRACT

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS E ABREVIATURAS

LISTA DE SÍMBOLOS

SUMÁRIO

CAPÍTULO 1 – INTRODUÇÃO

CAPÍTULO 2 - PROTEÍNAS

2.2 A Composição das Proteínas	18
2.3 A Conformação das Proteínas	21
2.4 Níveis de Estruturas Protéicas	23
2.5 Determinação da Estrutura das Proteínas	27
2.6 Aminoácidos.....	28
2.6.1 Os aminoácidos comuns das proteínas	28
2.6.2 Localização das cadeias laterais	30
2.6.3 Ligações entre Aminoácidos	30
2.6.4 Aminoácidos Raros das Proteínas.....	32
2.6.5 Aminoácidos não Protéicos.....	32
2.6.6 Reação Química dos Aminoácidos.....	33
2.6.7 Nomenclatura dos Aminoácidos	33
2.7 Conformação Tridimensional	34

CAPÍTULO 3 - HEMOGLOBINA

3.2 Estrutura da Hemoglobina	36
3.3 Grupo Heme	39
3.4 Mecanismo de Cooperatividade.....	40
3.5 Mutações.....	43
3.6 Hemoglobinas Mutantes ou Variantes.....	44
3.7 Alterações Mutacionais na Sequência de Aminoácidos de uma Espécie	45
3.8 Proteínas Alostéricas	46

CAPÍTULO 4 - APRENDIZADO DE MÁQUINA

4.2 Árvores de Decisão	49
4.2.1 Indução de Árvores de Decisão	50
4.2.2 Poda	51
4.2.3 Classificação.....	52
4.3 Exemplo da construção de uma AD	52
4.4 Técnica <i>Support Vector Machines</i> (SVMs).....	58
4.5 Descrição do Algoritmo Genético Criado para o Trabalho.....	59

CAPÍTULO 5 - MATERIAIS E MÉTODOS

5.2 WEKA	63
5.2.1 Organização dos Dados de Entrada	65
5.2.2 Procedimento para a Geração do Programa	69

CAPÍTULO 6 - RESULTADOS

6.2 Resultados com o AG Proposto	81
--	----

CAPÍTULO 7 – CONCLUSÃO

REFERÊNCIA BIBLIOGRÁFICA

APÊNDICE

APÊNDICE A
APÊNDICE B
APÊNDICE C
APÊNDICE D
APÊNDICE E
APÊNDICE F

CAPÍTULO 1 – INTRODUÇÃO

O que há de mais íntimo na natureza dos seres vivos é a dinâmica molecular das células. No mundo do infinitamente pequeno, no microscópio ambiente intracelular, há uma dança constante entre íons e moléculas, bem como um interminável jogo de potenciais elétricos que justificam o estado físico-químico típico dos sistemas vivos e que os diferencia de qualquer corpo bruto. A química da célula, que vai da simplicidade das moléculas inorgânicas à extrema complexidade das macromoléculas, explica o grande mistério da vida (SOARES, 1999).

As proteínas são componentes fundamentais de todos os seres vivos. Mesmo os vírus, as formas mais simples de vida, contêm proteínas em sua estrutura. Moléculas de proteínas são constituídas por centenas ou mesmo milhares de moléculas menores chamadas aminoácidos, ligados em sequência como os elos de uma corrente. Uma proteína pode ser definida, portanto, como uma sequência de aminoácidos encadeados. As moléculas de proteínas são relativamente grandes, quando comparada a outras e, portanto, classificadas como macromoléculas (AMABIS E MARTHO, 1998).

Segundo SOARES (1999), as proteínas são os compostos orgânicos mais numerosos na composição química do protoplasma¹ celular. São responsáveis em grande parte pela função estrutural e arquitetônica da célula. Toda proteína é formada pela polimerização (ligação em cadeia) de numerosos aminoácidos. Centenas de aminoácidos combinam-se para formar uma única molécula proteica. O organismo humano tem a capacidade de sintetizar a maioria dos aminoácidos comuns em reações de transaminação². Essas reações transcorrem no fígado. Alguns aminoácidos,

¹ O protoplasma é o conteúdo do interior das células.

² Reação em que ocorre a transformação entre um aminoácido e um cetoácido (composto que contém os grupos carbonila e carboxila).

entretanto, não são sintetizados pelo organismo humano e precisam ser constantemente obtidos pela alimentação. Esses são chamados de aminoácidos essenciais.

Atualmente, são conhecidas as seqüências de aminoácidos de mais de duas mil proteínas. O fato marcante é que cada proteína tem uma seqüência de aminoácidos própria, definida com precisão. As seqüências de aminoácidos são importantes por vários motivos: primeiro, o conhecimento da seqüência de uma proteína é útil, ou mesmo essencial, para a elucidação de seu mecanismo de ação (por exemplo: o mecanismo de catálise de uma enzima). Segundo, as análises das relações entre seqüências de aminoácidos e estruturas tridimensionais de proteínas estão revelando as regras que comandam o enovelamento das cadeias polipeptídicas. A seqüência de aminoácidos é o elo entre a mensagem genética no DNA e a estrutura tridimensional que executa a função biológica de uma proteína. Uma notável característica das proteínas é a existência de uma estrutura tridimensional bem definida. Uma cadeia polipeptídica distendida ou disposta aleatoriamente não tem atividade biológica (STRYER, 1988).

Acredita-se que as proteínas doam-se em um estado onde sua energia potencial ou sua energia livre é mínima. A função das posições dos seus átomos pode ser calculada, utilizando para tanto, modelos chamados de campo de forças. A energia calculada por um campo de forças pode ser minimizada e, desta maneira, pode-se encontrar qual é o estado de mínima energia, prevendo então, qual a forma tridimensional da proteína a partir de sua seqüência de aminoácidos (WERHLI E LEMKE, 2003). A função de uma proteína é conseqüência da sua conformação, que é a disposição tridimensional de seus átomos formando uma estrutura (STRYER, 1988).

A busca de novas drogas e a cura ou tratamento de doenças genéticas têm sido largamente pesquisadas nos dias de hoje. Neste contexto, o entendimento das funções das proteínas é essencial. A determinação da estrutura é fundamental para a compreensão da função de uma determinada proteína. Esta por sua vez, é essencial para o desenvolvimento de fármacos.

Uma outra abordagem importante para o descobrimento de novas drogas, que permitam a cura de muitas doenças de origem genética ou doenças em que ainda não se conhece direito a origem, tem sido a evolução de proteínas *in vitro* (STEMMER, 1994), (ver Apêndice A). O conhecimento das proteínas existentes e a criação de novas proteínas têm sido possível por meio do conhecimento da seqüência de aminoácidos de tais proteínas e também da estrutura 3D das mesmas. Surgem, assim, cada vez mais

possibilidades de se descobrir por meio do estudo das seqüências de aminoácidos, da estrutura 3D das proteínas e de técnicas de evolução *in vitro* a causa de muitas doenças e talvez a cura para outras doenças.

Vários estudos SUN (2003), STEMMER (1994), ZHAO E ARNOLD (1997), ZHANG et al. (1997) e KLUG E FAMULOK (1994) têm mostrado a evolução *in vitro* como um método laboratorial para a evolução de moléculas com propriedades desejadas. Esse método é comumente conhecido como seleção *in vitro*, evolução *in vitro* ou Selex (Evolução Sistemática de Ligações Exponenciais). Esse método tem sido aplicado com grande sucesso em vários estudos biológicos, tais como a interação entre DNA e Proteína, propriedades catalíticas de moléculas de RNA e propriedades catalíticas de moléculas individuais de DNA (FITZWATER E POLISKY, 1996) (GOLD et al., 1995). É uma técnica bastante desenvolvida em laboratórios de todo o mundo, principalmente nos países europeus e norte-americanos onde, com a ajuda de tecnologia e apoio financeiro, têm ocorrido grandes descobertas. No entanto, o custo para a promoção da técnica em laboratórios ainda é bastante alto e requer desprendimento de muito tempo.

Alguns trabalhos envolvem técnicas de AM e estruturas primárias de proteínas. TSUNODA E LOPES (2003), desenvolveram um trabalho utilizando uma técnica de AM, em especial AGs, para a análise e detecção de seqüências idênticas de aminoácidos em enzimas com o intuito de implementar uma classificação automática. SELBIG et al. (1991), aplicaram técnicas de AM, em especial ADs, para selecionar propriedades nos aminoácidos que fossem as mais significativa, descrevendo uma posição na seqüência de aminoácidos que também fosse determinada por tal característica. MADDOURI E ELLOUMI (2000), os autores utilizaram um algoritmo de aprendizado desenvolvido por eles mesmos para fazer a classificação de seqüências biológicas baseados em um grande banco de dados. BARISIC et al. (2002), utilizaram dados analíticos bioquímicos, morfológicos e parâmetros anamnéticos para juntamente com os métodos de AM, que neste caso foi o uso do algoritmo C4.5 presente no Software WEKA para quantificar os fatores de risco em pacientes assintomáticos de uma patologia denominada amiloidose, causada pela hemodiálise.

Buscando reduzir custo e o tempo necessário em laboratório, este plano de trabalho propõe a utilização de computadores buscando determinar a funcionalidade de proteínas mutantes a partir das seqüências de aminoácidos.

A determinação da funcionalidade da proteína computacionalmente (*in silício*) seria uma etapa para se poder desenvolver um processo de evolução de proteínas *in silício*. Processos evolutivos podem ser simulados computacionalmente utilizando procedimentos denominados Algoritmos Evolutivos (CARVALHO, 2003). Esses algoritmos podem produzir indivíduos (no caso, proteínas) cada vez mais adequados a uma finalidade de forma relativamente rápida. No entanto, é necessário uma forma eficiente de avaliar os indivíduos gerados pelo algoritmo. Neste sentido, este trabalho investiga técnicas computacionais avançadas buscando verificar o potencial das mesmas na determinação de funcionalidade de proteínas.

Dada a complexidade desse problema são investigadas técnicas utilizadas nas áreas de Computação Inteligente e Bioinspirada (CARVALHO, 2003). As principais técnicas investigadas são: Árvores de Decisão (QUINLAN, 1993), Redes Neurais Artificiais (MONARD, 2003), algoritmos SVM (HAYKIN, 1999) e Algoritmos Genéticos (CARVALHO, 2003). Neste projeto são consideradas as variantes de hemoglobina, uma vez que tais técnicas requerem grandes quantidades de amostras e a hemoglobina possui um grande número de variantes determinadas.

Árvores de Decisão revelam a lógica utilizada para a classificação das variantes. Desta forma, o sucesso desta técnica em classificar variantes de proteínas pode também resultar em um melhor entendimento das mutações de aminoácidos na funcionalidade das proteínas.

O Capítulo 2 apresenta uma introdução sobre proteínas. O Capítulo 3 discute as principais características da proteína hemoglobina. O Capítulo 4 apresenta o conceito de aprendizado de máquina destacando a técnica de Árvores de Decisão uma vez que esta revela a lógica de classificação. O Capítulo 5 introduz as demais técnicas investigadas e a metodologia utilizada para o estudo de classificação de proteínas. O Capítulo 6 apresenta os testes e resultados. Por fim, o Capítulo 7 sintetiza os principais aspectos investigados neste trabalho, conclusões e outras considerações.

CAPÍTULO 2 - PROTEÍNAS

As proteínas estão no centro da ação nos processos biológicos. Praticamente todas as transformações moleculares que definem o metabolismo celular são medidas pela catálise proteica. As proteínas exercem também funções regulatórias, controlando as condições intracelulares e extracelulares e mandando informações para outros componentes das células. Uma lista completa de funções conhecidas das proteínas teria milhares de itens, incluindo proteínas que transportam outras moléculas e proteínas que geram forças mecânicas e eletroquímicas. Existem também outras milhares de proteínas cujas funções ainda não estão inteiramente elucidadas ou, em muitos casos, são mesmo completamente desconhecidas (VOET, 2000).

As proteínas são as moléculas orgânicas mais abundantes nas células e perfazem cinquenta por cento ou mais de seu peso seco. São encontradas em todas as partes de todas as células, sendo fundamentais sob todos os aspectos da estrutura e função celulares. Existem muitas espécies diferentes de proteínas, em geral, cada proteína possui uma função biológica específica (LEHNINGER, 1976). As principais funções de proteínas são (STRYER, 1988):

- 1) Catálise enzimática: quase todas as reações químicas em sistemas biológicos são catalisadas por macromoléculas específicas chamadas enzimas, que aumentam geralmente as velocidades de reações em pelo menos um milhão de vezes;
- 2) Transporte e armazenamento: muitas moléculas e íons pequenos são transportados por proteínas específicas;
- 3) Movimento coordenado: os principais componentes dos músculos são proteínas. A contração muscular é conseguida pelo movimento de deslizamento de dois tipos de filamentos protéicos;

- 4) Sustentação mecânica: a alta força de tensão da pele e do osso é devida à presença de uma proteína fibrosa, o colágeno.
- 5) Proteção imunitária: os anticorpos são proteínas específicas que reconhecem substâncias estranhas, como os vírus, bactérias e células de outros organismos e com essas se combinam;
- 6) Geração e transmissão de impulsos nervosos: a resposta de células nervosas a estímulos específicos é intermediada por proteínas receptoras;
- 7) Controle do crescimento e da diferenciação celular.

Além de todas essas funções, a maior parte da informação genética é expressa pelas proteínas.

Todas as proteínas contêm carbono, hidrogênio, nitrogênio e oxigênio e quase todas contêm enxofre. Algumas proteínas contêm elementos adicionais, particularmente fósforo, ferro, zinco e cobre. Os pesos moleculares das proteínas são muito altos em relação ao de outras moléculas. Por hidrólise ácida, as moléculas proteicas liberam uma série de compostos orgânicos simples de baixo peso molecular, chamados α -aminoácidos. Os aminoácidos comuns são conhecidos como α -aminoácidos porque possuem um grupo amino primário ($-\text{NH}_2$) e um grupo carboxílico ($-\text{COOH}$) como substituinte do mesmo átomo de carbono (carbono α). Essas moléculas diferem umas das outras quanto à estrutura de suas cadeias laterais ou grupamentos R. Vinte α -aminoácidos diferentes são comumente encontrados como blocos construtivos das proteínas.

2.2 A Composição das Proteínas

Os vinte α -aminoácidos comumente encontrados nas proteínas são também chamados de aminoácidos padrão. Com exceção da prolina, esses aminoácidos têm como denominador comum um grupamento carboxílico livre e um aminogrupamento livre não-substituído, no átomo do carbono α . Diferentes aminoácidos possuem as estruturas de suas cadeias laterais distintas, denominadas grupamentos R (ver Figura 2.1).

Figura 2.1 – Esquema representativo da estrutura de um aminoácido

Nas moléculas proteicas, os resíduos de aminoácidos ligam-se covalentemente, formando longos polímeros não-ramificados. Nesses polímeros, os aminoácidos estão unidos em um arranjo “cabeça-cauda”, mediante ligações amídicas substituídas, denominadas ligações peptídicas (ver Figura 2.2). Esta ligação é formada pela união de um grupo amino do carbono α de um aminoácido, com o grupo carboxila do carbono α de um segundo aminoácido. O carbono α é chamado assim por ser o primeiro carbono depois do grupo carboxila. Esta união é acompanhada pela eliminação de água entre as moléculas que participam da ligação (LEHNINGER, 1976).

Figura 2.2 – Esquema representativo de uma ligação peptídica.

Tais polímeros, denominados cadeias polipeptídicas, podem conter centenas de resíduos de aminoácidos. Polímeros compostos de dois, três, alguns (três a dez) e muitos aminoácidos são conhecidos, respectivamente, como dipeptídeos, tripeptídeos, oligopeptídeos e os polipeptídeos. Após incorporados a um peptídeo, os aminoácidos individuais (as unidades monoméricas) são chamados resíduos de aminoácidos (VOET, 2000).

Algumas proteínas contêm somente uma cadeia polipeptídica; outras contêm duas ou mais. As cadeias polipeptídicas das proteínas não são polímeros que ocorrem ao acaso, ou com comprimento indefinido; cada cadeia polipeptídica apresenta um peso molecular, composição química, ordem sequencial de seus aminoácidos e disposição espacial tridimensional definidos.

As variações no comprimento e na sequência de aminoácidos de polipeptídeos contribuem para a diversidade na forma e nas funções biológicas das proteínas (VOET, 2000).

De acordo com sua composição, as proteínas são divididas em duas classes principais:

- Proteínas simples são aquelas que, por hidrólise, liberam somente aminoácidos e nenhum outro produto orgânico e inorgânico;
- Proteínas conjugadas são aquelas que liberam não somente aminoácidos, mas também outros componentes orgânicos e inorgânicos.

As proteínas conjugadas podem ser classificadas de acordo com a natureza química de seus grupos prostéticos (a porção de uma proteína não formada por aminoácidos). Assim tem-se: nucleoproteínas, lipoproteínas, fosfoproteínas, metaloproteínas e as glicoproteínas (LEHNINGER, 1976). A Tabela abaixo ilustra alguns exemplos de proteínas conjugadas.

Tabela 2.1- Exemplo de proteínas conjugadas.

Proteína	Grupo prostético
Glicoproteínas	Glúcidos
Lipoproteínas	Lípidos: Ácidos graxos Colesterol Triglicéridos Fosfolípidos
Nucleoproteínas	Ácidos nucleicos
Metaloproteínas	Fe, Cu, Mn, Mo, Zn

2.3 A Conformação das Proteínas

Cada tipo de molécula proteica tem, em seu estado nativo, uma configuração tridimensional peculiar, designada conformação. Dependendo de sua conformação, as proteínas podem ser classificadas como fibrosas ou globulares (ver Figuras 2.3 e 2.4) (LEHNINGER, 1976). Essa dicotomia é anterior aos métodos para determinação da estrutura das proteínas em escala atômica e não prevê proteínas que contenham regiões fibrosas, rígidas ou alongadas e regiões globulares mais compactas ou altamente dobradas (VOET, 2000).

- Proteínas fibrosas são materiais insolúveis em água e soluções salinas diluídas e fisicamente resistentes. São constituídas de cadeias polipeptídicas dispostas paralelamente ao longo de um único eixo. Os exemplos são: o colágeno dos tendões e da matriz óssea, a α queratina dos cabelos e a elastina do tecido conjuntivo elástico.
- Proteínas globulares são formadas de cadeias polipeptídicas que se dobram, adquirindo formas esféricas ou globulares. A maioria dessas proteínas é solúvel em sistemas aquosos.

Figura 2.3 – Uma proteína fibrosa. Molécula de colágeno.

Figura 2.4 –Proteínas Globulares.

Algumas proteínas possuem uma conformação intermediária entre fibrosa e globular. Essas proteínas são constituídas por longas estruturas em bastão como as fibrosas e são solúveis em solução salina aquosa como as globulares. O fibrinogênio (ver Figura 2.5 a) e a miosina (ver Figura 2.5 b) são proteínas que possuem estas características.

a)

Figura 2.5 – (a) Proteína Fibrinogênio. (b) Proteína Miosina.

Fonte: www.unesp.br/propp/dir_proj/Saude/saude46b.htm, <http://cbe.ivic.ve/LBEfolleto.html>> Acesso em março de 2004

2.4 Níveis de Estruturas Proteicas

De forma semelhante às demais moléculas poliméricas, as proteínas podem ser descritas em termos de níveis de organização. As proteínas possuem quatro níveis de organização denominados estrutura: primária, secundária, terciária e quaternária (VOET, 2000).

A seguir são definidas cada uma dessas estruturas segundo (CONN E STUMPF, 1990).

Estrutura Primária: é a seqüência linear dos resíduos de aminoácidos que constituem sua cadeia polipeptídica. Evidentemente, nesse conceito está implícita a noção de ligação peptídica entre os aminoácidos, mas nenhuma outra ligação ou força é indicada (ver Figura 2.6).

Figura 2.6 – Esquema representativo da estrutura primária de proteína.

Fonte: ICB, Instituto de Ciências Biológicas. Universidade Federal de Minas Gerais. Disponível em <http://www.icb.ufmg.br/~lbcd/grupo1/pag1.html>. Acesso em agosto de 2003.

Estrutura Secundária: esse termo refere-se geralmente à estrutura que um polipeptídeo ou uma proteína pode possuir devido às interações das ligações de hidrogênio entre aminoácidos distantes um do outro na estrutura primária. Em 1951, Linus Pauling e Robert B. Corey propuseram duas estruturas polipeptídicas periódicas, chamadas de α -hélice e folha β pregueada. A α -hélice (ver Figura 2.7b) é uma estrutura em bastão. A cadeia principal polipeptídica densamente enrolada forma a parte interna do bastão, e as cadeias laterais estendem-se para fora em um arranjo helicoidal. A α -hélice é estabilizada por pontes de hidrogênio entre os grupamentos NH e CO da cadeia principal. A α -hélice é orientada para a direita, ou seja, é torcida na mesma direção em que os dedos da mão direita se fecham quando o polegar aponta na direção em que a hélice sobe. A α -hélice possui 3,6 resíduos por volta e um passo (à distância que a hélice aumenta ao longo do eixo por volta) de 5,4 Å. As α -hélices das proteínas possuem um comprimento médio de aproximadamente 12 resíduos, o que corresponde a mais de três voltas da hélice e a um comprimento de aproximadamente 18 Å.

Uma estabilização semelhante ocorre nas estruturas em folha β pregueada (ver Figura 2.7a). A folha β pregueada difere-se muito da α -hélice porque é uma folha, em vez de um bastão. Uma cadeia polipeptídica na folha β pregueada é quase completamente distendida, em vez de fortemente enrolada, como na α -hélice. Outra diferença é que a folha β pregueada é estabilizada por pontes de hidrogênio entre grupamentos NH e CO em cadeias polipeptídicas diferentes, ao passo que, na α -hélice, as pontes de hidrogênio entre

grupos NH e CO estão na mesma cadeia polipeptídica. Cadeias adjacentes na folha β pregueada podem correr na mesma direção (folha β paralela). Por exemplo, a fibroína da seda é constituída quase inteiramente de pilhas de folhas β antiparalelas. Tais regiões de folha β são um tema estrutural recorrente em muitas proteínas. São comuns unidades estruturais constituídas de duas a cinco fitas β paralelas ou antiparalelas.

Figura 2.7 – (a) Estrutura em folha β pregueada. (b) Estrutura em α -hélice.

Fonte: www.unesp.br/propp/dir_proj/Saude/saude46b.htm, <http://cbe.ivic.vc/LBEfolleto.html>> Acesso em maio de 2004

Estrutura Terciária: as cadeias polipeptídicas têm a tendência de enrolar-se ou dobrar-se, formado uma estrutura complexa, mais ou menos rígida. Normalmente, o dobramento ocorre devido a interações entre resíduos de aminoácidos relativamente distantes na sequência peptídica. A estabilização desta estrutura é atribuída as diferentes reatividades associadas com os grupos R dos resíduos de aminoácidos. A estrutura terciária de uma proteína descreve o dobramento dos elementos estruturais secundários e especifica as posições de cada átomo na proteína, incluindo as cadeias laterais. As estruturas conhecidas de proteínas foram obtidas por cristalografia de raios-X ou por estudos de ressonância magnética nuclear (RMN) (VOET, 2000) (ver Figura 2.8).

Figura 2.8 – Esquema representativo da estrutura terciária da mioglobina.

Fonte: CAMPBELL, 2000.

Estrutura Quaternária: essa estrutura corresponde ao resultante de interações entre unidades polipeptídicas isoladas de uma proteína (estruturas terciárias) contendo mais de uma subunidade, formando uma estrutura super proteica. A maioria das proteínas, particularmente as com massas moleculares maiores 100KD, é constituída por mais de uma cadeia polipeptídica. As subunidades polipeptídicas associam-se com uma geometria específica. O arranjo espacial dessas subunidades é conhecido como estrutura quaternária da proteína (VOET, 2000) (ver Figura 2.9).

Figura 2.9 – Esquema representativo da estrutura quaternária da hemoglobina.

Fonte: ICB - Instituto de Ciências Biológicas. Universidade Federal de Minas Gerais. Disponível em <http://www.icb.ufmg.br/~lbcd/grupo1/pag1.html>. Acesso em agosto de 2003

2.5 Determinação da Estrutura das Proteínas

A cristalografia de raios-X é um dos métodos mais poderosos para o estudo da estrutura macromolecular. De acordo com os princípios ópticos, a incerteza na localização de um objeto é aproximadamente igual ao comprimento de onda da radiação utilizada para observá-lo. Os raios-X podem formar diretamente a imagem de uma molécula, pois os comprimentos de onda dos raios-X são comparáveis às distâncias das ligações covalentes (aproximadamente 1,5 Å; as moléculas não podem ser vistas individualmente em um microscópio eletrônico, pois o menor comprimento de onda da luz visível é de 4.000 Å).

Quando um cristal da molécula a ser visualizado é exposto a um raio colimado (paralelo) de raios-X, os átomos na molécula desviam os raios, os quais podem se cancelar ou se combinar aumentando a intensidade de raios em certas direções em um processo denominado difração. O padrão de difração resultante é captado em filme fotográfico ou em um contador de radiação. As intensidades de difração máximas (intensidade dos pontos escuros no filme) são utilizadas para construir matematicamente uma imagem tridimensional da estrutura do cristal (VOET, 2000).

2.6 Aminoácidos

Os cientistas, no início do século XIX, voltaram pela primeira vez sua atenção para a nutrição. Eles descobriram que produtos naturais contendo nitrogênio eram essenciais para a sobrevivência dos animais. Agora entendemos que os aminoácidos são essenciais para a vida e são unidades estruturais que compõem as proteínas. A função central dos aminoácidos em bioquímica talvez não seja surpreendente: vários aminoácidos estão entre os compostos orgânicos que, acredita-se, surgiram nos primórdios da história da terra. Os aminoácidos, como moléculas antigas, foram captados pela evolução para uma variedade de propósitos nos sistemas vivos (VOET, 2000).

Os aminoácidos são o alfabeto da estrutura proteica e determinam muitas das importantes propriedades das proteínas. O primeiro aminoácido isolado a partir de um hidrolisado protéico foi a glicina, em 1820, obtida da gelatina, por H. Braconnot (LEHNINGER, 1976).

O aminoácido descoberto mais recentemente foi dentre os vinte comumente encontrados nas proteínas, a treonina e foi isolado a partir da hidrólise da fibrina por W. C. Rose, em 1.935. Além desses vinte aminoácidos, vários aminoácidos adicionais tem sido encontrados desempenhando outras funções nas células.

2.6.1 Os Aminoácidos Comuns em Proteínas

Várias maneiras de se classificar os aminoácidos, tendo como base seus grupamentos R, foram propostas. A forma mais útil de classificar os vinte aminoácidos-padrão tem sido pela polaridade de suas cadeias laterais. De acordo com o esquema mais comum de classificação, há quatro tipos principais de aminoácidos:

1) Aqueles com grupos R não polares ou hidrofóbicos.

Essa família inclui cinco aminoácidos com grupos R de cadeia alifática hidrocarbonatada: Alanina, Leucina, Isoleucina, Valina e Prolina. Dois aminoácidos com anéis aromáticos: Fenilalanina e Triptofano. E um contendo Enxofre: Metionina.

O membro menos hidrofóbico dessa classe é a Alanina.

2) Aminoácidos com grupos R polares ou hidrofílicos.

Esses aminoácidos são relativamente mais solúveis em água do que aqueles com grupos R não-polares. Seus grupos R contêm grupos funcionais neutros (não carregados) polares que podem formar ligações de hidrogênio com a água. A polaridade da Serina, Treonina e da Tirosina é resultante de seus grupos hidroxílicos. A polaridade da Asparagina e da Glutamina resulta de seus grupos amídicos. A polaridade da Cisteína resulta de seu grupamento sulfidrílico (SH). A Glicina é algumas vezes classificada como um aminoácido não polar, porém seu grupamento R, um único átomo de hidrogênio, é muito pequeno para influenciar o alto nível de polaridade dos grupos α -amínico e α -carboxílico.

3) Aminoácidos com grupos R carregados positivamente ou básicos.

Os aminoácidos básicos, em que os grupos R apresentam uma carga positiva efetiva em pH 7,0, tem todos seis carbonos em seu esqueleto.

- A Lisina que apresenta um aminogruppo carregado positivamente;
- A Arginina possui o grupo guanidínico carregado positivamente;
- A Histidina contém a função imidazólica fracamente básica. A Histidina, em pH 6,0 ioniza na faixa de pH fisiológico.

4) Aminoácidos com grupos R carregados negativamente ou ácidos.

Os dois membros dessa classe:

- Ácido Aspártico e o Ácido Glutâmico, cada um com um grupo carboxílico, além do α -carboxílico, que é inteiramente ionizado e, portanto, carregado negativamente em pH 6,0 –7,0 (LEHNINGER, 1976).

A inclusão de um aminoácido em um ou em outro grupo não reflete apenas as propriedades do aminoácido isolado, mas também seu comportamento quando na composição de um polipeptídeo. As estruturas da maioria dos polipeptídeos dependem da tendência das cadeias laterais polares e iônicas solvatarem-se³ e das cadeias laterais apolares associarem-se entre si, em vez de associarem-se com a água. Essa propriedade dos polipeptídeos é o efeito hidrofóbico (VOET, 2000).

³ Solvatar: fenômeno de fixação de moléculas do solvente por um íon ou por uma partícula em solução

2.6.2 Localização das Cadeias Laterais

As cadeias laterais de aminoácidos nas proteínas globulares são distribuídas espacialmente de acordo com suas polaridades:

- 1) Os resíduos não-polares Valina (Val), Leucina (Leu), Isoleucina (Ile), Metionina (Met) e Fenilalanina (Phe) ocorrem principalmente no interior das proteínas, fora do contato com o solvente aquoso. Os efeitos hidrofóbicos que promovem essa distribuição são os principais responsáveis pela estrutura tridimensional das proteínas nativas;
- 2) Os resíduos polares carregados Arginina (Arg), Histidina (His), Lisina (Lys), Ácido aspártico (Asp) e Ácido glutâmico (Glu) estão em geral localizados na superfície de uma proteína, em contato com o solvente aquoso. Isso ocorre porque a imersão de um íon no interior de uma proteína, que é praticamente anidro, é energeticamente desfavorável;
- 3) Os grupos polares não-carregados Serina (Ser), Treonina (Thr), Asparagina (Asn), Glutamina (Gln) e Tirosina (Tyr) estão com frequência na superfície da proteína, podendo também ocorrer no interior da molécula. Quando estiverem imersos na proteína, esses resíduos estarão quase sempre ligados por pontes de hidrogênio a outros grupos.

2.6.3 Ligações entre Aminoácidos

Existem vários tipos de ligações entre os aminoácidos que são importantes na manutenção das suas estruturas conformacionais (DOSE, 1982). Essas ligações são descritas a seguir:

1) Pontes Dissulfeto

É a ligação covalente que contribui especificamente para manutenção de uma determinada conformação na cadeia, principalmente entre α -queratinas e na maioria das proteínas globulares. Essa ligação ocorre pela oxidação de dois resíduos de cisteína, que podem pertencer a cadeias diferentes ou, ainda, a mesma cadeia.

2) Pontes de Hidrogênio

Ocorrem quando duas ligações peptídicas, mais exatamente o grupo amino (doador de hidrogênio) de uma e o grupo carboxila (receptor de hidrogênio) de outra, são aproximados a uma distância de 0,28 nm. As cadeias laterais dos aminoácidos participam de pontes de hidrogênio. Essas ligações são geralmente de menor importância para a estabilização da estrutura espacial de uma cadeia peptídica.

3) Interações Hidrofóbicas

As ligações por interações hidrofóbicas entre determinadas cadeias laterais de aminoácidos (especialmente a Valina, Leucina, Isoleucina, fenilalanina e triptofano) são de grande importância para a estabilização da conformação das cadeias em proteínas. As ligações hidrofóbicas só podem formar-se na presença de água. A sua formação explica-se pelo fato de que os resíduos hidrofóbicos, em contato com a água ficam recobertos de uma camada de moléculas muito organizadas.

4) Ligações Eletrostáticas

Interações eletrostáticas (ligação íons-íons) ocorrem entre carboxilas dissociadas e bases nitrogenadas protonizadas do grupo guanidina da arginina. Essas ligações são relativamente fracas.

5) Ligação Peptídica

Os polipeptídeos são polímeros lineares; isto é, cada resíduo de aminoácido participa de duas ligações peptídicas e liga-se a seus vizinhos de forma cabeça-cauda, em vez de formar cadeias ramificadas. Os resíduos das extremidades da cadeia fazem apenas uma ligação peptídica cada um. Os resíduos com um grupo amino livre (por convenção é o resíduo da extrema esquerda) são chamados de aminoterminal ou N-terminal. O resíduo com um grupo carboxílico livre (o da direita) é chamado carboxi-terminal ou C-terminal (VOET, 2000). A Figura 2.10 ilustra as ligações entre os aminoácidos.

Figura 2.10 – Forças que estabilizam a estrutura terciária das proteínas.

Fonte: CAMPBELL, 2000.

2.6.4 Aminoácidos Raros das Proteínas

Os vinte aminoácidos padrão não são os únicos aminoácidos que ocorrem em sistemas biológicos. Os resíduos de aminoácidos incomuns são, em geral, importantes constituintes de proteínas e de peptídeos biologicamente ativos. Além disso, muitos aminoácidos não são apenas constituintes de polipeptídeos, mas desempenham, de modo independente, uma variedade de funções biológicas (VOET, 2000). Todos esses aminoácidos, listados a seguir são derivados de algum aminoácido padrão: 4 Hidroxiprolina, Hidroxilisina, Desmosina, Isodesmosina, Metilisina e Metilistidina (LEHNINGER, 1976).

2.6.5 Aminoácidos não Protéicos

Além dos vinte aminoácidos padrão comuns e de vários aminoácidos raros das proteínas, mais de cento e cinquenta outros aminoácidos são conhecidos como ocorrendo

biologicamente em forma livre ou combinada, porém nunca em proteínas. A seguir mostra-se alguns exemplos de aminoácidos não protéicos: Homocisteína, Homoserina, Citrulina, Ornitina, Ácido γ -aminobutírico, Ácido D-glutâmico, D-alanina e outros (LEHNINGER, 1976).

2.6.6 Reação Química dos Aminoácidos

As reações orgânicas características dos aminoácidos são aquelas de seus grupamentos funcionais, isto é, os grupos carboxílicos, os α aminogrupos e os grupos funcionais presentes nas diversas cadeias laterais. O conhecimento dessas reações é útil em vários aspectos importantes da química das proteínas:

- 1) Identificação e análise dos aminoácidos nos hidrolisados protéicos;
- 2) Identificação da sequência de aminoácidos nas moléculas proteicas;
- 3) Identificação dos resíduos específicos de aminoácidos das proteínas nativas que são necessários para sua atividade biológica;
- 4) Modificações químicas dos resíduos de aminoácidos nas moléculas proteicas capazes de produzir modificações em suas atividades biológicas ou em outras propriedades;
- 5) A síntese química dos polipeptídeos.

2.6.7 Nomenclatura dos Aminoácidos

As abreviaturas em três letras para os 20 aminoácidos listados no Apêndice C são bastante utilizadas na literatura bioquímica. A maior parte dessas abreviaturas é composta pelas três primeiras letras do nome do aminoácido correspondente e é pronunciada como escrita.

Os símbolos de uma só letra para os aminoácidos também são listados na tabela do mesmo apêndice. Esse código mais compacto é geralmente usado quando se compara seqüências de aminoácidos de várias proteínas similares. Note que o símbolo de uma letra é em geral a primeira letra do nome do aminoácido. Entretanto, para conjuntos de resíduos que têm a mesma letra inicial, isso é válido apenas para o resíduo mais abundante.

2.7 Conformação Tridimensional

Em cada tipo de proteína, a cadeia polipeptídica é enovelada em uma conformação tridimensional específica, que é indispensável para sua função biológica específica ou atividade.

Sabe-se que a cadeia polipeptídica de uma proteína possui somente uma conformação ou algumas em condições biológicas normais de temperatura e pH. Essa conformação nativa, que confere a atividade biológica é suficientemente estável, de modo que a proteína pode ser isolada e conservar sua condição nativa (STRYER, 1988).

Esse trabalho investiga um tipo específico de proteína, a hemoglobina, pois esta molécula proteica tem sido muito estudada na literatura, assim existem dados suficientes e adequados para desenvolver a pesquisa. O Capítulo seguinte apresenta as principais características da hemoglobina.

CAPÍTULO 3 - HEMOGLOBINA

A hemoglobina, a proteína intracelular que dá cor aos eritrócitos⁵, é uma das proteínas melhor caracterizadas e foi uma das primeiras a serem associadas com uma função fisiologicamente específica (transporte de oxigênio). Contudo, a hemoglobina não é somente um simples depósito de oxigênio, pois compõe um sistema sofisticado de transporte que, sob uma ampla variedade de circunstâncias, fornece a quantidade adequada de oxigênio aos tecidos. Os animais demasiadamente grandes (com mais de um milímetro de espessura), não são capazes de transportar quantidades suficientes de oxigênio para seus tecidos por difusão simples e possuem sistemas circulatórios contendo hemoglobina ou uma proteína com funções similares para realizar esta tarefa (VOET, 2000).

Os vertebrados desenvolveram dois mecanismos principais para suprir suas células com um fluxo contínuo e adequado de oxigênio. O primeiro é um sistema circulatório que distribui ativamente oxigênio às células. O segundo é a utilização de moléculas transportadoras de oxigênio para vencer as limitações impostas pela baixa solubilidade do oxigênio em água. Os carreadores de oxigênio nos vertebrados são proteínas: hemoglobina e mioglobina. As hemoglobinas, contidas nas hemácias, servem de carreador de oxigênio no sangue e também exercem um papel vital no transporte de dióxido de carbono (CO_2) e íons de hidrogênio (OH^- e H_2O).

Normalmente, as hemácias têm a forma de um disco bicôncavo. Essas células são muito flexíveis e passam facilmente por pequenos vasos sanguíneos. Isso é

⁵ As hemácias, eritrócitos ou glóbulos vermelhos são as células encontradas em maior quantidade no sangue e que lhe conferem a cor. O constituinte mais importante da hemácia é a hemoglobina, que transporta o oxigênio mediante ligação química com as suas moléculas. Dentro de cada hemácia existem milhões de moléculas de hemoglobina. Existem, em média, 4,5 milhões de hemácias em cada mililitro de sangue no homem e cerca de 4 milhões na mulher. Quando a quantidade de hemácias no sangue está diminuída, o paciente tem anemia; se estiver aumentada, o fenômeno se chama poliglobulia ou policitemia. A sobrevivência média das hemácias no sangue circulante é de 100 a 120 dias; a medula óssea produz hemácias continuamente, para a sua renovação no sangue circulante.

importante porque as hemácias percorrem de grandes vasos até pequenos vasos entregando o oxigênio para todos os órgãos do corpo. Desta forma, as hemoglobinas são perfeitamente adaptadas para a entrega do oxigênio para os vasos menores e para as áreas mais remotas do corpo. Se as hemácias não fossem flexíveis, não seriam capazes de se apertarem para passar nesses vasos de pequeno calibre e bloqueariam a passagem do sangue para certas partes do corpo. (ver Figura 3.1).

Figura 3.1 – Figura representativa de uma célula sanguínea.
Fonte: CAMPBELL, 2000.

3.2 Estrutura da Hemoglobina

A hemoglobina é a proteína alostérica (ver Seção 3.4.4) mais bem conhecida e, por isso, é importante o exame de sua estrutura e função normal (STRYER, 1988). Esta proteína contém duas cadeias α , de 141 resíduos denominadas $\alpha 1$ e $\alpha 2$ e duas cadeias β de 146 resíduos denominadas $\beta 1$ e $\beta 2$. A cadeia (subunidade) possui um resíduo heme (ver Seção 3.3) em ligação não covalente. As cadeias α contêm 7 segmentos helicoidais e as cadeias β 8 segmentos. As 4 subunidades da hemoglobina estabelecem contatos interatômicos umas com as outras mantendo a integridade da molécula. Esses contatos ocorrem por meio de interações do tipo ponte salina, ligação de hidrogênio e interações do tipo Van der Waals entre resíduos situados na interface. Alguns desses contatos são particularmente importantes para a molécula, em virtude da cooperatividade (VIANA, 1994).

As duas cadeias α e as duas cadeias β possuem resíduos idênticos em várias posições de suas seqüências; isto é, as duas cadeias possuem o que se chama homologia

seqüencial. Cada cadeia tem uma conformação irregularmente enovelada. As subunidades são estruturalmente muito parecidas com a mioglobina, sendo formadas predominantemente por oito α hélices (rotuladas com as oito primeiras letras do alfabeto em maiúsculo) (DELBONI, 1991). A molécula, em sua forma oxigenada, apresenta uma estrutura esferoidal compacta, com as dimensões de 6,4 por 5,5 por 5,0 nm ou 64x 55x 50 Å.

A estrutura quaternária da hemoglobina é mantida por ligações não-covalentes, que são muito mais numerosas entre as subunidades diferentes (α/β) do que entre subunidades iguais (α/α e β/β). O resultado desta associação desigual é uma molécula tetramérica, composta por dois dímeros, denominados $\alpha_1\beta_1$ e $\alpha_2\beta_2$, dispostos simetricamente ao redor de um eixo central. Os dois dímeros estabelecem contatos por meio da interface entre a subunidade α de um dímero e a subunidade β adjacente do outro dímero: interface $\alpha_1\beta_2$ e sua equivalente simétrica $\alpha_2\beta_1$ (MARZZOCO, 1999).

Os pontos de contato entre as cadeias α e β são basicamente hidrofóbicos e podem ser de dois tipos. O primeiro deles ocorre entre as subunidades $\alpha_1\beta_1$ (ou $\alpha_2\beta_2$) e denominados de contato de empacotamento (ver Figura 3.2), por representarem a interface das subunidades que não muda quando a molécula de hemoglobina passa da configuração deoxi para oxi (ver Seção 3.4).

A segunda forma de contato, representada pelas cadeias $\alpha_1\beta_2$ ou ($\alpha_2\beta_1$), por sua vez, são chamados de contato de deslizamento (ver Figura 3.3) uma vez que sofrem mudanças quando o oxigênio liga-se a molécula (ver Figura 3.8). As interfaces ($\alpha_1\alpha_2$ ou $\beta_1\beta_2$) são menos extensas e mais polares. A interface $\alpha_1\beta_2$ (ou $\alpha_2\beta_1$) entre as subunidades está conectada indiretamente ao grupo heme (ver Seção 3.3) e, portanto, é natural que alterações estruturais nestas regiões sejam transmitidas aos grupos hemes.

Figura 3.2 - Região de contato entre $\alpha_2\beta_2$ envolvendo as hélices B, o dobramento GH e as hélices G.

Fonte: DICKERSON, 1983.

Figura 3.3 - Interface de contato entre as subunidades $\alpha_1\beta_2$ entre os dobramentos FG e as hélices C.

Fonte: DICKERSON, 1983.

Os adultos também apresentam uma hemoglobina em menor quantidade, aproximadamente dois por cento da hemoglobina total no organismo, chamada de hemoglobina A₂, que contém cadeias δ em vez das cadeias β da hemoglobina. Os fetos têm hemoglobinas diferentes. Pouco depois da concepção, os fetos sintetizam cadeias ζ ,

que são semelhantes às cadeias α e cadeias ϵ , semelhantes às β . No transcorrer da vida fetal, as ζ são substituídas por α e a ϵ , pela γ , que é substituída pela β (STRYER, 1988).

As seqüências de aminoácidos das cadeias da hemoglobina de muitas espécies foram comparadas. Ainda que somente nove resíduos em cada cadeia sejam absolutamente invariáveis, as substituições de aminoácidos em muitas outras posições sugerem que as subunidades de cadeias polipeptídicas das hemoglobinas possuam a mesma estrutura terciária (ver Figura 3.4).

Figura 3.4 – Esquema representativo da molécula de mioglobina.

Fonte: CAMPBELL, 2000.

3.3 Grupo Heme

O grupo heme consiste de um íon metálico, o Fe (II), e uma porção orgânica, a protoporfirina IX (CAMPBELL, 2000). A porção porfirínica consiste de quatro anéis, estes anéis estão ligados por pontes de grupos metinos ($-\text{CH}=\text{}$), formando uma estrutura quadrada plana (ver Figura 3.5). De especial interesse é a localização dos quatro grupamentos de heme. Além do oxigênio outros elementos podem se ligar a este átomo de Fe concedendo à molécula de hemoglobina uma variada denominação. Por exemplo, a molécula é chamada de desoxihemoglobina se o átomo de ferro está livre; hidroxihemoglobina se o átomo de Fe ligar-se a um grupo OH^- e assim por diante

(VIANA, 1994). O grupo heme também dá a mioglobina e a hemoglobina sua cor característica (LEHNINGER, 1976) e (STRYER, 1988).

Figura 3.5 – Esquema representativo do grupo heme.

3.4 Mecanismo de Cooperatividade

Em uma molécula de hemoglobina ligam-se quatro átomos de oxigênio, sendo que a quarta ligação ocorre cerca de trezentas vezes mais facilmente que a primeira. Este fenômeno é chamado de cooperatividade, por resultar da comunicação entre as subunidades como descrito na seção acima.

Estudos de difração de raios X mostraram que a oxihemoglobina (ver Figura 3.8) e a desoxi-hemoglobina (ver Figura 3.7) diferem muito em estrutura quaternária. A oxigenação envolve mudanças na conformação da proteína, conferindo dois estados conformacionais diferentes e designados como tenso e relaxado.

A molécula da desoxi-hemoglobina é presa por ligações salinas entre cadeias diferentes, os aminoácidos carboxi-terminais das quatro cadeias estão ancorados e, assim como, suas cadeias laterais dos aminoácidos C-terminais, participam de ligações salinas. A ligação do oxigênio à proteína desloca o equilíbrio para a direção da estrutura de maior afinidade, o estado relaxado ou oxigenado, resultando em um aumento da afinidade pelo oxigênio à medida que a reação ocorre.

A Figura 3.6 ilustra a mudança na estrutura quaternária que acompanha a ligação de oxigênio à hemoglobina. A ligação provoca uma ligeira mudança da posição do

dímero $\alpha_1\beta_1$ em relação ao dímero $\alpha_2\beta_2$ (ou vice-versa). Na Figura da estrutura ligada os dímeros $\alpha_1\beta_1$ estão superpostos.

Figura 3.6 – Representação esquemática dos estado tenso (não ligada) e relaxado (ligada).

Figura 3.7 – Desoxi-hemoglobina.

Fonte: CAMPBELL, 2000.

Figura 3.8 – Oxihemoglobina.

Fonte: CAMPBELL, 2000.

A ligação de oxigênio à hemoglobina estimula a ligação de mais oxigênio à mesma molécula, em outras palavras, o O_2 liga-se cooperativamente à hemoglobina. Essa ligação cooperativa do oxigênio torna a hemoglobina mais eficiente para o transporte do oxigênio, capacitando a molécula de hemoglobina para liberar 1,83 vezes mais oxigênio em condições fisiológicas do que se seus centros fossem independentes, como na mioglobina.

A saturação (Y) da hemoglobina (fração de ocupação dos centros de ligação a oxigênio) muda rapidamente com mudanças na pressão parcial de O_2 (pO_2). O gráfico de Y contra pO_2 é chamado de curva de dissociação de oxigênio (ver Figura 3.9). Essa curva difere da mioglobina para a hemoglobina; para qualquer pO_2 dado, Y é maior para mioglobina, assim conclui-se que a afinidade da mioglobina é maior que da hemoglobina (VOET, 2000).

Figura 3.9 – Curva de saturação da mioglobina e da hemoglobina.

Outra característica importante é o efeito Bohr, que consiste no aumento ou diminuição da afinidade da molécula de hemoglobina pelo oxigênio em virtude da concentração de CO₂ e H⁺.

3.5 Mutações

Algumas vezes um gene que especifica uma determinada proteína sofre uma alteração química, como resultado de algum agente físico, por exemplo, radiação, ou ação de determinados agentes químicos, de modo que uma das três bases de uma trinca de codificação para um determinado resíduo de aminoácido é alterada quimicamente ou perdida; algumas vezes um nucleotídeo extra é inserido. Como resultado, a continuidade normal da sequência de trincas de codificação do DNA é alterada e produz uma alteração correspondente na sequência de nucleotídeos do RNAm que, por sua vez, irá codificar então uma cadeia polipeptídica alterada. Em uma cadeia polipeptídica anormal, um ou muitos resíduos de aminoácidos de sua sequência específica podem ser substituídos por outros. Como consequência, a proteína correspondente pode ser deficiente quanto à sua função biológica.

3.6 Hemoglobinas Mutantes ou Variantes

A descoberta de hemoglobinas mutantes revelou que a mudança de um só aminoácido em uma proteína pode causar uma doença. O conceito de doença molecular, atualmente parte integrante da medicina, advém de estudos da hemoglobina anormal que causa a anemia falciforme (STRYER, 1988). A hemoglobina tem sido também uma rica fonte de entendimento das bases moleculares da evolução. As hemoglobinas anormais são de diversos tipos:

- Exterior alterado;
- Centro ativo alterado;
- Estrutura terciária alterada;
- Estrutura quaternária alterada.

A Tabela 3.1 apresenta exemplos de hemoglobinas variantes. A coluna mutação apresenta a subunidade (cadeia) da hemoglobina com a respectiva posição e aminoácidos trocados pela mutação. A nomenclatura dos aminoácidos é apresentada no Apêndice B.

Tabela 3.1 – Algumas Hemoglobinas Variantes

Nome	Mutação	Efeito
Volga	Beta 27 Ala → Asp	Anemia crônica
Bristol	Beta 67 Val → Asp	Enfraquece a ligação do heme
Knossos	Beta 27 Ala → Ser	Anemia moderada
Savannah	Beta 24 Gly → Val	Rompe a interface entre a hélice B-E
Philly	Beta 35 Tyr → Phe	Rompe as pontes de hidrogênio na interface α_1 - β_1
Miyano	Alfa 41 Thr → Ser	Eritrocitose
Torino	Alfa 43 Phe → Val	Anemia hemolítica
Saint Louis	Beta 28Leu → Gln	Cianose e anemia crônica
Yokohama	Beta 31 Leu → Pro	Anemia hemolítica severa

Fonte: <http://globin.cse.psu.edu/hbvar>

3.7 Alterações Mutacionais na Seqüência de Aminoácidos de uma Espécie

Antes do advento das técnicas do DNA recombinante ⁶, as hemoglobinas mutantes proporcionaram uma oportunidade única para o estudo das relações entre a estrutura e a função das proteínas. Isso ocorreu porque a hemoglobina foi, por muitos anos, a única proteína com estrutura conhecida a ter um grande número de variantes naturais bem caracterizadas. O estudo de indivíduos com deficiências fisiológicas, juntamente com a análise eletroforética rotineira de amostra de sangue humano, levou á descoberta de aproximadamente 500 hemoglobinas variantes. Cerca de 95% delas são resultado de substituições de um único aminoácido na cadeia polipeptídica da globina. De fato cerca de 5% da população mundial é portadora de uma variante congênita de hemoglobina. Nem todas as hemoglobinas variantes provocam sintomas clínicos, mas algumas moléculas anormais causam doenças debilitantes (nunca foram encontradas hemoglobinas variantes letais). As mutações que desestabilizam as estruturas terciária ou quaternária alteram a afinidade da hemoglobina pelo O₂ e reduzem sua cooperatividade.

Determinadas mutações no sítio de ligações do O₂ favorecem a oxidação do Fe (II) para Fe (III)⁷. As mutações que aumentam a afinidade da hemoglobina pelo oxigênio levam a um aumento do número de eritrócitos (VOET, 2000).

Estudos químicos tem demonstrado que a hemoglobina falciforme difere da hemoglobina normal em apenas um único resíduo de aminoácido. Em 1.945, Linus Pauling formulou a hipótese de que a anemia falciforme seria o resultado de uma hemoglobina mutante, mas o efeito genético só foi identificado em 1.956, quando Vernon Igran mostrou que a hemoglobina S (hemoglobina falcêmica) continha Valina na posição seis da cadeia β , em vez de Ácido Glutâmico. Essa foi a primeira

⁶ *A Criação do DNA Recombinante* envolve a união de um fragmento de DNA a uma molécula maior, utilizando-se uma endonuclease de restrição e a DNA ligase. A clivagem do DNA com a mesma enzima de restrição cria extremidades complementares adesivas que são unidas pela ação da DNA ligase. Desta forma, um fragmento de DNA pode ser inserido em uma molécula maior, que passa a ser recombinante. Assim, um determinado gene do genoma humano pode ser inserido no genoma de uma bactéria e ser transcrito várias vezes.

⁷ O átomo de ferro pode estar no estado de oxidação ferroso (+2) ou férrico (+3), cujas formas correspondentes de hemoglobina são chamadas respectivamente ferro-hemoglobina e ferri-hemoglobina ou meta-hemoglobina, sendo que somente a ferro-hemoglobina é capaz de se ligar ao oxigênio.

demonstração da existência de uma doença congênita originada pela troca de um aminoácido específico em uma proteína (VOET, 2000).

As duas valinas na posição um e seis formam uma associação hidrofóbica, levando a molécula de hemoglobina falciforme a assumir a conformação que altera a forma das hemácias, tornando-as falciforme. A hemoglobina alterada é chamada de HbS por adquirir em sua forma patológica uma forma de “foice” (*Sickle*) e, por isso, a terminologia S. A anemia falciforme é assim uma doença molecular de origem genética; a substituição do aminoácido é o resultado de uma mutação na molécula do DNA que codifica para a síntese da cadeia β da hemoglobina.

A substituição específica de aminoácidos em uma proteína mutante pode ser determinada de maneira muito simples pela aplicação da técnica do mapa peptídico⁸. Os nomes dessas formas anormais são muitas vezes derivadas das localizações geográficas de suas descobertas (LEHNINGER, 1976).

Figura 3.10 – Representação de uma célula sanguínea com a anemia falciforme

Fonte: CAMPBELL, 2000.

3.8 Proteínas Alostéricas

A cooperatividade da ligação do oxigênio à hemoglobina é um modelo clássico utilizado para descrever o comportamento de outras proteínas multiméricas (ou multicadeia) que ligam moléculas pequenas. A união de um ligante a um sítio aumenta a afinidade de outros sítios de ligação na mesma proteína, como é o caso da ligação do O_2

⁸ Para saber qual a alteração causada pelas mutações em cada uma das proteínas mutantes, utiliza-se a técnica de impressão digital (*fingerprint*) ou mapa peptídico.

à hemoglobina. Em outros casos, o ligante reduz a afinidade de outros sítios de ligação, como por exemplo, quando o BPG⁹ reduz a afinidade da hemoglobina pelo O₂. Um sítio de ligação é uma região localizada na superfície da molécula proteica onde ocorrerá a união de um ligante a molécula, por exemplo, no caso da proteína hemoglobina a ligação do oxigênio a esta molécula. Todos esses efeitos são o resultado de interações alostéricas (do grego, *allos* significa outro e *stereos* quer dizer sólido ou espaço). Os efeitos alostéricos, nos quais a ligação de um ligante a um sítio afeta a ligação de outro ligante a outro sítio, requerem interações entre subunidades de proteínas oligoméricas¹⁰ (VOET, 2000).

No Capítulo seguinte apresenta-se uma introdução a respeito de técnicas de aprendizado de máquina, destacando a técnica de Árvores de Decisão, uma vez que esta pode revelar a lógica utilizada para classificar as moléculas de hemoglobina mutantes.

⁹ O BPG (2,3 - bifosfoglicerato) diminui a afinidade da hemoglobina pelo oxigênio, o que é essencial para que a hemoglobina descarregue o oxigênio nos capilares dos tecidos. O BPG liga-se à desoxi-hemoglobina mas não na forma oxigenada, ligando-se na cavidade central da molécula. Na oxigenação o BPG é expulso porque a cavidade central fica pequena demais.

¹⁰ Proteínas Oligoméricas - Formadas por mais de uma cadeia polipeptídica São as proteínas de estrutura e função mais complexas.

CAPÍTULO 4 - APRENDIZADO DE MÁQUINA

O emprego de métodos computacionais na biologia iniciou-se na década de 1980, quando biólogos experimentais, em conjunto com cientistas da computação, físicos e matemáticos, começaram a desenvolver modelos computacionais para sistemas biológicos. Dentre as diversas áreas da biologia, aquela em que a aplicação de técnicas computacionais tem se mostrado mais promissora é a Biologia Molecular. A computação pode ser aplicada na resolução de problemas como a comparação de seqüências (DNA, RNA e proteínas), montagem de fragmentos, reconhecimento de genes, identificação e análise da expressão de genes e determinação da estrutura das proteínas (SOUTO, 2003).

O Aprendizado de Máquina (AM) é uma área de pesquisa cujo objetivo é o desenvolvimento de sistemas computacionais capazes de aprenderem e adquirirem conhecimento de forma automática (MONARD E BARANAUSKAS, 2003). Um algoritmo de aprendizado é um programa computacional capaz de tomar decisões baseadas em soluções de problemas anteriores. Os sistemas de aprendizado possuem características que permitem sua classificação de acordo com a linguagem de descrição, modo, paradigma e forma de aprendizado (MONARD E BARANAUSKAS, 2003).

Técnicas de AM são cada vez mais empregadas para tratar problemas em biologia molecular, por sua capacidade de aprender automaticamente a partir de grandes quantidades de dados e produzir hipóteses úteis.

Simon (1983) definiu a aprendizagem como mudanças adaptáveis no sistema, no sentido de que permitem que o sistema, da próxima vez, faça a mesma tarefa ou tarefas tiradas do mesmo grupo com mais eficiência e eficácia (RICH, 1993).

As pesquisas em AM investigam como construir programas de computador que melhoram seu desempenho em alguma tarefa por meio de experiência. Aprender, nesse contexto pode ser definido da seguinte forma: um programa computacional aprende a partir da experiência E, em relação a uma classe de tarefas T, com medida do

desempenho P , se seu desempenho nas tarefas T , medida por P , melhora com a experiência E (SOUTO, 2003).

Dentre as técnicas de AM existentes, tais como redes neurais (rede MLP- *Multi Layer Perceptron*), Algoritmos Genéticos e algoritmo SVM (*Support Vector Machines*), este trabalho irá explorar com maior profundidade a técnica denominada Árvores de Decisão (ADs) uma vez que esta pode revelar as regras utilizadas para classificação das moléculas mutantes ou variantes de hemoglobina, conforme a patologia associada.

É importante observar que a classificação é um componente importante de muitas tarefas de solução de problemas. Em sua forma mais simples, é considerada como uma tarefa direta de reconhecimento. A classificação é o processo de atribuir a uma determinada informação recebida, o nome de uma classe a qual a informação pertence (RICH, 1993).

4.2 Árvores de Decisão

Uma abordagem associada ao conceito de aprendizagem é a indução de ADs, conforme mostra o programa ID3 de QUINLAN (1986). O ID3 foi um dos primeiros algoritmos que constroem ADs, tendo sua elaboração baseada em sistemas de inferência e em conceitos de sistemas de aprendizagem. Este programa representa conceitos em forma de árvore, criando-as automaticamente a partir de instâncias positivas e negativas de um conceito¹¹. Esse algoritmo constrói uma AD a partir de um conjunto de dados de treinamento pela aplicação da estratégia de dividir e conquistar e um algoritmo *greedy* que utiliza uma relação de ganho como critério para a construção da árvore (QUINLAN, 1993).

ADs têm esse nome por produzirem uma lógica de classificação baseada em uma estrutura de árvore. A partir desta árvore é possível entender a lógica utilizada pelo programa para classificar as amostras. Devido à complexidade de muitos problemas e a dificuldade de serem conceituados, as ADs podem ser utilizadas para explicar tais problemas de forma clara (LEVINE, 1988).

O C4.5 é um programa de aprendizado simbólico que também gera classificadores na forma de ADs. Este seleciona um atributo de entrada, divide as

¹¹ Na verdade, a representação em árvore de decisão é mais genérica: as folhas podem denotar várias classes, e não apenas as positivas e negativas (ver Apêndice F).

instâncias de treinamento em sub-conjuntos correspondentes aos valores do atributo selecionado e calcula a relação de ganho para esse atributo. Esse processo é repetido para cada atributo de entrada das instâncias de treinamento. O C4.5 seleciona o atributo com melhor relação de ganho como raiz da AD. O algoritmo é aplicado recursivamente para formar as sub-árvores do modelo, terminando quando um dado contém somente o atributo classe (QUINLAN, 1993).

Após a construção da AD, o modelo resultante pode ser muito complexo e específico para o conjunto de dados de treinamento. Para melhorar a precisão e tornar a árvore inicial menos específica ao conjunto de dados de treinamento, o algoritmo C4.5 efetua um processo de poda da árvore, que consiste em retirar aqueles nós que não contribuem para a predição de novos valores, produzindo uma árvore de decisão menos complexa e com uma melhor generalização.

O C4.5 utiliza um método iterativo para construir ADs, dando preferência as árvores simples sobre as complexas, com base na teoria de que as árvores simples podem classificar com maior precisão entradas futuras (RICH, 1993).

4.2.1 Indução de Árvores de Decisão

Algoritmos que induzem ADs pertencem a família de algoritmos *Top Down Induction of Decision Trees*-TDIDT (MONARD E BARANAUSKAS, 2003).

Uma AD é uma estrutura de dados definida recursivamente como:

- Um nó folha que corresponde a uma classe;
- Um nó de decisão que contém um teste sobre algum atributo. Para cada resultado do teste existe uma aresta para uma subárvore. Cada subárvore tem a mesma estrutura que a árvore.

A seguir é apresentado um exemplo ilustrativo de uma árvore de decisão para o diagnóstico de um paciente. Cada elipse é um teste em um atributo para um dado conjunto de dados de pacientes. Cada retângulo representa uma classe, ou seja, o diagnóstico. Para diagnosticar (classificar) uma paciente, basta começar pela raiz, seguindo cada teste até que uma folha seja alcançada (ver Figura 4.1). O conhecimento contido nesta árvore é sintetizado nas regras da Figura 4.2.

É fácil perceber que árvore pode ser representada como um conjunto de regras. Cada regra tem seu início na raiz da árvore e caminha até uma de suas folhas.

Figura 4.1 – Árvore de decisões.

SE paciente se sente bem = sim Então
Classe = saudável
Senão
SE paciente tem dor = não Então
SE temperatura do paciente < 37 Então
Classe = saudável
SENAO {temperatura do paciente > 37}
Classe = doente
Fim SE
SENAO {paciente tem dor = sim}
Classe = doente
Fim SE
Fim SE

Figura 4.2 – Regras de conhecimento utilizadas

4.2.2 Poda

Poda é um sistema sofisticado que guia a seleção de uma resposta adequada a uma situação específica. Assim como se corta os galhos de uma árvore para que esta cresça com mais força, o mecanismo de poda guia as tomadas de decisão focalizando apenas aquelas regras relativas à solução do problema imediato (LEVINE, 1988).

Após a construção da AD, é possível que o classificador induzido seja muito específico para o conjunto de treinamento. Nesse caso, diz-se que o classificador superajustou os dados do treinamento, ou seja, ocorreu um *overfitting*.¹² Como os exemplos de treinamento são apenas uma amostra de todos os exemplos possíveis, é possível adicionar na árvore arestas que melhorem seu desempenho nos dados de treinamento, mas que pioram seu desempenho em um conjunto de teste.

Para tentar solucionar o problema de superajuste dos dados, alguns indutores podam a AD depois de induzi-la. Esse processo reduz o número de nós (testes) internos, reduzindo a complexidade da árvore enquanto produz um desempenho melhor que a árvore original.

Segundo LEVINE (1988) os indutores de ADs separam por si próprios o conjunto de exemplos em um conjunto de exemplos de treinamento que será utilizado para construir a AD e um conjunto de exemplos de poda, o qual é efetivamente utilizado para realizar o processo de poda. A pós-poda consiste em podar a árvore após a sua indução e a pré-poda é efetuada enquanto a AD é induzida.

4.2.3 Classificação

A AD, após construída, pode ser utilizada para classificar novos exemplos iniciando-se pela raiz da árvore e caminhando através de cada nó de decisão até que uma folha seja encontrada. Quando uma folha é encontrada, a classe do novo exemplo é dada pela classe daquela folha.

4.3 Exemplo da Construção de uma AD

Para exemplificar a construção de uma AD será apresentado o problema de decisão de passear ou não conforme a condição climática. Suponha um conjunto de instâncias consistindo de medidas diárias sobre as condições do tempo, na qual cada exemplo é composto pelos seguintes atributos:

- Aparência: assume os valores discretos “sol”, “nublado” ou “chuva”;
- Temperatura: um valor numérico indicando a temperatura em graus Celsius;
- Umidade: também um valor numérico indicando a porcentagem de umidade e;

¹² Ajuste em excesso ao conjunto de treinamento.

- Ventando: assume valores discretos “sim” ou “não” indicando se é um dia com vento.

A Tabela 4.1 mostra o conjunto de instâncias para a decisão de passear com base no clima.

Tabela 4.1 – Conjunto de Instâncias.

Exemplo N°	Aparência	Temperatura	Umidade	Ventando	Viajar?
T ₁	Sol	25	72	Sim	Vá
T ₂	Sol	28	91	Sim	Não_vá
T ₃	Sol	22	70	Não	Vá
T ₄	Sol	23	95	Não	Não_vá
T ₅	Sol	30	85	Não	Não_vá
T ₆	Nublado	23	90	Sim	Vá
T ₇	Nublado	29	78	Não	Vá
T ₈	Nublado	19	65	Sim	Não_vá
T ₉	Nublado	26	75	Não	Vá
T ₁₀	Nublado	20	87	Sim	Vá
T ₁₁	Chuva	22	95	Não	Vá
T ₁₂	Chuva	19	70	Sim	Não_vá
T ₁₃	Chuva	23	80	Sim	Não_vá
T ₁₄	Chuva	25	81	Não	Vá
T ₁₅	Chuva	21	80	Não	Vá

Para cada dia (instância), alguém rotulou cada medida diária como “vá”, se o tempo estiver bom o suficiente para uma viagem ao campo ou “não_vá”, caso contrário. Embora esse exemplo possua apenas duas classes, é importante lembrar que uma AD pode trabalhar com qualquer número k de classes. A indução da AD inicia considerando que o conjunto de treinamento T contém exemplos pertencentes a mais de uma classe. Assim, é necessário escolher um teste baseado em um único atributo, a escolha do atributo para particionar os exemplos depende de cada indutor. Para este exemplo, vamos escolher aparência como teste, tendo três possíveis resultados $\{O1, O2, O3\} = \{\text{sol, nublado, chuva}\}$. Então o treinamento T é dividido em três subconjuntos como é mostrado na Tabela 4.2 e na Figura 4.3.

O primeiro passo é escolher um teste que corresponda ao nó raiz da árvore e separar as instâncias segundo os atributos (arcos) referentes a este teste. Por exemplo, selecionando o teste sobre aparência, obtém-se a árvore mostrada na Figura 4.3 que corresponde a organização das instâncias apresentadas na Tabela 4.2.

Figura 4.3 – Representação esquemática da Tabela 4.2

Tabela 4.2 – Construindo uma AD a partir do conjunto de exemplos. Passo 1.

Teste	Exemplo	Aparência	Temperatura	Umidade	Vento	Viajar?
If aparência = sol	T1	sol	25	72	sim	vá
	T2	sol	28	91	sim	Não_vá
	T3	sol	22	70	não	vá
	T4	sol	23	95	não	Não_vá
	T5	sol	30	85	não	Não_vá
If aparência = nublado	T6	nublado	23	90	sim	vá
	T7	nublado	29	78	sim	vá
	T8	nublado	19	65	sim	Não_vá
	T9	nublado	26	75	não	vá
	T10	nublado	20	87	sim	vá
If aparência = chuva	T11	chuva	22	95	não	vá
	T12	chuva	19	70	sim	Não_vá
	T13	chuva	23	80	sim	Não_vá
	T14	chuva	25	81	não	vá
	T15	chuva	21	80	não	vá

Como pode ser notado na Tabela 4.2, cada subconjunto ainda contém exemplos pertencentes a várias classes, portanto é necessário escolher um outro teste baseado em um único atributo. Assumindo o atributo umidade para as subárvores “sol” e “nublado” e vento para a subárvore “chuva”. Cada subconjunto é agora particionado como é mostrado na Tabela 4.3 e a Figura 4.4 correspondente.

Figura 4.4 – Representação esquemática da Tabela 4.3

Tabela 4.3 – Construindo uma AD a partir do conjunto de exemplos. Passo 2.

Teste	Exemplo	Aparência	Temperatura	Umidade	Ventando	Viajar?
If aparência= sol e umidade ≤ 78	T1	sol	25	72	sim	vá
	T3	sol	22	70	não	vá
If aparência= sol e umidade > 78	T2	sol	28	91	sim	Não_vá
	T4	sol	23	95	não	Não_vá
	T5	sol	30	85	não	Não_vá
If aparência= nublado e umidade > 70	T6	nublado	23	90	sim	vá
	T7	nublado	29	78	não	vá
	T9	nublado	26	75	não	vá
	T10	nublado	20	87	sim	vá
If aparência= nublado e umidade ≤ 70 If aparência= chuva e ventando= não If aparência= chuva e ventando= sim	T8	nublado	19	65	sim	Não_vá
If aparência= chuva e ventando= não	T11	chuva	22	95	não	vá
	T14	chuva	25	81	não	vá
	T15	chuva	21	80	não	vá
If aparência= chuva e ventando= sim	T12	chuva	19	70	sim	Não_vá
	T13	chuva	23	80	sim	Não_vá

É possível notar que o exemplo (T8) satisfaz o teste “umidade ≤ 70”; todos os outros exemplos para a subárvore nublado pertencem à “classe = vá”. Isso pode indicar um *overfitting* dos dados e o indutor pode podar essa subárvore, como pode ser visto na Tabela 4.4 e na Figura 4.5 correspondente.

Figura 4.5 – Representação esquemática da Tabela 4.4. Passo 3.

Tabela 4.4 – Podando a AD. Passo 3.

Teste	Exemplo	Aparência	Temperatura	Umidade	Ventando	Viajar?
If $\text{aparência}=\text{sol}$ e $\text{umidade} \leq 78$	T1	sol	25	72	sim	vá
	T3	sol	22	70	Não	vá
If $\text{aparência}=\text{sol}$ e $\text{umidade} > 78$	T2	Sol	28	91	sim	Não_vá
	T4	Sol	23	95	Não	Não_vá
	T5	Sol	30	85	não	Não_vá
If $\text{aparência}=\text{nublado}$	T6	Nublado	23	90	Sim	Vá
	T7	Nublado	29	78	Não	vá
	T8	Nublado	19	65	Sim	Não_vá
	T9	Nublado	26	75	Não	Vá
	T10	Nublado	20	87	sim	vá
If $\text{aparência}=\text{chuva}$ e $\text{ventando}=\text{não}$	T11	Chuva	22	95	Não	Vá
	T14	Chuva	25	81	Não	Vá
	T15	chuva	21	80	não	vá
If $\text{aparência}=\text{chuva}$ e $\text{ventando}=\text{sim}$	T12	Chuva	19	70	Sim	Não_vá
	T13	chuva	23	80	sim	Não_vá

A poda da AD pode, em geral, melhorar o desempenho para instâncias não utilizadas na construção da AD. Isso pode parecer contra-intuitivo, uma vez que a poda descarta alguma informação (a instância T8 neste caso). Entretanto, quando o

aprendizado ocorre em instâncias contendo ruído¹³, um grau adequado de poda pode melhorar o desempenho para novas instâncias. De fato, a poda em geral elimina erros provenientes de ruídos em vez de descartar informação relevante (BRATKO, 1990).

4.4 Técnica *Support Vector Machines* (SVMs)

As Máquinas de Vetores Suporte (SVMs, do inglês - *Support Vector Machines*) correspondem a uma família de algoritmos de AM, que nos últimos anos tem recebido grande atenção. As principais características que estão popularizando sua utilização são a sua boa capacidade de generalização e robustez diante de dados de grande dimensão. As SVMs foram desenvolvidos por um matemático russo chamado Vladimir Vapnik na sequência da sua teoria da aprendizagem estatística (*statistical learning theory*).

Enquanto que outros algoritmos de AM (como ADs ou redes neurais - RNs) apenas procuram descrever com sucesso os dados de treino, SVMs vão um pouco mais longe, preferindo um modelo mais simples em detrimento de outros mais complexos. Esta medida de complexidade do modelo produzido é calculada utilizando a dimensão de Vapnik-Chervonenkis.

A técnica engloba uma classe de algoritmos de aprendizado baseados na teoria de aprendizado estatístico, combinando controle de generalização com uma técnica para tratar o problema da dimensionalidade¹⁴ (VAPNIK, 1998). Basicamente, é um algoritmo linear que constrói hiperplanos como superfícies de decisão de maneira que a fronteira de separação entre classes positivas e negativas seja maximizada (HAYKIN, 1999).

Atualmente, as SVMs possuem ainda a desvantagem de apenas resolverem problemas binários (que se possam caracterizar em 2 classes). No entanto, existe uma investigação no sentido de generalizar o algoritmo para n classes. Um exemplo de uma abordagem utilizada é a técnica de "um contra os outros". Nesta abordagem para cada classe constrói-se uma SVM que toma uma decisão sim/não para essa classe (a cada classe corresponde uma SVM). Ao classificar um exemplo de teste escolhe-se a SVM que diz sim para esse exemplo e que tem o hiperplano de decisão mais longe do exemplo de teste. A classe escolhida é a classe que corresponde a essa SVM. Um dos

¹³ Dados imperfeitos.

¹⁴ Os algoritmos de AM podem ter seu desempenho prejudicado ao trabalhar com conjunto de dados com um número elevado de atributos (alta dimensionalidade), necessitando da utilização de técnicas de seleção de atributos para a redução da dimensionalidade original dos dados.

desafios associados às SVMs consiste em descobrir, inventar e utilizar *kernels* que obtenham desempenho adequado nos problemas em estudo. Muito sucintamente, *kernels* são funções $k(x, x')$ que retornam uma medida de semelhança entre dois objetos.

4.5 Descrição do Algoritmo Genético Criado para o Trabalho

Os Algoritmos Genéticos pertencem a uma classe de algoritmo denominados Algoritmos Evolutivos. Esses caracterizam-se por trabalharem com populações de soluções para um problema ao invés de apenas uma única solução e mimetizarem o processo de seleção natural na busca pelas melhores soluções para um problema. A seguir descrevem-se as características principais de um AG.

De acordo com GOLDBERG (1989) Algoritmos Genéticos (AGs) são algoritmos de busca inspirados em mecanismos da seleção natural e da genética. A solução do problema é codificada em uma estrutura de dados, usualmente um vetor de *bits*, chamada de cromossomo. Vários destes cromossomos coexistem em um conjunto denominado população. Assim, uma população de cromossomos nada mais é que um conjunto de possíveis soluções. Esta população é criada, no início da execução do algoritmo, e mantida ao longo de várias iterações onde acontece uma sucessão de eventos semelhantes ao que ocorre na natureza. Os AGs utilizam funções aleatórias para seu funcionamento, entretanto diferem das buscas aleatórias uma vez que regiões do espaço de busca que se mostram mais promissoras são melhor exploradas. Esses algoritmos buscam mimetizar o processo de seleção natural, de forma que os indivíduos melhor adaptados sobrevivam. No caso de um AG, cada indivíduo da população é uma solução, assim, ao término de um determinado período de tempo, as soluções melhores adaptadas são mantidas na população e as demais são descartadas. Os AGs diferem dos métodos tradicionais de busca por manterem uma população de soluções potenciais. Métodos tradicionais geram uma seqüência de passos com os quais a solução é determinísticamente encontrada. Muitos deles utilizam informações adicionais, como derivadas e gradientes, para guiar a busca pontual pelo espaço de busca. Os AGs, por se valerem de vários pontos de busca, são menos susceptíveis a ótimos locais ao contrário dos métodos tradicionais (MITCHELL, 1997).

Os indivíduos da população são criados com base em características de outro(s) indivíduos da população em um processo denominado reprodução. A primeira população é gerada aleatoriamente. Por serem resultado de uma combinação dos

conteúdos dos progenitores, os indivíduos possuem semelhanças com esses (hereditariedade). A variação das características acontece justamente nessa combinação dos conteúdos. Pelo fato das informações dos pais serem combinadas, e não copiadas aos seus descendentes, as características podem se juntar em padrões inéditos na população, produzindo a variação necessária para se obter uma melhor solução.

A seguir apresenta-se o funcionamento de um AG simples (SGA). Inicialmente a forma com que as soluções potenciais são representadas deve ser definida. As variáveis de controle do problema, por exemplo, podem ser representadas por uma *string* binária de comprimento arbitrário dependente do problema. A esta *string* completa dá-se o nome de cromossomo e cada *bit* é chamado de gene. Esta população de m soluções (*strings*) binárias de comprimento k é inicializada de forma aleatória onde cada gene tem a mesma probabilidade de assumir qualquer um dos dois símbolos ("0", "1"). Esta etapa é chamada de inicialização.

Cada cromossomo é então avaliado, ou seja, a solução que ele carrega em forma de *string* binária é decodificada e transformada em um valor que reflete a qualidade desta resposta. Este valor é chamado de aptidão e é armazenado no próprio cromossomo.

Na etapa seguinte seleciona-se alguns indivíduos que serão responsáveis por gerar os novos elementos. Esta seleção é feita atribuindo-se uma probabilidade a cada cromossomo. Esta probabilidade é proporcional à aptidão do cromossomo. Cada par de progenitores é usado para se gerar um novo indivíduo. Esse processo é realizado por dois operadores de reprodução: *crossover* e mutação. O *crossover* utilizado por Goldberg é chamado de *crossover* de um ponto. Um número aleatório c , variando de 1 a $k - 1$ (onde k é o tamanho do cromossomo) é gerado. Copiam-se os genes de um dos progenitores, $P1$, do intervalo 1 até c . Os demais genes, $c + 1$ até k são copiados do outro progenitor $P2$. Se o mesmo processo for aplicado, mas trocando de lugar $P1$ por $P2$, é possível se gerar um segundo indivíduo. Assim os resultados deste processo são dois novos elementos.

Durante estas cópias de genes há uma probabilidade de erro. Esta falha tem o nome de mutação e a probabilidade de que este operador seja utilizado é igual para todos os genes do cromossomo, e definido arbitrariamente. Em outras palavras, se o valor a ser copiado para o novo indivíduo, em um determinado momento, fosse o símbolo '1', a mutação iria copiar erroneamente o valor '0' em seu lugar.

Estes dois operadores, *crossover* e mutação, são responsáveis pela variabilidade genética necessária à seleção. Combinando cromossomos e alterando de forma pontual seu conteúdo propicia uma pesquisa do espaço de busca que é fundamental para a qualidade da solução encontrada ao término da execução do algoritmo. Os novos indivíduos são gerados até que sua quantidade atinja a quantidade da população inicial, substituindo-a totalmente. O ciclo se encerra, retornando a etapa de avaliação da população. O algoritmo é finalizado quando o número de gerações pré-estabelecido foi alcançado.

Segue o algoritmo genético simples na sua forma original:

```
início  
gen := 0  
inicializa população aleatoriamente  
repete  
  poptemp = reproduz(pop)  
  pop = seleciona(poptemp)  
  gen := gen + 1  
até (gen >= maxgen)  
fim
```

Para o AG desenvolvido neste trabalho, todos os operadores de mutação e o operador de *crossover* utilizam o torneio de duas como forma de escolha de indivíduos a serem operados. É usado também o elitismo para selecionar o melhor indivíduo da população para que continue na nova população. A população é gerada a partir dos operadores empregados aos indivíduos da população anterior. Todos os indivíduos são expandidos pelos operadores de expansão. Foi verificado que qualquer pequena mudança nos indivíduos poderia gerar indivíduos com aptidão nula. Isto ocorre, pois a classe de proteínas usada como base para a avaliação é muito semelhante, ou seja, cada proteína se difere uma da outra por poucos caracteres diferentes. A classe é formada por proteínas mutantes de uma única proteína, formando assim um classe de proteínas semelhantes que se diferem por no máximo 5 caracteres. Como facilmente apareciam indivíduos com aptidão nula, estes foram “deletados” das populações ao longo das gerações e inserindo no lugar indivíduos novos e aleatórios com restrição de tamanho, o que aumenta a probabilidade de termos uma população melhor avaliada. Para evitar que a população tivesse muitos indivíduos repetidos, diminuindo assim a variabilidade da população, estes foram retirados das populações inserindo também no lugar indivíduos novos e aleatórios com restrição de tamanho.

Operadores de reprodução:

Mutacao1 = Altera um caracter aleatório em uma posição aleatória no individuo.

Mutacao2 = Deleta um caracter aleatório se o individuo possui mais de dois caracteres.

Mutacao3 = Troca posições vizinhas enquanto isto melhora a aptidão do individuo. Se piorar ou se percorrer todo o individuo então para a mutação.

Crossover = São escolhidos dois indivíduos que geram outros dois como filhos. Nestes indivíduos escolhidos (pais) faz-se dois cortes, um em cada pai, e junta-se a parte inicial do primeiro pai com a parte final do segundo pai resultando assim no primeiro filho. Analogamente é construído o segundo filho.

Expansao1 = O individuo é expandido com caracteres à direita a partir de ocorrências deste mesmo nas classes de proteínas que serve de base para a avaliação.

Expansao2 = É semelhante ao Expansão1, porém com a diferença da expansão ocorrer com caracteres à esquerda.

Inicialização da População:

A população é inicializada de forma aleatória e com restrição de comprimento para os indivíduos. Esta restrição é importante para que a população cresça mais rapidamente. O comprimento pode variar de dois até quatro caracteres.

Avaliação dos Indivíduos:

Na avaliação de um indivíduo compara-se o indivíduo com uma das classes de proteínas (patogênicas ou não patogênicas). Conta-se o número de vezes que o indivíduo aparece nesta classe e multiplica-se este número pelo comprimento do indivíduo. Em outras palavras, a aptidão do indivíduo é o número de ocorrências do indivíduo na classe de proteínas considerada, vezes seu comprimento.

CAPÍTULO 5 - MATERIAIS E MÉTODOS

Este Capítulo apresenta a ferramenta utilizada para a realização dos testes a partir das seqüências mutantes de aminoácidos da proteína hemoglobina, o *software* WEKA, (WITTEN E FRANK, 2000). As etapas necessárias para a realização dos testes serão apresentadas na Seção 5.2.

5.2 WEKA

O sistema WEKA (Waikato Environment Knowledge Analysis) foi desenvolvido na Universidade de Waikato na Nova Zelândia (WITTEN E FRANK, 2000). O sistema foi implementado a partir de uma linguagem de programação orientada a objeto (JAVA) amplamente disponível para as principais plataformas computacionais. O WEKA está sendo testado nos sistemas operacionais Windows, Linux, Unix e Macintosh.

Ao contrário da maioria das ferramentas de AM, a ferramenta WEKA apresenta uma interface gráfica para o usuário. O formato de arquivo aceito pela WEKA é o “arff” que, inicialmente apresenta os valores possíveis dos atributos da base (cabeçalho) e, em seguida, os dados propriamente dito, separados por vírgula (OLIVEIRA, 2002).

Esta ferramenta implementa inúmeros algoritmos com o intuito de facilitar a execução de várias técnicas de AM a partir de um mesmo formato de arquivo, que são: a biblioteca *MLC++* (*Machine Learning Library in C++*) e *WEKA*.

A ferramenta WEKA é formada pelo seguinte conjunto de pacotes: *attribute selection*, *classifiers*, *clustering*, *association rules*, *filters* e *estimators*. O pacote *Selection* é utilizado como o próprio nome indica, para selecionar atributos de uma base de dados para a extração de padrões. O pacote *classifiers* é formado por implementações de algoritmos de aprendizagem, tais como: IBK (*K-nearest neighbor*), *WEKA classifiers J48.J48* e *WEKA classifiers J48.PART* entre outros. O *WEKA cluster*

contém uma implementação de dois métodos de aprendizagem não supervisionada: *Cobweb* e o algoritmo EM. O pacote WEKA *associations* contém duas classes, *ItemSet* e *Apriori*, que juntos implementam esse algoritmo. O pacote *estimators* contém subclasses que são utilizadas pelo *Naive Bayes* para computar os diferentes tipos de distribuição de probabilidade. O pacote *filters* permite aos usuários selecionar um subconjunto de atributos ou selecionar um subconjunto de instâncias de dados baseada em algum critério.

A linguagem de programação JAVA permite uma interface uniforme para os diferentes algoritmos de aprendizado, além de métodos para pré e pós-processamento e a avaliação dos resultados a partir de qualquer banco de dados.

O sistema WEKA com as implementações dos principais algoritmos de aprendizado pode ser utilizado por dados na linha de comando. O programa também inclui uma variedade de ferramentas para a modificação do banco de dados, como por exemplo, o algoritmo para discretização. Pode ser criado no WEKA um conjunto de dados que realimentará um esquema de aprendizado e, em seguida, analisará o resultado do classificador e do seu desempenho sem que seja necessário escrever algum código de programa.

Este programa de aprendizado de máquina é bastante utilizado por possuir uma interface amigável, além de outras inúmeras vantagens como as descritas acima. Este *software* possui licença GNU (*General Public License*) e, portanto, pode ser utilizado gratuitamente. A Figura 5.1 a seguir representa a tela de apresentação do sistema WEKA.

A ferramenta WEKA dispõe também de métodos de meta aprendizagem que são utilizados para a construção de conjuntos de classificadores. Dentre os métodos disponíveis, destacam-se: *Bagging* e *Boosting* (WITTEN E FRANK, 2000).

As próximas Seções apresentam a organização dos dados de entrada e o processo de treinamento no WEKA exemplificando com instâncias referentes a seqüências de mutantes de hemoglobina.

Figura 5.1 – Apresentação do programa.

5.2.1 Organização dos Dados de Entrada

Anteriormente ao processo de treinamento do classificador utilizando o WEKA, foram organizados os dados de entrada para o programa a partir da sequência de aminoácidos da cadeia β da proteína hemoglobina (ver Figura 5.2). Cada atributo (aminoácido da sequência) possui um conjunto de características cujas intensidades são expressas em números reais. As características consideradas para a criação dos arquivos de entrada para o WEKA foram respectivamente (ver Seção 2.6.1):

- Ambiente químico dos aminoácidos na hemoglobina;
- Frequência do aminoácido em proteínas;
- Massa do resíduo do aminoácido;
- Ponto isoelétrico do aminoácido;
- Índice de hidropatia.

A massa do resíduo é o equivalente a massa molecular do aminoácido. O ponto isoelétrico é o ponto onde as cargas dos aminoácidos são eletricamente nulas. O índice de hidropatia foi uma escala utilizada onde há a combinação da hidrofobicidade¹⁵ e da

¹⁵ Hidrofobicidade: a cadeia lateral do aminoácido (grupo R) não apresenta afinidade pela água;

hidrofilicidade¹⁶ dos grupos R. A frequência de cada aminoácido em proteínas refere-se exatamente a um dado estatístico, em porcentagem, de quanto um aminoácido está presente na maioria das proteínas. Os ambientes químicos referem-se a uma estimativa que busca caracterizar os pares formados pelos aminoácidos e ambientes químicos.

Todos os arquivos de entrada possuem a estrutura descrita a seguir. A primeira linha contém a instância formada pela sequência dos aminoácidos da molécula da hemoglobina sem mutações, ou seja, a sequência original. As demais linhas correspondem as instâncias de sequências de hemoglobinas mutantes. Ao final de cada linha com uma sequência é adicionado um atributo correspondente a molécula indicando se esta é provoca patologia ou não. Neste trabalho, denomina-se “boa” as moléculas que são mutantes e não apresentam uma patologia e “ruim” as moléculas que são mutantes e apresentam uma patologia associada (ver Figura 5.2).

Os aminoácidos das sequências em cada linha do arquivo, quando trocados por valores reais correspondentes às várias características de cada aminoácido formam os arquivos de entrada diferentes.

1) Nüigata_beta 1(NA1) Val>Leu----- Boa 1 V _____ L= BOA

In one-letter code:

1	11	21	31	41	51
L	H	L	T	P	E
E	K	S	A	V	T
A	L	W	G	K	V
N	V	D	E	V	G
G	E	A	L	G	R
L	L	V	V	Y	
P	W	T	Q	R	F
F	E	S	F	G	D
L	S	T	P	D	A
V	M	G	N	P	K
V	...	60			
61	K	A	H	G	K
K	V	L	G	A	F
S	D	G	L	A	H
L	D	N	L	K	G
T	F	A	T	L	S
E	L	H	C	D	K
L	H	V	D	P	E
N	F	R	L	L	G
N	V	L	V	C	V
L	A	H	H	F	G
K	...	120			
121	E	F	T	P	P
V	Q	A	A	Y	Q
K	V	V	A	G	V
A	N	A	L	A	H
K	Y	H			

Sequência de hemoglobina sem mutações

1	V	H	L	T	P	E	E	K	S	A	V	T	A	L	W	G	K	V	N	V	D	E	V	G	G	E	A	L	G	R	L	L	V	V	Y
P	W	T	Q	R	F	F	E	S	F	G	D	L	S	T	P	D	A	V	M	G	N	P	K	V	...	60									
61	K	A	H	G	K	K	V	L	G	A	F	S	D	G	L	A	H	L	D	N	L	K	G	T	F	A	T	L	S	E	L	H	C	D	
K	L	H	V	D	P	E	N	F	R	L	L	G	N	V	L	V	C	V	L	A	H	H	F	G	K	...	120								
121	E	F	T	P	P	V	Q	A	A	Y	Q	K	V	V	A	G	V	A	N	A	L	A	H	K	Y	H									

Figura 5.2 – Sequência de aminoácidos da hemoglobina.

Os arquivos de entrada utilizados nos testes com o programa WEKA foram desenvolvidos no editor de texto chamado *word pad* disponível no sistema operacional

¹⁶ Hidrofilicidade: a cadeia lateral do aminoácido (grupo R) apresenta afinidade pela água.

windows 98. Qualquer outro editor poderia ser utilizado para criar a entrada desde que os dados sejam salvos no formato texto, ou seja, com a extensão txt. A Figura 5.3 mostra o início de um arquivo de entrada. Em cada linha é definido um atributo (por exemplo, campo 1) e a sua direita é indicado de que tipo é o atributo (real). A Figura 5.4 apresenta as últimas linhas de definição de atributos e apresenta o campo de dados (*@ data*) do arquivo de entrada. Cada linha corresponde a informações de uma variante de hemoglobina, cada número na linha corresponde a uma característica de um aminoácido, em os respectivos aminoácidos aparecem na sequência. A Figura 5.5 ilustra o final das linhas no campo de dados do arquivo de entrada, mostrando como é definida a classe (BOA ou RUIM) de cada variante.

Figura 5.3 – Definição dos atributos no arquivo de entrada.

Figura 5.4 – Campo de dados do arquivo de entrada.

Figura 5.5 – Exemplo do final de linhas no campo de dados, mostrando a classe de cada variante.

5.2.2 Procedimento para a Geração do Programa

Abaixo menciona-se os passos a serem seguidos para a realização dos testes com os arquivos de entrada referentes as variantes de hemoglobina utilizando o programa WEKA:

- 1) Inicialização do programa com a tela principal (Figura 5.6a);
- 2) Busca do arquivo referente à entrada de dados (Figura 5.6b);
- 3) Seleção do arquivo (Figura 5.7a);
- 4) Visualização do arquivo de entrada (Figura 5.7b);
- 5) Escolha do teste estatístico para análise dos resultados (Figura 5.8a);
- 6) Seleção do algoritmo de AM J48 (correspondente ao algoritmo de Árvores de Decisão C4.5) – (Figura 5.8b);
- 7) Verificação dos resultados obtidos com o algoritmo de AM C4.5 por meio árvore de decisão construída (Figura 5.9a);
- 8) Visualização da resposta obtida pelo C4.5 (Figura 5.9a) e (Figura 5.9b).

(a)

(b)

Figura 5.6 – (a) Tela de inicialização do programa, (b) Busca do arquivo.

(a)

(b)

Figura 5.7 – (a) Escolha do arquivo a ser aberto, (b) Arquivo aberto.

(a)

(b)

Figura 5.8 – (a) Tela para escolha do teste estatístico para análise dos resultados, (b) Tela para escolha do algoritmo de AM

(a)

(b)

Figura 5.9 – (a) Resultados do algoritmo C4.5 mostrando a AD formada, (b) Análise estatística produzida pelo C4.5.

CAPÍTULO 6 - RESULTADOS

Os resultados apresentados nessa seção utilizam o algoritmo de AM C4.5 para construção de AD, comparando esses resultados com os produzidos por outros algoritmos de AM tais como o SMO, Kstar, Random Forest (RF) e o FLR (*Fuzzy Lattice Reasoning*) disponíveis no WEKA. Deve-se observar que esses algoritmos não geram um esquema que permite o entendimento da lógica de classificação utilizada. Por outro lado, o algoritmo C4.5 permite a visualização desta lógica por meio da AD.

Para a realização dos experimentos foi utilizado um conjunto de dados de proteínas disponíveis nos bancos de dados acessados por meio dos seguintes *sites*: <http://us.expasy.org> e <http://globin.cse.psu.edu>. A partir desses bancos de dados foram adquiridas seqüências mutantes da cadeia β da proteína hemoglobina associadas ou não a uma patologia, bem como a seqüência original da hemoglobina, totalizando 383 seqüências da proteína hemoglobina.

A Figura 6.1 ilustra duas seqüências de hemoglobina (cadeia β com 146 aminoácidos), sendo uma mutante e a outra não. A seqüência mutante apresentada na Figura 6.1 foi classificada como BOA, por não apresentar uma patologia associada. Em vermelho é destacado o aminoácido mutante, uma leucina (L) na posição 1 desta seqüência e na seqüência original, a posição 1 contém uma valina (V).

Seqüência mutante.

1 L H L T P E E K S A V T A L W G K V N V D E V G G E A L G R L L V V Y
 P W T Q R F F E S F G D L S T P D A V M G N P K V K A H G K K V L G A F S D
 G L A H L D N L K G T F A T L S E L H C D K L H V D P E N F R L L G N V L V C
 V L A H H F G K E F T P P V Q A A Y Q K V V A G V A N A L A H K Y H 146

Seqüência original sem mutações.

1 V H L T P E E K S A V T A L W G K V N V D E V G G E A L G R L L V V Y
 P W T Q R F F E S F G D L S T P D A V M G N P K V K A H G K K V L G A F S D
 G L A H L D N L K G T F A T L S E L H C D K L H V D P E N F R L L G N V L V C
 V L A H H F G K E F T P P V Q A A Y Q K V V A G V A N A L A H K Y H 146

Figura 6.1 – Seqüências de hemoglobina mutante (denominada Niigata).**Fonte:** <http://globin.cse.psu.Edu/hbvar>

Os experimentos realizados seguiram a metodologia *10-fold-cross-validation* (MONARD E BARANAUSKAS, 2003). De acordo com essa metodologia, as seqüências são aleatoriamente divididas em dez grupos de tamanho aproximadamente iguais. Um grupo é utilizado como conjunto de teste e os nove grupos restantes são utilizados como seqüências de treinamento. Após o treinamento utilizando os nove grupos, o grupo de teste é apresentado ao sistema e a média dos erros calculada. Esse processo é realizado dez vezes, em cada vez, considerando-se cada um dos grupos como de teste e os demais como grupos de treinamento.

As simulações foram realizadas em um computador Pentium III, processador 750 MHz, com sistema operacional Windows 98, 198 Mbytes de memória e HD de 20 Gigabytes.

Para o treinamento dos classificadores, cada arquivo de entrada possuía informações referentes as trezentos e oitenta e três seqüências de aminoácidos da cadeia β disponíveis no *site* <http://globin.cse.psu.Edu/hbvar>. Foram utilizados diferentes arquivos de entrada com informações referentes às características (afinidades) de cada resíduo dos aminoácidos (ver Seção 5.2.1). Inicialmente foram considerados doze arquivos de entrada, a partir dos quais realizaram-se doze testes buscando identificar os conjuntos de informações que produziram os melhores resultados com os classificadores. As informações referentes aos resíduos utilizados em cada arquivo de entrada são descritas a seguir:

- Teste 1) Código de uma letra (ver Tabela 6.1);
- Teste 2) Massa (ver Apêndice D);
- Teste 3) Ponto isoeletrico (ver Apêndice D);
- Teste 4) Índice de hidropatia (ver Apêndice D);
- Teste 5) Frequência dos resíduos nas proteínas (ver Apêndice D);
- Teste 6) Massa em conjunto com seus pontos isoeletricos;
- Teste 7) Massa em conjunto com a frequência dos resíduos;
- Teste 8) Massa, ponto isoeletrico e as frequência dos resíduos nas proteínas;
- Teste 9) Ambientes químicos calculados para a cadeia β da molécula de hemoglobina (ver Apêndice E);
- Teste 10) Diferença entre a massa de cada resíduo da sequência mutante e a massa do mesmo resíduo na sequência original (não mutante);
- Teste 11) Diferença entre o ambientes químico dos aminoácidos de cada resíduo da sequência mutante e o ambiente químico do mesmo resíduo na sequência original (não mutante);
- Teste 12) Massa, ponto isoeletrico, frequência dos resíduos nas proteínas, índice de hidropatia e os ambientes químicos.

Na Tabela 6.1 são apresentados os resultados obtidos considerando-se os testes de 1 a 12 com a cadeia β das variantes de hemoglobina para construção de ADs (C4.5). Para comparação, são apresentados os resultados obtidos por outros algoritmos de AM disponíveis no WEKA (*Kstar*, RF, SMO e o FLR). De acordo com a Tabela 6.1, ambos os algoritmos apresentaram resultados insatisfatórios, mostrando que as diferentes características dos aminoácidos, bem como as combinações dessas, não possibilitaram melhora significativa no desempenho utilizando tanto o C4.5 quanto os demais algoritmos empregados.

Tabela 6.1 – Resultados produzido pelo C4.5, SMO, RF, *Kstar* e FLR.

	Erro Médio											
	Teste 1	Teste 2	Teste 3	Teste 4	Teste 5	Teste 6	Teste 7	Teste 8	Teste 9	Teste 10	Teste 11	Teste 12
C4.5	48.5%	42.4%	47.6%	47.6%	43.2%	47.8%	47.7%	48.2%	47.6%	46.3%	47.7%	47.7%
SMO	23.2%	35.0%	42.7%	37.3%	34.3%	38.1%	33.7%	32.9%	41.1%	39.0%	41.0%	31.7%
RF	41.5%	26.1%	30.8%	36.3%	27.9%	30.2%	30.3%	32.5%	36.5%	31.1%	31.6%	36.0%
Kstar	27.1%	28.5%	31.3%	34.7%	28.8%	30.4%	26.7%	24.3%	35.1%	30.2%	35.6%	30.6%
FLR	30.3%	31.3%	32.5%	32.9%	25.9%	26.9%	25.8%	24.5%	35.1%	28.1%	33.0%	25.0%

Buscando novos conjuntos de informações que pudessem resultar em um melhor desempenho dos algoritmos de AM, foram realizados testes considerando também a cadeia α da hemoglobina. A cadeia α apresenta um número menor de mutações e, conseqüentemente, é menor o número de mutações que estão associadas a uma patologia nesta cadeia.

As variantes da molécula de hemoglobina da cadeia β e também da cadeia α foram separadas de acordo com certas características presentes nos indivíduos. Uma destas características os dividem em: indivíduos heterozigotos (indivíduos com par de alelos diferentes) e em indivíduos homozigotos (indivíduos com par de alelos idênticos). Estas características dos indivíduos são importantes devido à manifestação ou não de uma patologia. A hemoglobina da maioria das pessoas é conhecida como hemoglobina A (HbA). Porém, existem algumas variantes químicas de hemoglobina A que são encontradas em um número pequeno de indivíduos e uma dessas variantes, a hemoglobina S (HbS) está envolvida na doença falciforme. O par de alelos¹⁷ com dominância incompleta, responsável por esses tipos de hemoglobina é Hb^AHb^S(indivíduos heterozigotos). A maioria das pessoas pertence ao genótipo Hb^AHb^A. Os indivíduos com anemia falciforme são do genótipo Hb^SHb^S (indivíduos homozigotos), sendo caracterizados por um conjunto de sintomas principalmente uma anemia hemolítica crônica (GARDNER, 1987).

¹⁷ Alelos são formas alternativas de um mesmo gene e que, conseqüentemente ocupam mesmo loco em cromossomos homólogos. Os efeitos genéticos destes alelos dependem de suas relações de dominância. Estes alelos têm origem nas mutações, que são capazes de causar alterações estruturais nos genes de tal forma que é possível ocorrer mais de um par de alelos para um determinado gene.

Os testes de 12 a 14 representados nas Tabelas 6.2, 6.3 e 6.4 mostram os resultados obtidos a partir dessas informações. Para a realização desses testes utilizou-se somente o arquivo de entrada referente aos ambientes químicos, por serem bioquimicamente considerados mais apropriados para caracterizar os aminoácidos de uma proteína (BOWIE, 1991).

O teste com homozigotos utilizando a cadeia α não foi realizado devido a existência de poucos indivíduos com esta característica, inviabilizando a utilização dos classificadores disponíveis no WEKA.

O teste com heterozigotos utilizando a cadeia α apresentou um acerto maior, porém as classes (BOA e RUIM, que correspondem respectivamente ao conjunto de variantes com ou sem patologia associada) estão desbalanceadas, o que deve ter contribuído para o melhor desempenho em relação aos testes apresentados na Tabela 6.1. Quando são apresentados aos classificadores classes com números de padrões desproporcionais, como por exemplo, nos testes com heterozigotos utilizando a cadeia α , os algoritmos de classificação conseguem um menor erro médio em relação aos testes da Tabela 6.1, pois o índice de acerto tenderá a ser proporcional a porcentagem de elementos da maior classe, mesmo utilizando um classificador aleatório.

Tabela 6.2 – Resultados do Teste 13 considerando ambientes químicos e a cadeia α de indivíduos heterozigotos.

Instâncias = 139	Boas = 114
Atributos = 142	Ruins = 25
	Nós = 1
Ads	Folhas = 1
Erro Médio	
C4.5 = 17.9%	
SMO = 18.7%	
Random Forest = 17.9%	
Kstar = 19.4%	
FLR = 23.7%	

Tabela 6.3 – Resultados do Teste 14 considerando ambientes químicos e a cadeia β de indivíduos homozigotos.

Instâncias = 15	Boas = 7
Atributos = 147	Ruins = 8
ADs	Nós = 5
	Folhas = 3
Erro Médio	
C4.5 = 53.3%	
SMO = 53.3%	
Random Forest = 46.0%	
Kstar = 40.0%	
FLR = 40.0%	

Tabela 6.4 – Resultados do Teste 15 considerando ambientes químicos e a cadeia β de indivíduos heterozigotos.

Instâncias = 138	Boas = 88
Atributos = 147	Ruins = 50
ADs	Nós = 1
	Folhas = 1
Erro Médio	
C4.5 = 36.2%	
SMO = 29.7%	
Random Forest = 33.3%	
Kstar = 26.8%	
FLR = 31.9%	

Buscando ainda aumentar o desempenho dos algoritmos de AM, as informações das seqüências das variantes foram preprocessadas selecionando-se as posições consideradas mais relevantes para a funcionalidade da proteína. As posições escolhidas são consideradas importantes devido à alta freqüência em que os mesmos aminoácidos aparecem nas diversas espécies da família das globinas conforme apresentado no estudo de (BASHFORD, 1987). A partir deste trabalho, foram extraídas 49 posições da seqüência de aminoácidos de importância para as cadeias α e β .

As Tabelas 6.5, 6.6 e 6.7 expressam os resultados dos testes 16, 17 e 18 realizados com as cadeias α e β da proteína hemoglobina utilizando-se apenas variantes de indivíduos homozigotos e heterozigotos. Os resultados mostram novamente que os dados referentes à cadeia α de heterozigotos permitem um melhor desempenho para todos os algoritmos de AM utilizados. O C4.5 mostrou um acerto de 82.1%, o melhor resultado encontrado entre todos os testes apresentados. No entanto, as classes também

estão desbalanceadas neste caso, o que deve ter contribuído para o melhor desempenho obtido.

Tabela 6.5 – Resultados do Teste 16 considerando ambientes químicos e a cadeia β de indivíduos heterozigotos e as principais posições de acordo com BASHFORD (1987).

Instâncias = 138	Boas = 88
Atributos = 50	Ruins = 50
ADs	Nós = 1
	Folhas = 1
Erro Médio	
C4.5 = 36.2%	
SMO = 30.4%	
Random Forest = 26.0%	
Kstar = 26.0%	
FLR = 39.1%	

Tabela 6.6 – Resultados do Teste 17 considerando ambientes químicos e a cadeia β de indivíduos homozigotos e as principais posições de acordo com BASHFORD (1987).

Instâncias = 15	Boas = 7
Atributos = 50	Ruins = 8
ADs	Nós = 1
	Folhas = 1
Erro Médio	
C4.5 = 66.6%	
SMO = 60.0%	
Random Forest = 46.6%	
Kstar = 60.0%	
FLR = 46.7%	

Tabela 6.7 – Resultados do Teste 18 considerando ambientes químicos e a cadeia α de indivíduos heterozigotos e as principais posições de acordo com BASHFORD (1987).

Instâncias = 139	Boas = 114
Atributos = 50	Ruins = 25
ADs	Nós = 1
	Folhas = 1
Erro Médio	
C4.5 = 17.9%	
SMO = 17.3%	
Random Forest = 18.7%	
Kstar = 17.9%	
FLR = 25.2%	

Buscando reduzir o efeito do desbalanceamento um novo conjunto de dados foi organizado considerando conjuntamente as cadeias α e β . A Tabela 6.8 mostra que os resultados também são insatisfatórios quando esse conjunto de dados é utilizado.

Tabela 6.8 – Resultados do Teste 19 considerando ambientes químicos, a cadeia α e a cadeia β de indivíduos homozigotos e heterozigotos com as principais posições de acordo com BASHFORD (1987).

Instâncias = 293	Boas = 187
Atributos = 50	Ruins = 106
ADs	Nós = 1
	Folhas = 1
Erro Médio	
C4.5 = 38.4%	
SMO = 38.4%	
Random Forest = 32.5%	
Kstar = 35.0%	
FLR = 36.1%	

O AG apresentado no Capítulo 4 foi empregado na seleção das posições mais relevantes de cada sequência buscando salientar as informações na entrada dos classificadores, de forma a aumentar a eficiência dos mesmos. Desta forma, o AG desenvolvido funciona como um Filtro reduzindo a presença de informações que seriam irrelevantes para o processo de classificação. A seguir, na Seção 6.2, são apresentados os testes e os resultados obtidos com o AG. A partir desses resultados, são produzidos quatro arquivos de entrada diferentes que compõem quatro Filtros. Esses arquivos foram denominados de Filtro1, Filtro 3 e Filtro 4. O Filtro 2 também apresentado na Seção 6.2 é base para a elaboração do Filtro 3.

6.2 Resultados com o AG Proposto

Os testes apresentados nesta Seção utilizam arquivos de entrada referentes a cadeia β , indivíduos homozigotos e heterozigotos, uma vez que o número de variantes na cadeia β é maior e as classes (BOA/RUIM) estão relativamente balanceadas.

O Filtro 1 foi construído como segue. Os melhores indivíduos (trechos da sequência de aminoácidos da hemoglobina) classificados pelo AG, ou seja, os que aparecem mais vezes na população e tem o maior tamanho, são selecionados como

padrões da amostra de seqüências. Esses padrões são, então, utilizados para modificar as seqüências de aminoácidos da hemoglobina do arquivo de entrada como descrito a seguir. Primeiramente, os indivíduos selecionados são identificados em cada uma das seqüências do arquivo de entrada. Então, as posições, em cada trecho da seqüência que não correspondem a um indivíduo identificado são substituídas pelo caractere “?”, que significa qualquer um dos atributos (qualquer um dos vinte aminoácidos). Desta forma, as posições com “?” não são utilizadas para a construção do classificador. A Figura 6.2 ilustra o processo de alteração das seqüências para a construção do arquivo de entrada.

Figura 6.2 – Exemplo de seqüência de aminoácido alterada pelo Filtro 1.

Os resultados produzidos utilizando-se o arquivo de entrada modificado por “?” não foram satisfatórios, pois os classificadores obtiveram um alto erro no processo de validação (ver Tabela 6.9).

Tabela 6.9 – Resultados do Teste 20 considerando o Filtro 1.

Instâncias = 383	Boas = 197
Atributos = 147	Ruins = 186
ADs	Nós = 1
	Folhas = 1
Erro Médio	
C4.5 = 48.6%	
SMO = 48.5%	

Buscando ressaltar a presença das mutações para facilitar o processo de aprendizado do classificador foi desenvolvido o Filtro2. Deve-se observar que este Filtro não utiliza resultados obtidos pelo AG desenvolvido. O Filtro 2 identifica, primeiramente, apenas os aminoácidos mutantes presentes nas seqüências da hemoglobina. Em seguida, substitui os aminoácidos mutantes por um novo atributo (o caractere “*”), representando uma mutação.

Os resultados produzidos utilizando-se o Filtro 2 apresentaram uma sensível melhora em relação ao teste anterior (ver Tabela 6.10).

Tabela 6.10 – Resultados do Teste 21 considerando o Filtro 2.

Instâncias = 383	Boas = 197
Atributos = 147	Ruins = 186
ADs	Nós = 799
	Folhas = 761
Erro Médio	
C4.5 = 30.4%	
SMO = 20.9%	

Com base nos Filtros 1 e 2 foi elaborado o Filtro 3. Este altera as seqüências do arquivo de entrada com “*” na posição dos aminoácidos mutantes e “?” nas posições que não correspondem a indivíduos obtidos pelo AG proposto. Os resultados obtidos (ver Tabela 6.11) apresentaram baixo erro, mostrando a adequação deste Filtro.

Tabela 6.11 – Resultados do Teste 22 considerando o Filtro 3.

Instâncias = 383	Boas = 197
Atributos = 147	Ruins = 186
ADs	Nós = 22
	Folhas = 21
Erro Médio	
C4.5 = 3.6%	
SMO = 2.1%	

Buscando melhorar os resultados obtidos foi desenvolvido o Filtro 4. Os melhores indivíduos novamente são classificados pelo AG, ou seja, os que aparecem mais vezes na população e tem o maior tamanho, são selecionados para modificar as seqüências de aminoácidos do arquivo de entrada. Este arquivo é alterado como segue. Primeiramente, os indivíduos selecionados são identificados em cada uma das

seqüências do arquivo de entrada. Cada posição de cada seqüência que não corresponde a um indivíduo identificado é substituído pelo caractere “*”, correspondente a um atributo adicional, indicando a irrelevância da posição. Desta forma, no Filtro 4 as posições com “*” serão utilizadas para a construção do classificador, diferentemente do que ocorre quando se utiliza o Filtro 1.

Os resultados produzidos com base no Filtro 4 foram satisfatórios, pois os classificadores obtiveram erro relativamente baixo no processo de validação (ver Tabela 6.12).

Tabela 6.12 – Resultados do Teste 23 considerando o Filtro 4.

Instâncias = 383	Boas = 197
Atributos = 147	Ruins = 186
ADs	Nós = 127
	Folhas = 121
Erro Médio	
C4.5 = 1.8%	
SMO = 1.6%	

Os resultados com o Filtro 4 (1.8% de erro) são melhores do que os obtidos com o Filtro 3 (3.6% de erro). No entanto, a AD produzida a partir do arquivo de entrada gerado pelo Filtro 4 possui 127 nós e 121 folhas, enquanto a AD obtida com a utilização do Filtro 3 possui 22 nós e 21 folhas. Esta AD é menor e, portanto, menos específica, possuindo uma maior capacidade de generalizar (ver Capítulo 4). Assim, o classificador mais adequado obtido pelo C4.5 corresponde a AD de 22 nós.

Apesar dos resultados motivadores obtidos com os Filtros 3 e 4, uma análise de robustez do AG desenvolvido foi realizada para verificar a capacidade (frequência) do mesmo em reproduzir os resultados obtidos, uma vez que AGs são processos estocásticos. Verificou-se, então, que os resultados gerados pelo AG não reproduziam Filtros que aumentavam significativamente o desempenho dos classificadores apresentados nas Tabelas 6.11 e 6.12.

Após análise intensiva dos resultados produzidos pelo AG, verificou-se uma tendência deste (“*bias*”) em eliminar certas cadeias gerando resultados inconsistentes. Após a eliminação do efeito “*bias*” do AG, novos experimentos foram realizados, os resultados são apresentados a seguir.

O AG reformulado foi utilizado de forma semelhante á apresentada anteriormente. Os melhores indivíduos (trechos da seqüência de aminoácidos da

hemoglobina) gerados pelo AG, ou seja, os que aparecem mais vezes na população e tem o maior tamanho, são selecionados como padrões da amostra de seqüências. Com base nesses padrões, são desenvolvidos os Filtros 5, 6, 7 e 8.

O Filtro 5 correspondente ao caractere “?” substitui os caracteres de trechos de seqüências que não correspondem aos padrões encontrados pelo AG. Além disso, um caractere “*” substitui os aminoácidos mutantes. A Figura 6.3 ilustra o processo de alteração das seqüências para a construção do arquivo de entrada.

Figura 6.3 – Processo de alteração das seqüências de acordo com o Filtro 5.

Os resultados obtidos com o Filtro 5 não foram satisfatórios, pois os classificadores obtiveram um alto erro no processo de validação (ver Tabela 6.13).

Tabela 6.13 – Resultados do Teste 24.

Instâncias = 383	Boas = 197
Atributos = 147	Ruins = 186
ADs	Nós = 736
	Folhas = 701
Erro Médio	
C4.5 = 36.8%	
SMO = 25.6%	

No Filtro 6 substitui-se os trechos considerados não padrão segundo o AG pelos caracteres “*”, além disso o caractere “*” substitui os aminoácidos mutantes. A Figura 6.4 ilustra este processo de alteração das seqüências.

Figura 6.4 –A alteração das seqüências produzida pelo Filtro 6.

Os resultados produzidos utilizando-se o Filtro 6 foram satisfatórios, pois os classificadores obtiveram erro relativamente baixo no processo de validação (ver Tabela 6.14).

Tabela 6.14 – Resultados do Teste 25.

Instâncias = 383	Boas = 197
Atributos = 147	Ruins = 186
ADs	Nós = 568
	Folhas = 541
Erro Médio	
C4.5 = 7.8%	
SMO = 5.7%	

No Filtro 7 substitui-se por caracteres “*” os aminoácidos dos trechos obtidos como não padrão pelo AG e mantém-se o aminoácido no local da mutação. A Figura 6.5 ilustra o processo de alteração das seqüências para a construção do arquivo de entrada.

Figura 6.5 – A alteração do arquivo de entrada produzido pelo Filtro 7.

Os resultados obtidos pelo Filtro 7 apresentaram baixo erro mostrando a adequação deste teste (ver Tabela 6.15).

Tabela 6.15 – Resultados do Teste 26.

Instâncias = 383	Boas = 197
Atributos = 147	Ruins = 186
	Nós = 358
ADs	Folhas = 341
Erro Médio	
C4.5 = 21.4%	
SMO = 4.7%	

No Filtro 8 utilizou-se os caracteres “*” para substituir os aminoácidos dos trechos considerados não padrão pelo AG e as posições de aminoácidos mutantes foram substituídas pelo caractere “\$” (representando qualquer outro símbolo diferente dos vinte aminoácidos). A Figura 6.6 ilustra o processo de alteração das seqüências.

Figura 6.6 – Processo de alteração das seqüências de acordo com o Filtro 8.

Os resultados produzidos com o Filtro 8, apresentaram baixo erro quando classificados com os algoritmos de AM (ver Tabela 6.16).

Tabela 6.16 – Resultados do Teste 27.

Instâncias = 383	Boas = 197
Atributos = 147	Ruins = 186
	Nós = 573
ADs	Folhas = 547
Erro Médio	
C4.5 = 7.8%	
SMO = 6.0 %	

Outros testes foram realizados substituindo por “*” os trechos considerados padrões ao invés de não padrão como nos Filtros 5, 6, 7 e 8. Os resultados obtidos foram semelhantes aos obtidos com esses Filtros. Esses últimos resultados revelam que a separação dos trechos de seqüências em duas classes (padrão e não padrão) é o fator importante que permite aumentar o desempenho dos classificadores.

CAPÍTULO 7 - CONCLUSÃO

Buscando aumentar a velocidade de obtenção de fármacos, bem como as qualidades dos mesmos, pesquisas têm sido desenvolvidas para a realização em laboratório da evolução de moléculas *in vitro*. Por meio de mutações forçadas, o método busca novas propriedades de moléculas conhecidas ou de moléculas desconhecidas. Para se obter moléculas apropriadas de uma certa proteína, criar-se o maior número possível de moléculas mutadas e separa-se as moléculas mais adequadas de acordo com suas características. Tais técnicas requerem muito tempo e aporte financeiro.

Neste contexto, este trabalho investiga técnicas de computacionais avançadas buscando auxiliar os processos laboratoriais de evolução *in vitro*. A redução de tempo e custo financeiro é um fator consideravelmente importante em evolução de moléculas *in vitro*. Com a simulação computadorizada poderia-se reduzir os gastos com materiais, bem como obter resultados mais rapidamente.

A utilização de técnicas e ferramentas de computação para a resolução de problemas da biologia, dentre as diversas áreas, tem se mostrado muito promissora, principalmente na biologia molecular. Técnicas de AM são cada vez mais empregadas para tratar problemas em Biologia Molecular, por sua capacidade de aprender automaticamente a partir de grandes quantidades de dados e produzir hipóteses úteis. O AM estuda como construir programas de computador que melhorem seu desempenho em alguma tarefa por meio da experiência (SOUTO, 2003). No entanto, o número de trabalhos envolvendo técnicas de AM para a classificação de seqüências de proteínas é relativamente pequeno. TSUNODA. E LOPES (2003) desenvolveram um trabalho utilizando uma técnica de AM, em especial AGs, para a análise e detecção de seqüências idênticas de aminoácidos em enzimas com o intuito de implementar uma classificação automática. SELBIG et al (1991), aplicaram técnicas de AM, em especial, ADs para selecionar propriedades nos aminoácidos que fossem a mais significativa, descrevendo uma posição na seqüência de aminoácidos que também fosse determinada

por tal característica. MADDOURI E ELLOUMI (2000), os autores utilizaram um algoritmo de aprendizado desenvolvido por eles mesmos para fazer a classificação de seqüências biológicas baseados em um grande banco de dados. BARISIC et al (2002), utilizaram dados analíticos bioquímicos, morfológicos e parâmetros anamnéticos para juntamente com os métodos de AM, que neste caso foi o uso do algoritmo C4.5 presente no *Software* WEKA para quantificar os fatores de risco em pacientes assintomáticos de uma patologia denominada amiloidose, causada pela hemodiálise.

Neste trabalho, escolheu-se investigar as moléculas mutantes de hemoglobina, uma vez que a quantidade de informações disponíveis sobre a mesma é bastante extensa na literatura. Buscando reduzir as dificuldades inerentes ao processo de evolução *in vitro*, a evolução *in silício* investigaria técnicas computacionais capazes de simular mutações nas moléculas e determinar computacionalmente a funcionalidade das mesmas. O trabalho utilizou-se de estruturas primárias, que já foram também utilizadas em outros estudos como os descritos acima. Foram considerados os aminoácidos da cadeia α e da cadeia β de proteínas hemoglobina com mutações que provocaram ou não, uma patologia em indivíduos heterozigotos bem como em indivíduos homozigotos.

Para determinação da funcionalidade de mutantes ou variantes de moléculas de hemoglobina a partir de suas seqüências de aminoácidos, foi proposto a utilização de técnicas de computação avançadas como ADs, SVM e também AGs. As atividades desenvolvidas neste trabalho de pesquisa concentraram-se no aprendizado do software WEKA e na elaboração dos bancos de dados a partir de informações compiladas da literatura peculiares aos aminoácidos da proteína considerada na pesquisa (LEHNINGER, 1976), <http://us.expasy.org> e <http://globin.cse.psu.edu>.

Os resultados apresentados no Capítulo 6 mostram que as técnicas computacionais investigadas não produzem classificadores capazes de distinguir adequadamente entre seqüências de aminoácidos patogênicas e não patogênicas. Para obtenção de classificadores eficientes desenvolveu-se Filtros (baseados em AGs) capazes de revelar os trechos de seqüências menos significativos.

Com base nos Filtros 6, 7 e 8 (ver Capítulo 6) são produzidos classificadores relativamente eficientes. Esses Filtros identificam os trechos de cada seqüência que mais freqüentemente repetem-se em todo o conjunto de seqüências considerado. Pode-se supor, então, que os trechos restantes são os que permitem diferenciar uma seqüência da outra. Desta forma, esses Filtros permitem que os classificadores foquem nos trechos mais relevantes, aumentando o desempenho desses algoritmos.

Deve-se observar que a AD produzida a partir do Filtro 7 não é tão eficiente quanto as ADs obtidas com os Filtros 6 e 8. No entanto, a AD relativa ao Filtro 7 é significativamente menor (358 nós) que as demais (573 e 568 nós). Isso indica que essa árvore pode ter uma maior capacidade de generalização (ver Capítulo 4), característica importante para o classificador obter sucesso quando apresentado as novas entradas (variantes de hemoglobina). Além disso, o Filtro 7 preserva a mutação que produz a variante, informação que pode ser relevante na classificação de novas entradas.

Os resultados com os Filtros 6, 7 e 8 (trechos de seqüências que mais se repetem), de certa forma, indicam trechos de seqüências que, à princípio, não seriam importantes na classificação da proteína e, conseqüentemente, na determinação da função da mesma. Assim, a metodologia que produziu tais Filtros, poderia ser um processo alternativo para se identificar as regiões da proteína mais relevantes (não identificadas pelo Filtro). Para uma avaliação mais adequada dos Filtros propostos para a classificação de variantes de hemoglobina quanto para identificação das regiões mais relevantes seria apropriado a síntese de novas variantes e a verificação da ocorrência ou não de patologia em laboratório. Por fim, deveria-se comparar os resultados obtidos experimentalmente com os resultados produzidos pelos classificadores. As dificuldades experimentais e o alto custo desses processos inviabilizam atualmente a realização de tais testes em nossos laboratórios.

A partir das ADs produzidas utilizando-se os Filtros 6, 7 e 8, pode-se extrair a lógica que permite definir, com probabilidade de acerto relativamente alta, se uma variante será patogênica ou não. Assim, a continuidade natural da pesquisa realizada seria a análise das ADs obtidas do ponto de vista bioquímico, buscando extrair possíveis regras capazes de determinar a funcionalidade de uma proteína. Para realização desta proposta, é importante observar que o número de nós das ADs apresentadas no Capítulo 6 podem ser reduzidas, produzindo ADs mais simples e, portanto, de análise mais fácil. O algoritmo C4.5 produz, em vários casos, um nó Filho relativo a cada um dos vinte aminoácidos para obter a mesma conclusão. Esses nós podem, em geral, ser agrupados em um único nó, simplificando a AD original.

Uma outra pesquisa a ser considerada seria a comparação das regiões (posições) mais relevantes obtidas pelos Filtros com as posições principais identificadas por BASHFORD (1987) para as proteínas da família das globinas. Utilizando os resultados de BASHFORD (1987) não se consegue classificadores mais eficientes. Porém, com os Filtros 7 e 8, obtém-se resultados melhores. Assim, pode-se questionar: qual a

metodologia que realmente retoma as posições mais relevantes. Há uma intersecção entre as informações produzidas nessas metodologias?

Tais investigações poderiam auxiliar os estudos da funcionalidade de proteína com base na seqüência primária, de forma, a produzir algoritmos capazes de avaliar uma proteína a partir da seqüência de aminoácidos. Com base nestes algoritmos, poderia-se construir um sistema evolutivo *in silício* para auxílio de técnicas de evolução *in vitro*.

REFERÊNCIA BIBLIOGRÁFICA

AMABIS, J. C. M; MARTHO, G. R. (1997). **Fundamentos da Biologia Moderna**. 2º ed. São Paulo: Editora Moderna.

BARISIC, I.; WILHELM, V.; STAMBUK, N.; KARAMAN, K. (2002). Machine Learning Based Analysis of Biochemical and Morphologic Parameters in Patients with Dialysis Related Amyloidosis. **Croatica Chemica Acta**, v.75, n.04, p. 935-944.

BASHFORD, D.; CHOTHIA, C.; LESK, M. A. (1987). Determinants of a Protein Fold-Unique Features of Globin Amino Acid Sequences. **Journal of Molecular Biology**, v.196, p. 199-216.

BOWIE, J. V.; LÜTHY, R.; EISENBER, D. (1991). A Method to identify protein sequences that fold into a know three-dimensional structure. **Science**, v.253, p.167-170.

BRATKO, I. (1990). Prolog Programing for Artificial Inteligence. Addison-Wesley.

CARVALHO, A. C. P. L. F.; BRAGA, A. P.; LUDERMIR, T. B. (2003). Computação Evolutiva. In: REZENDE, S. O. **Sistemas Inteligentes-Fundamentos e Aplicações**. Barueri: Editora Manole, p.525.

CARVALHO, A. C. P. L. F. (2003). Computação Bioinspirada. n.22, out.,nov.,dez. Disponível: http://www.cdcc.sc.usp.br/ciência/artigos/art_22/computaçãobioinspirada
Acesso em: 10 set. 2004.

CAMPBELL, M. K. (2000). **Bioquímica**. 3º ed. Porto Alegre: Editora Artes Médicas Sul.

CONN, E. E.; STUMPF, P. K. (1990). **Introdução a Bioquímica**. 4º ed. São Paulo: Editora Edgard Blucher Ltda.

DELBONI, L. F. (1991). Cristalografia estrutural: Estudos da hemoglobina do peixe *leporinus frederici* e determinação de estruturas de pequenas moléculas por difração de raio x. Dissertação de mestrado, IFSC. USP.

DICKERSON, R. E.; GEIS, I. (1983). **Hemoglobin: Structure, function, evolution, and pathology**. Menlo Park: Editora Benjamin/Cummings publishing.

DOSE, K. (1982) **Bioquímica**. 2º ed. Editora USP: Springer.

FITZWATER, T. ; POLISKY, B. (1996). **A SELEX primer**. Methods in Enzymology, v. 267, p. 275-301.

GARDNER, E. J.; SNUSTAD, D. P. (1987). **Genética**. 7ª ed. Rio de Janeiro: Editora Guanabara Koogan.

GIVER, L.; ARNOLD, F. H. (1998). Combinatorial protein design by *in vitro* recombination. **Current Opinion in Chemical Biology**, v.2, p.335-338.

GOLD, L. (1995). **Oligonucleotides as research, diagnostic, and therapeutic agents**. J. Biol. Chem, v. 270, p. 13581-13584.

GOLDBERG, D. E. (1989). Genetic Algorithm in Search, Optimization, and Machine Learning. Addison-Wesley Publishing Company INC.

HAYKIN, S. (1999). **Support Vector Machines**. Em Neural Networks; A Comprehensive Foundation. Capítulo 6. Prentice Hall.

ICB, Instituto de Ciências Biológicas. Universidade Federal de Minas Gerais. Disponível em <http://www.icb.ufmg.br/~lbcd/grupo1/pag1.html>. Acesso em agosto de 2003.

KLUG, S. J.; FAMULOK, M. (1994). All you wanted to know about SELEX. **Molecular Biology Reports**, v.20, p.97-107.

LEHNINGER, A. L. (1976). **Bioquímica**. 2º ed. v.1. São Paulo: Edgard Blucher Ltda.

LEVINE, R.I. (1988). Inteligencia Artificial e Sistemas Especialistas. São Paulo: McGraw-hill.

MARZZOCO, A.; TORRES, B. B. (1999). **Bioquímica Básica**. 2º ed. Rio de Janeiro: Guanabara Koogan S.A.

MADDOURI, M.; ELLOUMI, M. (2002). A Data mining approach based on Machine Learning techniques to classify biological sequences. **Knowledge-Based Systems**, v.15, p. 217-223.

MITCHELL, T. M. (1997). Machine Learning. McGraw-Hill.

MONARD, M. C.; BARANAUSKA, J. A. (2003). Indução de Regras e Árvores de Decisão. In: REZENDE, S. O. **Sistemas Inteligentes-Fundamentos e Aplicações**. Barueri: Editora Manole, p.525.

MOORE, L.G.; MARANAS, D. C.; LUTZ, S.; BENKOVIC, S.J. (2001). Predicting Crossover Generation in DNA Shuffling, v.98, nº 6, March.

OLIVEIRA, L. F.; PADILHA, P. P. T.; PREVIERO, A C.; MELO, G. C. A (2002). Utilização de Algoritmos Simbólicos para a Identificação do Número de Carótiolos do Fruto Pequi. Encoinfo.

QUINLAN, R. (1993). C4.5: Programs for machine learning. Morgan Kaufmann.

QUINLAN, J.R. (1986). "Induction of Decision Trees" . In: Machine Learning.

RICH, E. (1993). **Inteligência Artificial**. 2º ed. São Paulo: Makron Books.

SELBIG, J.; KADEN, F.; KOCH, I. (1992). Applying Machine Learning Methods for significant amino acid properties in proteins. **Federation of European Biochemical Societies**, v.297, n. 03, p. 241-246.

SIMON, H. A. (1983). “Why should machines learn?”. In: Machine learning, An Artificial Intelligence Approach. 1º ed. Palo Alto: Tioga Press.

SOARES, J. L. (1999). **Fundamentos de Biologia**. 1º ed. São Paulo: Editora Scipione.

SOUTO, M.C. P.; LORENA, A. C.; Delbem, A C.B.; Carvalho, A C. P. L. F. (2003). “Técnicas de Aprendizado de Máquina para Problemas de Biologia Molecular”.

STEMMER, W. P. C. (1994). DNA Shuffling by random fragmentation and reassembly: *in vitro* recombination for molecular evolution. v.91, p. 10747-10751.

STRYER, L. (1988). **Bioquímica**. 3º ed. Rio de Janeiro: Editora Guanabara Koogan.

SUN, F. (1998). Proceedings of the second annual international conference on Computational molecular biology. Annual Conference on Research in Computational Molecular Biology. **Modeling DNA Shuffling**. New York. p.251-257. Disponível em <<http://portal.acm.org/citation.cfm>. Acesso em agosto de 2003.

TSUNODA, F. D.; LOPES, H.S. (2003). Enzclass- Classificador Baseado em Algoritmo Memético e Árvore de Decisão para a descoberta e Seleção de Sequências de Aminoácidos em Enzimas. **III Workshop de Informática aplicada à Saúde-CBComp**. Itajaí- SC.

UNESP, Universidade Estadual Paulista. Disponível em: [http:// www.unesp.br/propp/dir_proj/Saude/saude46b.htm](http://www.unesp.br/propp/dir_proj/Saude/saude46b.htm), <http://cbe.ivic.ve/LBEfolleto.html>> Acesso em maio de 2004.

VAPNIK, V. (1998). Statistical Learning Theory. John Wiley & Sons.

VIANA, E. (1994). Modelo Molecular da Cadeia d da Hemoglobina de *Lumbricus terrestris*. Dissertação de mestrado, IQSC, USP.

VOET, D. (2000). **Fundamentos de Bioquímica**. 1º ed. Porto Alegre: Editora Artes Médicas sul.

WERHLI, A. V.; LEMKE, N. (2003). Anais do XXIII Congresso da Sociedade Brasileira de Computação. IV Encontro Nacional de Inteligência Artificial (ENIA).v. VII, **Um novo operador evolutivo para a determinação da estrutura tridimensional de proteínas**. Campinas: SBC, p.287-295.

WITTEN, H. I.; FRANK, E. (2000). Data Mining: Practical machine learning tools with Java implementations, Morgan Kaufmann, San Francisco.

ZHAO, H.; ARNOLD, F. H. (1997). Optimization of DNA shuffling for high fidelity recombination. **Nucleic Acids Research**, v.25, n.06, p.1307-1308.

ZHANG, J., DAWES, G.; STEMMER, W. P. C. (1997). Directed evolution of a fucosidase from a galactosidase by DNA shuffling and screening. **Proc. Atl. Acad. Sci. USA**. v.94, p. 4504-4509.

APÊNDICE A – Evolução *in Vitro*

A evolução *in Vitro* é um método laboratorial para a evolução de moléculas com propriedades desejadas. Esse método é comumente conhecido como seleção *in vitro*, evolução *in vitro* ou Selex (Evolução Sistemática de Ligações Exponenciais). Esse método tem sido aplicado com grande sucesso em uma ampla cadeia de estudos biológicos, tais como a interação entre DNA e Proteína, propriedades catalíticas de moléculas de RNA, e propriedades catalíticas de moléculas individuais de DNA (FITZWATER E POLISKY, 1996) e (GOLD, 1995).

O princípio básico do experimento da evolução *in vitro* pode ser sumarizado como segue abaixo:

Primeiro uma biblioteca de moléculas ao acaso de DNA, RNA ou Proteínas é construída. A biblioteca de moléculas pode ser composta de moléculas completamente criadas ao acaso de peptídeos ou de oligonucleotídeos. Ela também pode ser composta de muitas variantes de uma ou mais moléculas de uma matriz, obtida através de mutagênese.

Uma vez que uma biblioteca de moléculas é construída, algumas moléculas desta biblioteca podem ter uma função específica de interesse (SUN, 1998).

Uma seleção é feita para isolar essas moléculas. Moléculas funcionais são selecionadas principalmente de moléculas não funcionais de RNA ou de DNA, por colunas cromatográficas ou por outras técnicas de seleção adequadas para a propriedade que for desejada (KLUG, 1994). Para melhorar a diversidade das moléculas a serem exploradas, as moléculas que foram selecionadas poderão ser colocadas em um processo de mutagênese. As moléculas geradas através da mutagênese são então ampliadas pelo PCR ou por outros métodos de ampliação (SUN, 1998).

Os processos de Seleção, Mutagênese e Ampliação formam um ciclo do experimento. O experimento é repetido por múltiplos ciclos até que moléculas com propriedades desejadas forem obtidas. Ciclos repetidos de mutagêneses pontuais, recombinações e seleções podem permitir a evolução molecular *in vitro*, de seqüências complexas tais como as proteínas (STEMMER, 1998).

Um número de diferentes estratégias de mutagênese existe, tais como a mutagênese pontual pelo PCR, a mutagênese de oligonucleotídeos ou o uso de mutadores forçados como o DNA Shuffling (ZHANG et. al, 1997).

A seleção *in vitro* permite o isolamento de moléculas de ácidos nucleicos funcionais, alteração de ribossomos, síntese e avaliação de receptores originais baseados em ácidos nucleicos. Todas essas propriedades podem ser atingidas sem qualquer conhecimento da estrutura dessas moléculas.

A alta complexidade das informações usada na seleção *in vitro* faz necessária a ampliação de seqüências funcionais. As recombinações são particularmente úteis quando as seqüências geradas são viáveis. Os custos de tais seqüências devem ser, entretanto ponderado contra os custos da evolução pela mutagênese ao acaso (SUN, 1998).

DNA Shuffling

Métodos de mutagênese são continuamente desenvolvidos. O método de mutagênese mais recentemente desenvolvido é o do DNA Shuffling, também chamado de PCR sexual. Foi desenvolvido por STEMMER em 1994. O DNA Shuffling, ao longo de suas variações é a mais nova recombinação de DNA usada. Tem sido aplicado com muito sucesso na melhoria de medicamentos (SMITH, 1994), (STEMMER, 1995), (ZHANG et al, 1997), (CRAMERI et al, 1997) e (PATTEN et al, 1997); para otimizar enzimas industriais, (ZHAO E ARNOLD, 1996), (ARNOLD E MOORE, 1997) e (KUCHNER E ARNOLD, 1997); para ajudar no desenvolvimento de vacinas e de fármacos (PATTERN et al, 1997); e para distinguir mutações funcionais das não funcionais (ZHAO E ARNOLD, 1997). Recentemente foi utilizado para recombinar uma família de moléculas de diversas espécies, (CRAMERI et al, 1997). Antes de STEMMER introduzir a técnica de DNA Shuffling, a diversidade genética para a evolução de proteínas direcionadas era gerada primariamente por mutagênese pontual ou por mutagênese combinatorial.

A vantagem do método é a possibilidade de recombinações simultâneas, gerando múltiplas permutações cromossômicas por seqüências reagrupadas (MOORE et.al, 2001).

O método do DNA Shuffling consiste de quatro etapas:

- Preparação de genes para serem “embaralhados”.
- Fragmentação com Dnase I (enzima deoxiribonuclease).
- Remontagem por termociclo na presença do Dna polimerase (enzima que sintetiza Dna).

- Ampliação das moléculas remontadas pelo método convencional PCR (ZHAO E ARNOLD, 1997).

A evolução de moléculas por mutagênese é o equivalente a um processo evolucionário assexuado. Mutações nocivas aparecem em conjunto com mutações benéficas, essas (nocivas) se tornam fixas e podem até mesmo limitar o potencial evolucionário (GIVER et. al, 1998).

APÊNDICE B – Estrutura dos Aminoácidos

Fonte: CAMPBELL (2000).

APÊNDICE C – Tabela de Aminoácidos

Propriedades e convenções associadas com os aminoácidos								
Aminoácidos	Abreviaturas	Mr	Valores de pka			pI	Hidropatia	Ocorrência Proteínas
			pK ₁ (-COOH)	pK ₂ (-NH ₃ ⁺)	pK _R (R group)			
Grupos R Alifáticos								
Glicina	Gly G	75	2.34	9.60		5.97	-0.4	7.2
Alanina	Ala A	89	2.34	9.69		6.01	1.8	7.8
Valina	Val V	117	2.32	9.62		5.97	4.2	6.6
Leucina	Leu L	131	2.36	9.60		5.98	3.8	9.1
Isoleucina	Ile I	131	2.36	9.68		6.02	4.5	5.3
Metionina	Met M	149	2.28	9.21		5.74	1.9	2.3
Grupos R Aromáticos								
Fenilalanina	Phe F	165	1.83	9.13		5.48	2.8	3.9
Tirosina	Tyr Y	181	2.20	9.11	10.07	5.66	-1.3	3.2
Triptofano	Trp W	204	2.38	9.39		5.89	-0.9	1.4
Grupos R Polares								
Serina	Ser S	105	2.21	9.15		5.68	-0.8	6.8
Prolina	Pro P	115	1.99	10.96		6.48	1.6	5.2
Treonina	Thr T	119	2.11	9.62		5.87	-0.7	5.9
Cisteína	Cys C	121	1.96	10.28	8.18	5.07	2.5	1.9
Asparagina	Asn N	132	2.02	8.80		5.41	-3.5	4.3
Glutamina	Gln Q	146	2.17	9.13		5.65	-3.5	4.2
Grupos R carregados +								
Lisina	Lys K	146	2.18	8.95	10.53	9.74	-3.9	5.9
Histidina	His H	155	1.82	9.17	6.00	7.59	-3.2	2.3
Arginina	Arg R	174	2.17	9.04	12.48	10.76	-4.5	5.1
Grupos R carregados -								
Ácido Aspártico	Asp D	133	1.88	9.60	3.65	2.77	-3.5	5.3
Ácido Glutâmico	Glu E	147	2.19	9.67	4.25	3.22	-3.5	6.3

Fonte: CAMPBELL (2000).

APÊNDICE D – Tabela de Ambientes Químicos.

		W	F	Y	L	I	V	M	A	G	P	C	T	S	Q	N	E	D	H	K	R
A	Alpha	1,11	1,28	0,27	1,3	1,11	0,74	1,26	-0,77	-2,22	-1,56	-0,43	-1,72	-2,43	-1,38	-1,76	-2,15	-2,48	-0,34	-1,37	-1,8
A	Beta	0,92	0,96	0,17	1,07	1,5	1,18	0,51	-1,05	-2,35	-0,77	-0,45	-1,27	-2,56	-2,03	-2,18	-1,59	-1,8	-2,26	-3,04	-1,52
A	Other	0,96	1,4	0,52	1,06	0,93	1	0,91	-0,54	-2,78	0,59	-0,59	-1,41	-2,99	-0,84	-2,61	-2,01	-2,63	-0,61	-2,78	-2,35
B	Alpha	1,01	0,87	0,86	0,71	0,55	0,41	1,02	-0,65	-2,04	-0,97	0,15	-0,67	-1,33	0,16	-0,48	-0,58	-0,8	0,82	-0,94	-0,11
B	Beta	0,83	1,32	1,3	0,36	1,07	0,71	0,49	-1,52	-2,22	-0,86	-0,72	-1,14	-0,82	-0,79	-0,26	-0,2	-2,08	-0,05	-0,83	-0,41
B	Other	1,62	1,04	1,14	0,77	0,81	0,66	1	-0,81	-1,71	-0,07	-0,62	-1,03	-1,23	-0,87	-0,56	-1,13	-1,97	0,54	-2,12	-0,44
C	Alpha	0,86	-0,22	0,5	0,16	-0,02	-0,29	0,87	-0,44	-1,09	-1,11	-1,38	-0,69	-1,01	0,16	-0,07	0,09	-0,43	0,61	0,56	1,1
C	Beta	0,07	0,37	1,09	0,14	0,26	0,16	-0,68	-1,08	-2,29	-0,01	-0,79	-0,1	-0,71	0,52	-0,33	-0,42	-0,76	0,8	0,35	0,84
C	Other	1,12	0,71	1,25	0,29	-0,54	-0,4	0,23	-0,87	-0,61	-0,11	-0,98	-0,48	-0,61	0,1	0,09	-0,46	-0,83	1,04	0,08	0,71
D	Alpha	-1,29	-0,85	-0,88	-0,3	-0,06	0,3	-0,42	0,76	-0,46	-0,41	0,95	0,39	0,47	-0,32	-0,58	-0,43	-0,28	-0,91	-0,5	-0,51
D	Beta	0,34	-0,61	-0,09	-0,81	0,09	0,44	-0,4	0,59	-0,22	-0,65	1,28	0,95	0,49	-2,38	-0,92	-0,68	-0,61	-0,53	-2,01	-0,89
D	Other	-1,25	-1,29	-1,4	-0,33	-0,28	-0,09	-0,9	0,49	-0,39	0,64	1,29	0,55	0,59	-0,57	-0,26	-0,59	0,34	-1,21	-0,72	-0,88
E	Alpha	-1,09	-1,35	-0,55	-0,46	-0,59	-0,62	-0,27	-0,02	-0,58	-0,25	-0,7	-0,13	-0,38	0,62	-0,02	0,62	0,29	0,17	0,66	0,56
E	Beta	-0,71	-0,56	-0,3	-1,33	-0,35	0,08	-0,76	-0,52	-0,87	-1,01	-0,87	0,79	0,49	0,1	0	0,41	-0,03	-0,49	0,55	0,19
E	Other	-0,42	-0,84	-0,43	-0,68	-0,94	-0,74	-0,83	-0,25	-0,42	0,44	-0,81	0,08	0,17	0,25	0,51	0,28	0,51	0,2	0,47	0,24
F	Alpha	-1,26	-1,81	-1,7	-1,37	-2,36	-1,25	-0,9	0,44	0,63	0,05	-0,17	-0,2	0,16	0,29	0,32	0,6	0,44	-0,06	0,07	-0,2
F	Beta	0,81	-0,83	-0,03	-1,6	-1,39	-1,66	-0,62	0,14	1,75	-0,88	-0,04	-0,17	0,65	-0,12	0,01	-0,37	-0,3	-0,76	-1,54	-1,12
F	Other	-2,06	-1,63	-1,04	-1,14	-1,63	-0,8	-1,3	0,14	1,1	0,25	-0,35	0,08	0,34	-0,03	0,41	0,04	0,23	-0,41	-0,1	-0,41

APÊNDICE E – Árvores de Decisão

Teste 13

Number of Leaves : 1

Size of the tree : 1

Time taken to build model: 0.05 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	114	82.0144 %
Incorrectly Classified Instances	25	17.9856 %
Kappa statistic	0	
Mean absolute error	0.2953	
Root mean squared error	0.3844	
Relative absolute error	98.9207 %	
Root relative squared error	99.9932 %	
Total Number of Instances	139	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
1	1	0.82	1	0.901	BOA
0	0	0	0	0	RUIM

=== Confusion Matrix ===

a	b	<-- classified as
114	0	a = BOA
25	0	b = RUIM

Teste 15

Number of Leaves : 1

Size of the tree : 1

Time taken to build model: 0.05 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	88	63.7681 %
Incorrectly Classified Instances	50	36.2319 %
Kappa statistic	0	
Mean absolute error	0.4621	
Root mean squared error	0.4807	
Relative absolute error	99.8702 %	
Root relative squared error	99.999 %	
Total Number of Instances	138	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
1	1	0.638	1	0.779	BOA
0	0	0	0		RUIM

=== Confusion Matrix ===

```

a b <-- classified as
88 0 | a = BOA
50 0 | b = RUIM

```

Teste 18

Number of Leaves : 1

Size of the tree : 1

Time taken to build model: 0 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	114	82.0144 %
Incorrectly Classified Instances	25	17.9856 %
Kappa statistic	0	
Mean absolute error	0.2953	
Root mean squared error	0.3844	
Relative absolute error	98.9207 %	
Root relative squared error	99.9932 %	
Total Number of Instances	139	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
1	1	0.82	1	0.901	BOA
0	0	0	0		RUIM

=== Confusion Matrix ===

```

a  b  <-- classified as
114  0 | a = BOA
25   0 | b = RUIM

```

Teste 19

Number of Leaves : 1

Size of the tree : 1

Time taken to build model: 0.11 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	182	73.3871 %
Incorrectly Classified Instances	66	26.6129 %
Kappa statistic	0	
Mean absolute error	0.3907	
Root mean squared error	0.442	
Relative absolute error	99.7521 %	
Root relative squared error	99.9991 %	
Total Number of Instances	248	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
1	1	0.734	1	0.847	BOA
0	0	0	0		RUIM

=== Confusion Matrix ===

```

a  b  <-- classified as
182  0 | a = BOA
 66  0 | b = RUIM

```

Teste 23

```

campo145 = *
| campo144 = *
| | campo140 = *: RUIM (20.0/1.0)
| | campo140 = A
| | | campo137 = *: BOA (52.0)
| | | campo137 = A: BOA (0.0)
| | | campo137 = R: BOA (0.0)
| | | campo137 = N: BOA (0.0)
| | | campo137 = D: BOA (0.0)
| | | campo137 = C: BOA (0.0)
| | | campo137 = E: BOA (0.0)
| | | campo137 = Q: BOA (0.0)
| | | campo137 = G: BOA (0.0)
| | | campo137 = H: BOA (0.0)
| | | campo137 = I: BOA (0.0)
| | | campo137 = L: BOA (0.0)
| | | campo137 = K: BOA (0.0)
| | | campo137 = M: BOA (0.0)
| | | campo137 = F: BOA (0.0)
| | | campo137 = P: BOA (0.0)
| | | campo137 = S: BOA (0.0)
| | | campo137 = T: BOA (0.0)
| | | campo137 = W: BOA (0.0)
| | | campo137 = Y: BOA (0.0)
| | | campo137 = V
| | | | campo141 = *: RUIM (3.0)
| | | | campo141 = A: RUIM (0.0)
| | | | campo141 = R: RUIM (0.0)
| | | | campo141 = N: RUIM (0.0)
| | | | campo141 = D: RUIM (0.0)
| | | | campo141 = C: RUIM (0.0)
| | | | campo141 = E: RUIM (0.0)
| | | | campo141 = Q: RUIM (0.0)
| | | | campo141 = G: RUIM (0.0)
| | | | campo141 = H: RUIM (0.0)
| | | | campo141 = I: RUIM (0.0)
| | | | campo141 = L
| | | | | campo142 = *: BOA (2.0)
| | | | | campo142 = A: RUIM (9.0/2.0)
| | | | | campo142 = R: RUIM (0.0)
| | | | | campo142 = N: RUIM (0.0)
| | | | | campo142 = D: RUIM (0.0)
| | | | | campo142 = C: RUIM (0.0)
| | | | | campo142 = E: RUIM (0.0)
| | | | | campo142 = Q: RUIM (0.0)
| | | | | campo142 = G: RUIM (0.0)
| | | | | campo142 = H: RUIM (0.0)
| | | | | campo142 = I: RUIM (0.0)

```

				campo142 = L: RUIM (0.0)
				campo142 = K: RUIM (0.0)
				campo142 = M: RUIM (0.0)
				campo142 = F: RUIM (0.0)
				campo142 = P: RUIM (0.0)
				campo142 = S: RUIM (0.0)
				campo142 = T: RUIM (0.0)
				campo142 = W: RUIM (0.0)
				campo142 = Y: RUIM (0.0)
				campo142 = V: RUIM (0.0)
				campo141 = K: RUIM (0.0)
				campo141 = M: RUIM (0.0)
				campo141 = F: RUIM (0.0)
				campo141 = P: RUIM (0.0)
				campo141 = S: RUIM (0.0)
				campo141 = T: RUIM (0.0)
				campo141 = W: RUIM (0.0)
				campo141 = Y: RUIM (0.0)
				campo141 = V: RUIM (0.0)
				campo140 = R: BOA (0.0)
				campo140 = N: BOA (0.0)
				campo140 = D: BOA (0.0)
				campo140 = C: BOA (0.0)
				campo140 = E: BOA (0.0)
				campo140 = Q: BOA (0.0)
				campo140 = G: BOA (0.0)
				campo140 = H: BOA (0.0)
				campo140 = I: BOA (0.0)
				campo140 = L: BOA (0.0)
				campo140 = K: BOA (0.0)
				campo140 = M: BOA (0.0)
				campo140 = F: BOA (0.0)
				campo140 = P: BOA (0.0)
				campo140 = S: BOA (0.0)
				campo140 = T: BOA (0.0)
				campo140 = W: BOA (0.0)
				campo140 = Y: BOA (0.0)
				campo140 = V: BOA (0.0)
				campo144 = A: RUIM (0.0)
				campo144 = R: RUIM (0.0)
				campo144 = N: RUIM (0.0)
				campo144 = D: RUIM (0.0)
				campo144 = C: RUIM (0.0)
				campo144 = E: RUIM (0.0)
				campo144 = Q: RUIM (0.0)
				campo144 = G: RUIM (0.0)
				campo144 = H: RUIM (0.0)
				campo144 = I: RUIM (0.0)
				campo144 = L: RUIM (0.0)
				campo144 = K: RUIM (157.0)

```

| campo144 = M: RUIM (0.0)
| campo144 = F: RUIM (0.0)
| campo144 = P: RUIM (0.0)
| campo144 = S: RUIM (0.0)
| campo144 = T: RUIM (0.0)
| campo144 = W: RUIM (0.0)
| campo144 = Y: RUIM (0.0)
| campo144 = V: RUIM (0.0)
campo145 = A: BOA (0.0)
campo145 = R: BOA (0.0)
campo145 = N: BOA (0.0)
campo145 = D: BOA (0.0)
campo145 = C: BOA (0.0)
campo145 = E: BOA (0.0)
campo145 = Q: BOA (0.0)
campo145 = G: BOA (0.0)
campo145 = H: BOA (0.0)
campo145 = I: BOA (0.0)
campo145 = L: BOA (0.0)
campo145 = K: BOA (0.0)
campo145 = M: BOA (0.0)
campo145 = F: BOA (0.0)
campo145 = P: BOA (0.0)
campo145 = S: BOA (0.0)
campo145 = T: BOA (0.0)
campo145 = W: BOA (0.0)
campo145 = Y: BOA (140.0)
campo145 = V: BOA (0.0)

```

Number of Leaves : 121

Size of the tree : 127

Time taken to build model: 0.33 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	376	98.1723 %
Incorrectly Classified Instances	7	1.8277 %
Kappa statistic	0.9634	
Mean absolute error	0.0241	
Root mean squared error	0.1201	
Relative absolute error	4.8249 %	
Root relative squared error	24.02 %	
Total Number of Instances	383	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.975	0.011	0.99	0.975	0.982	BOA
0.989	0.025	0.974	0.989	0.981	RUIM

=== Confusion Matrix ===

```
a  b  <-- classified as
192  5 | a = BOA
 2 184 | b = RUIM
```

Teste 25

```

campo128 = *: RUIM (15.0)
campo128 = A
| campo70 = *: RUIM (10.0)
| campo70 = A
| | campo77 = *: BOA (15.0)
| | campo77 = A: BOA (0.0)
| | campo77 = R: BOA (0.0)
| | campo77 = N: BOA (0.0)
| | campo77 = D: BOA (0.0)
| | campo77 = C: BOA (0.0)
| | campo77 = E: BOA (0.0)
| | campo77 = Q: BOA (0.0)
| | campo77 = G: BOA (0.0)
| | campo77 = H
| | | campo82 = *: RUIM (11.0)
| | | campo82 = A: BOA (0.0)
| | | campo82 = R: BOA (0.0)
| | | campo82 = N: BOA (0.0)
| | | campo82 = D: BOA (0.0)
| | | campo82 = C: BOA (0.0)
| | | campo82 = E: BOA (0.0)
| | | campo82 = Q: BOA (0.0)
| | | campo82 = G: BOA (0.0)
| | | campo82 = H: BOA (0.0)
| | | campo82 = I: BOA (0.0)
| | | campo82 = L: BOA (0.0)
| | | campo82 = K
| | | campo99 = *
| | | | campo95 = *: BOA (3.0/1.0)
| | | | campo95 = A: RUIM (0.0)
| | | | campo95 = R: RUIM (0.0)
| | | | campo95 = N: RUIM (0.0)
| | | | campo95 = D: RUIM (0.0)
| | | | campo95 = C: RUIM (0.0)
| | | | campo95 = E: RUIM (0.0)
| | | | campo95 = Q: RUIM (0.0)
| | | | campo95 = G: RUIM (0.0)
| | | | campo95 = H: RUIM (0.0)
| | | | campo95 = I: RUIM (0.0)
| | | | campo95 = L: RUIM (0.0)
| | | | campo95 = K: RUIM (21.0)
| | | | campo95 = M: RUIM (0.0)
| | | | campo95 = F: RUIM (0.0)
| | | | campo95 = P: RUIM (0.0)
| | | | campo95 = S: RUIM (0.0)
| | | | campo95 = T: RUIM (0.0)
| | | | campo95 = W: RUIM (0.0)
| | | | campo95 = Y: RUIM (0.0)

```


	campo27 = *: BOA (2.0)
	campo27 = A: RUIM (11.0)
	campo27 = R: RUIM (0.0)
	campo27 = N: RUIM (0.0)
	campo27 = D: RUIM (0.0)
	campo27 = C: RUIM (0.0)
	campo27 = E: RUIM (0.0)
	campo27 = Q: RUIM (0.0)
	campo27 = G: RUIM (0.0)
	campo27 = H: RUIM (0.0)
	campo27 = I: RUIM (0.0)
	campo27 = L: RUIM (0.0)
	campo27 = K: RUIM (0.0)
	campo27 = M: RUIM (0.0)
	campo27 = F: RUIM (0.0)
	campo27 = P: RUIM (0.0)
	campo27 = S: RUIM (0.0)
	campo27 = T: RUIM (0.0)
	campo27 = W: RUIM (0.0)
	campo27 = Y: RUIM (0.0)
	campo27 = V: RUIM (0.0)
	campo36 = A: BOA (0.0)
	campo36 = R: BOA (0.0)
	campo36 = N: BOA (0.0)
	campo36 = D: BOA (0.0)
	campo36 = C: BOA (0.0)
	campo36 = E: BOA (0.0)
	campo36 = Q: BOA (0.0)
	campo36 = G: BOA (0.0)
	campo36 = H: BOA (0.0)
	campo36 = I: BOA (0.0)
	campo36 = L: BOA (0.0)
	campo36 = K: BOA (0.0)
	campo36 = M: BOA (0.0)
	campo36 = F: BOA (0.0)
	campo36 = P
	campo108 = *
	campo104 = *: BOA (3.0)
	campo104 = A: RUIM (0.0)
	campo104 = R: RUIM (7.0)
	campo104 = N: RUIM (0.0)
	campo104 = D: RUIM (0.0)
	campo104 = C: RUIM (0.0)
	campo104 = E: RUIM (0.0)
	campo104 = Q: RUIM (0.0)
	campo104 = G: RUIM (0.0)
	campo104 = H: RUIM (0.0)
	campo104 = I: RUIM (0.0)
	campo104 = L: RUIM (0.0)
	campo104 = K: RUIM (0.0)

		campo99 = H: BOA (0.0)
		campo99 = I: BOA (0.0)
		campo99 = L: BOA (0.0)
		campo99 = K: BOA (0.0)
		campo99 = M: BOA (0.0)
		campo99 = F: BOA (0.0)
		campo99 = P: BOA (0.0)
		campo99 = S: BOA (0.0)
		campo99 = T: BOA (0.0)
		campo99 = W: BOA (0.0)
		campo99 = Y: BOA (0.0)
		campo99 = V: BOA (0.0)
		campo82 = M: BOA (0.0)
		campo82 = F: BOA (0.0)
		campo82 = P: BOA (0.0)
		campo82 = S: BOA (0.0)
		campo82 = T: BOA (0.0)
		campo82 = W: BOA (0.0)
		campo82 = Y: BOA (0.0)
		campo82 = V: BOA (0.0)
		campo77 = I: BOA (0.0)
		campo77 = L: BOA (0.0)
		campo77 = K: BOA (0.0)
		campo77 = M: BOA (0.0)
		campo77 = F: BOA (0.0)
		campo77 = P: BOA (0.0)
		campo77 = S: BOA (0.0)
		campo77 = T: BOA (0.0)
		campo77 = W: BOA (0.0)
		campo77 = Y: BOA (0.0)
		campo77 = V: BOA (0.0)
		campo70 = R: BOA (0.0)
		campo70 = N: BOA (0.0)
		campo70 = D: BOA (0.0)
		campo70 = C: BOA (0.0)
		campo70 = E: BOA (0.0)
		campo70 = Q: BOA (0.0)
		campo70 = G: BOA (0.0)
		campo70 = H: BOA (0.0)
		campo70 = I: BOA (0.0)
		campo70 = L: BOA (0.0)
		campo70 = K: BOA (0.0)
		campo70 = M: BOA (0.0)
		campo70 = F: BOA (0.0)
		campo70 = P: BOA (0.0)
		campo70 = S: BOA (0.0)
		campo70 = T: BOA (0.0)
		campo70 = W: BOA (0.0)
		campo70 = Y: BOA (0.0)
		campo70 = V: BOA (0.0)

campo128 = R: BOA (0.0)
 campo128 = N: BOA (0.0)
 campo128 = D: BOA (0.0)
 campo128 = C: BOA (0.0)
 campo128 = E: BOA (0.0)
 campo128 = Q: BOA (0.0)
 campo128 = G: BOA (0.0)
 campo128 = H: BOA (0.0)
 campo128 = I: BOA (0.0)
 campo128 = L: BOA (0.0)
 campo128 = K: BOA (0.0)
 campo128 = M: BOA (0.0)
 campo128 = F: BOA (0.0)
 campo128 = P: BOA (0.0)
 campo128 = S: BOA (0.0)
 campo128 = T: BOA (0.0)
 campo128 = W: BOA (0.0)
 campo128 = Y: BOA (0.0)
 campo128 = V: BOA (0.0)

Number of Leaves : 541

Size of the tree : 568

Time taken to build model: 4.23 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	353	92.1671 %
Incorrectly Classified Instances	30	7.8329 %
Kappa statistic	0.8431	
Mean absolute error	0.1017	
Root mean squared error	0.2568	
Relative absolute error	20.3594 %	
Root relative squared error	51.3795 %	
Total Number of Instances	383	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.939	0.097	0.911	0.939	0.925	BOA
0.903	0.061	0.933	0.903	0.918	RUIM

=== Confusion Matrix ===

a b <-- classified as
 185 12 | a = BOA
 18 168 | b = RUIM

Teste 26

```

campo51 = *: RUIM (13.0)
campo51 = A: BOA (0.0)
campo51 = R: BOA (1.0)
campo51 = N: BOA (0.0)
campo51 = D: BOA (0.0)
campo51 = C: BOA (0.0)
campo51 = E: BOA (0.0)
campo51 = Q: BOA (0.0)
campo51 = G: BOA (0.0)
campo51 = H: BOA (1.0)
campo51 = I: BOA (0.0)
campo51 = L: BOA (0.0)
campo51 = K: BOA (0.0)
campo51 = M: BOA (0.0)
campo51 = F: BOA (0.0)
campo51 = P
| campo19 = *: RUIM (9.0)
| campo19 = A: BOA (0.0)
| campo19 = R: BOA (0.0)
| campo19 = N
| | campo99 = *: RUIM (17.0/2.0)
| | campo99 = A: RUIM (1.0)
| | campo99 = R: BOA (0.0)
| | campo99 = N: RUIM (1.0)
| | campo99 = D
| | | campo108 = *: RUIM (14.0/1.0)
| | | campo108 = A: BOA (0.0)
| | | campo108 = R: BOA (0.0)
| | | campo108 = N
| | | | campo90 = *: RUIM (2.0)
| | | | campo90 = A: BOA (0.0)
| | | | campo90 = R: BOA (0.0)
| | | | campo90 = N: BOA (0.0)
| | | | campo90 = D: RUIM (1.0)
| | | | campo90 = C: BOA (0.0)
| | | | campo90 = E
| | | | | campo92 = *: RUIM (4.0)
| | | | | campo92 = A: BOA (0.0)
| | | | | campo92 = R: RUIM (1.0)
| | | | | campo92 = N: RUIM (1.0)
| | | | | campo92 = D: RUIM (1.0)
| | | | | campo92 = C: BOA (0.0)
| | | | | campo92 = E: BOA (0.0)
| | | | | campo92 = Q: RUIM (1.0)
| | | | | campo92 = G: BOA (0.0)
| | | | | campo92 = H
| | | | | | campo146 = *: RUIM (40.0/8.0)
| | | | | | campo146 = A: BOA (0.0)

```

					campo146 = R: BOA (1.0)
					campo146 = N: BOA (0.0)
					campo146 = D: RUIM (1.0)
					campo146 = C: BOA (0.0)
					campo146 = E: BOA (0.0)
					campo146 = Q: RUIM (1.0)
					campo146 = G: BOA (0.0)
					campo146 = H
					campo97 = *: RUIM (4.0/1.0)
					campo97 = A: BOA (0.0)
					campo97 = R: BOA (0.0)
					campo97 = N: BOA (1.0)
					campo97 = D: BOA (0.0)
					campo97 = C: BOA (0.0)
					campo97 = E: BOA (0.0)
					campo97 = Q: RUIM (1.0)
					campo97 = G: BOA (0.0)
					campo97 = H
					campo82 = *
					campo78 = *: BOA (3.0)
					campo78 = A: RUIM (0.0)
					campo78 = R: RUIM (0.0)
					campo78 = N: RUIM (0.0)
					campo78 = D: RUIM (0.0)
					campo78 = C: RUIM (0.0)
					campo78 = E: RUIM (0.0)
					campo78 = Q: RUIM (0.0)
					campo78 = G: RUIM (0.0)
					campo78 = H: RUIM (0.0)
					campo78 = I: RUIM (0.0)
					campo78 = L: RUIM (6.0)
					campo78 = K: RUIM (0.0)
					campo78 = M: RUIM (0.0)
					campo78 = F: RUIM (0.0)
					campo78 = P: RUIM (0.0)
					campo78 = S: RUIM (0.0)
					campo78 = T: RUIM (0.0)
					campo78 = W: RUIM (0.0)
					campo78 = Y: RUIM (0.0)
					campo78 = V: RUIM (0.0)
					campo82 = A: BOA (0.0)
					campo82 = R: BOA (0.0)
					campo82 = N: BOA (2.0/1.0)
					campo82 = D: BOA (0.0)
					campo82 = C: BOA (0.0)
					campo82 = E: RUIM (1.0)
					campo82 = Q: RUIM (1.0)
					campo82 = G: BOA (0.0)
					campo82 = H: BOA (0.0)
					campo82 = I: BOA (0.0)

		campo90 = T: BOA (0.0)
		campo90 = W: BOA (0.0)
		campo90 = Y: BOA (0.0)
		campo90 = V: BOA (0.0)
		campo108 = D: RUIM (1.0)
		campo108 = C: BOA (0.0)
		campo108 = E: BOA (0.0)
		campo108 = Q: BOA (0.0)
		campo108 = G: BOA (0.0)
		campo108 = H: BOA (1.0)
		campo108 = I: RUIM (1.0)
		campo108 = L: BOA (0.0)
		campo108 = K: BOA (1.0)
		campo108 = M: BOA (0.0)
		campo108 = F: BOA (0.0)
		campo108 = P: BOA (0.0)
		campo108 = S: RUIM (1.0)
		campo108 = T: BOA (0.0)
		campo108 = W: BOA (0.0)
		campo108 = Y: BOA (0.0)
		campo108 = V: BOA (0.0)
		campo99 = C: BOA (0.0)
		campo99 = E: RUIM (1.0)
		campo99 = Q: BOA (0.0)
		campo99 = G: RUIM (1.0)
		campo99 = H: RUIM (1.0)
		campo99 = I: BOA (0.0)
		campo99 = L: BOA (0.0)
		campo99 = K: BOA (0.0)
		campo99 = M: BOA (0.0)
		campo99 = F: BOA (0.0)
		campo99 = P: BOA (0.0)
		campo99 = S: BOA (0.0)
		campo99 = T: BOA (0.0)
		campo99 = W: BOA (0.0)
		campo99 = Y: RUIM (1.0)
		campo99 = V: RUIM (1.0)
		campo19 = D: BOA (1.0)
		campo19 = C: BOA (0.0)
		campo19 = E: BOA (0.0)
		campo19 = Q: BOA (0.0)
		campo19 = G: BOA (0.0)
		campo19 = H: BOA (0.0)
		campo19 = I: BOA (0.0)
		campo19 = L: BOA (0.0)
		campo19 = K: BOA (1.0)
		campo19 = M: BOA (0.0)
		campo19 = F: BOA (0.0)
		campo19 = P: BOA (0.0)
		campo19 = S: BOA (0.0)

```
| campo19 = T: BOA (0.0)
| campo19 = W: BOA (0.0)
| campo19 = Y: BOA (0.0)
| campo19 = V: BOA (0.0)
campo51 = S: BOA (0.0)
campo51 = T: BOA (0.0)
campo51 = W: BOA (0.0)
campo51 = Y: BOA (0.0)
campo51 = V: BOA (0.0)
```

Number of Leaves : 341

Size of the tree : 358

Time taken to build model: 2.36 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	301	78.5901 %
Incorrectly Classified Instances	82	21.4099 %
Kappa statistic	0.5683	
Mean absolute error	0.236	
Root mean squared error	0.3803	
Relative absolute error	47.2448 %	
Root relative squared error	76.0936 %	
Total Number of Instances	383	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.914	0.349	0.735	0.914	0.814	BOA
0.651	0.086	0.877	0.651	0.747	RUIM

=== Confusion Matrix ===

a b <-- classified as

180 17 | a = BOA

65 121 | b = RUIM

Teste 27

```

campo128 = *: RUIM (12.0)
campo128 = $: RUIM (3.0)
campo128 = A
| campo51 = *: RUIM (13.0)
| campo51 = $: BOA (2.0)
| campo51 = A: BOA (0.0)
| campo51 = R: BOA (0.0)
| campo51 = N: BOA (0.0)
| campo51 = D: BOA (0.0)
| campo51 = C: BOA (0.0)
| campo51 = E: BOA (0.0)
| campo51 = Q: BOA (0.0)
| campo51 = G: BOA (0.0)
| campo51 = H: BOA (0.0)
| campo51 = I: BOA (0.0)
| campo51 = L: BOA (0.0)
| campo51 = K: BOA (0.0)
| campo51 = M: BOA (0.0)
| campo51 = F: BOA (0.0)
| campo51 = P
| campo19 = *: RUIM (9.0)
| campo19 = $: BOA (2.0)
| campo19 = A: BOA (0.0)
| campo19 = R: BOA (0.0)
| campo19 = N
| campo99 = *
| | campo95 = *: BOA (3.0/1.0)
| | campo95 = $: RUIM (0.0)
| | campo95 = A: RUIM (0.0)
| | campo95 = R: RUIM (0.0)
| | campo95 = N: RUIM (0.0)
| | campo95 = D: RUIM (0.0)
| | campo95 = C: RUIM (0.0)
| | campo95 = E: RUIM (0.0)
| | campo95 = Q: RUIM (0.0)
| | campo95 = G: RUIM (0.0)
| | campo95 = H: RUIM (0.0)
| | campo95 = I: RUIM (0.0)
| | campo95 = L: RUIM (0.0)
| | campo95 = K: RUIM (14.0)
| | campo95 = M: RUIM (0.0)
| | campo95 = F: RUIM (0.0)
| | campo95 = P: RUIM (0.0)
| | campo95 = S: RUIM (0.0)
| | campo95 = T: RUIM (0.0)
| | campo95 = W: RUIM (0.0)
| | campo95 = Y: RUIM (0.0)
| | campo95 = V: RUIM (0.0)

```

			campo99 = \$: RUIM (7.0)
			campo99 = A: BOA (0.0)
			campo99 = R: BOA (0.0)
			campo99 = N: BOA (0.0)
			campo99 = D
			campo92 = *
			campo83 = *: BOA (2.0)
			campo83 = \$: RUIM (0.0)
			campo83 = A: RUIM (0.0)
			campo83 = R: RUIM (0.0)
			campo83 = N: RUIM (0.0)
			campo83 = D: RUIM (0.0)
			campo83 = C: RUIM (0.0)
			campo83 = E: RUIM (0.0)
			campo83 = Q: RUIM (0.0)
			campo83 = G: RUIM (4.0)
			campo83 = H: RUIM (0.0)
			campo83 = I: RUIM (0.0)
			campo83 = L: RUIM (0.0)
			campo83 = K: RUIM (0.0)
			campo83 = M: RUIM (0.0)
			campo83 = F: RUIM (0.0)
			campo83 = P: RUIM (0.0)
			campo83 = S: RUIM (0.0)
			campo83 = T: RUIM (0.0)
			campo83 = W: RUIM (0.0)
			campo83 = Y: RUIM (0.0)
			campo83 = V: RUIM (0.0)
			campo92 = \$: RUIM (6.0)
			campo92 = A: BOA (0.0)
			campo92 = R: BOA (0.0)
			campo92 = N: BOA (0.0)
			campo92 = D: BOA (0.0)
			campo92 = C: BOA (0.0)
			campo92 = E: BOA (0.0)
			campo92 = Q: BOA (0.0)
			campo92 = G: BOA (0.0)
			campo92 = H
			campo89 = *: RUIM (3.0)
			campo89 = \$: RUIM (3.0)
			campo89 = A: BOA (0.0)
			campo89 = R: BOA (0.0)
			campo89 = N: BOA (0.0)
			campo89 = D: BOA (0.0)
			campo89 = C: BOA (0.0)
			campo89 = E: BOA (0.0)
			campo89 = Q: BOA (0.0)
			campo89 = G: BOA (0.0)
			campo89 = H: BOA (0.0)
			campo89 = I: BOA (0.0)

					campo89 = L: BOA (0.0)
					campo89 = K: BOA (0.0)
					campo89 = M: BOA (0.0)
					campo89 = F: BOA (0.0)
					campo89 = P: BOA (0.0)
					campo89 = S
					campo146 = *
					campo137 = *
					campo133 = *
					campo135 = *: BOA (10.0/2.0)
					campo135 = \$: RUIM (2.0)
					campo135 = A: BOA (0.0)
					campo135 = R: BOA (0.0)
					campo135 = N: BOA (0.0)
					campo135 = D: BOA (0.0)
					campo135 = C: BOA (0.0)
					campo135 = E: BOA (0.0)
					campo135 = Q: BOA (0.0)
					campo135 = G: BOA (0.0)
					campo135 = H: BOA (0.0)
					campo135 = I: BOA (0.0)
					campo135 = L: BOA (0.0)
					campo135 = K: BOA (0.0)
					campo135 = M: BOA (0.0)
					campo135 = F: BOA (0.0)
					campo135 = P: BOA (0.0)
					campo135 = S: BOA (0.0)
					campo135 = T: BOA (0.0)
					campo135 = W: BOA (0.0)
					campo135 = Y: BOA (0.0)
					campo135 = V: BOA (0.0)
					campo133 = \$: RUIM (2.0)
					campo133 = A: BOA (0.0)
					campo133 = R: BOA (0.0)
					campo133 = N: BOA (0.0)
					campo133 = D: BOA (0.0)
					campo133 = C: BOA (0.0)
					campo133 = E: BOA (0.0)
					campo133 = Q: BOA (0.0)
					campo133 = G: BOA (0.0)
					campo133 = H: BOA (0.0)
					campo133 = I: BOA (0.0)
					campo133 = L: BOA (0.0)
					campo133 = K: BOA (0.0)
					campo133 = M: BOA (0.0)
					campo133 = F: BOA (0.0)
					campo133 = P: BOA (0.0)
					campo133 = S: BOA (0.0)
					campo133 = T: BOA (0.0)
					campo133 = W: BOA (0.0)

	campo82 = \$: RUIM (6.0/1.0)
	campo82 = A: BOA (0.0)
	campo82 = R: BOA (0.0)
	campo82 = N: BOA (0.0)
	campo82 = D: BOA (0.0)
	campo82 = C: BOA (0.0)
	campo82 = E: BOA (0.0)
	campo82 = Q: BOA (0.0)
	campo82 = G: BOA (0.0)
	campo82 = H: BOA (0.0)
	campo82 = I: BOA (0.0)
	campo82 = L: BOA (0.0)
	campo82 = K
	campo108 = *: RUIM (5.0/1.0)
	campo108 = \$
	campo104 = *: BOA (2.0)
	campo104 = \$: RUIM (0.0)
	campo104 = A: RUIM (0.0)
	campo104 = R: RUIM (3.0)
	campo104 = N: RUIM (0.0)
	campo104 = D: RUIM (0.0)
	campo104 = C: RUIM (0.0)
	campo104 = E: RUIM (0.0)
	campo104 = Q: RUIM (0.0)
	campo104 = G: RUIM (0.0)
	campo104 = H: RUIM (0.0)
	campo104 = I: RUIM (0.0)
	campo104 = L: RUIM (0.0)
	campo104 = K: RUIM (0.0)
	campo104 = M: RUIM (0.0)
	campo104 = F: RUIM (0.0)
	campo104 = P: RUIM (0.0)
	campo104 = S: RUIM (0.0)
	campo104 = T: RUIM (0.0)
	campo104 = W: RUIM (0.0)
	campo104 = Y: RUIM (0.0)
	campo104 = V: RUIM (0.0)
	campo108 = A: BOA (0.0)
	campo108 = R: BOA (0.0)
	campo108 = N
	campo36 = *
	campo27 = *: BOA (10.0)
	campo27 = \$: BOA (0.0)
	campo27 = A: RUIM (6.0)
	campo27 = R: BOA (0.0)
	campo27 = N: BOA (0.0)
	campo27 = D: BOA (0.0)
	campo27 = C: BOA (0.0)
	campo27 = E: BOA (0.0)
	campo27 = Q: BOA (0.0)

[illegible]

					campo146 = Y: BOA (0.0)
					campo146 = V: BOA (0.0)
					campo89 = T: BOA (0.0)
					campo89 = W: BOA (0.0)
					campo89 = Y: BOA (0.0)
					campo89 = V: BOA (0.0)
					campo92 = I: BOA (0.0)
					campo92 = L: BOA (0.0)
					campo92 = K: BOA (0.0)
					campo92 = M: BOA (0.0)
					campo92 = F: BOA (0.0)
					campo92 = P: BOA (0.0)
					campo92 = S: BOA (0.0)
					campo92 = T: BOA (0.0)
					campo92 = W: BOA (0.0)
					campo92 = Y: BOA (0.0)
					campo92 = V: BOA (0.0)
					campo99 = C: BOA (0.0)
					campo99 = E: BOA (0.0)
					campo99 = Q: BOA (0.0)
					campo99 = G: BOA (0.0)
					campo99 = H: BOA (0.0)
					campo99 = I: BOA (0.0)
					campo99 = L: BOA (0.0)
					campo99 = K: BOA (0.0)
					campo99 = M: BOA (0.0)
					campo99 = F: BOA (0.0)
					campo99 = P: BOA (0.0)
					campo99 = S: BOA (0.0)
					campo99 = T: BOA (0.0)
					campo99 = W: BOA (0.0)
					campo99 = Y: BOA (0.0)
					campo99 = V: BOA (0.0)
					campo19 = D: BOA (0.0)
					campo19 = C: BOA (0.0)
					campo19 = E: BOA (0.0)
					campo19 = Q: BOA (0.0)
					campo19 = G: BOA (0.0)
					campo19 = H: BOA (0.0)
					campo19 = I: BOA (0.0)
					campo19 = L: BOA (0.0)
					campo19 = K: BOA (0.0)
					campo19 = M: BOA (0.0)
					campo19 = F: BOA (0.0)
					campo19 = P: BOA (0.0)
					campo19 = S: BOA (0.0)
					campo19 = T: BOA (0.0)
					campo19 = W: BOA (0.0)
					campo19 = Y: BOA (0.0)
					campo19 = V: BOA (0.0)

```
| campo51 = S: BOA (0.0)
| campo51 = T: BOA (0.0)
| campo51 = W: BOA (0.0)
| campo51 = Y: BOA (0.0)
| campo51 = V: BOA (0.0)
campo128 = R: BOA (0.0)
campo128 = N: BOA (0.0)
campo128 = D: BOA (0.0)
campo128 = C: BOA (0.0)
campo128 = E: BOA (0.0)
campo128 = Q: BOA (0.0)
campo128 = G: BOA (0.0)
campo128 = H: BOA (0.0)
campo128 = I: BOA (0.0)
campo128 = L: BOA (0.0)
campo128 = K: BOA (0.0)
campo128 = M: BOA (0.0)
campo128 = F: BOA (0.0)
campo128 = P: BOA (0.0)
campo128 = S: BOA (0.0)
campo128 = T: BOA (0.0)
campo128 = W: BOA (0.0)
campo128 = Y: BOA (0.0)
campo128 = V: BOA (0.0)
```

Number of Leaves : 547

Size of the tree : 573

Time taken to build model: 2.31 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	353	92.1671 %
Incorrectly Classified Instances	30	7.8329 %
Kappa statistic	0.8432	
Mean absolute error	0.1018	
Root mean squared error	0.2583	
Relative absolute error	20.3718 %	
Root relative squared error	51.6775 %	
Total Number of Instances	383	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.924	0.081	0.924	0.924	0.924	BOA
0.919	0.076	0.919	0.919	0.919	RUIM

=== Confusion Matrix ===

a	b	<--	classified as
182	15		a = BOA
15	171		b = RUIM

APÊNDICE F- Nomenclatura de Árvores

Árvores de Decisão

Mineração de dados é o processo de extração de informações previamente desconhecidas, a partir de grandes bases de dados. Dentro da mineração de dados, o método de classificação é um dos mais utilizados. Ele busca determinar através de algoritmos, a que conjunto de exemplos, um elemento na base de dados pertence, a partir de conjuntos (classes) pré-definidos. Por exemplo: um banco utilizando técnicas para selecionar, entre seus clientes, aqueles com menor risco de não efetuar o pagamento de um empréstimo. A partir desta seleção, oferece linhas de crédito para os correntistas cujos filhos tem entre 18 e 21 anos e, portanto, precisem de dinheiro para ajudar os filhos a comprar o próprio carro ou arcar com os custos da faculdade.

Amplamente utilizadas em algoritmos de classificação, as árvores de decisão são representações simples do conhecimento e, um meio eficiente de construir classificadores que predizem classes baseadas nos valores de atributos de um conjunto de dados. As árvores de decisão consistem de nós que representam os atributos, de arcos, provenientes destes nós e que recebem os valores possíveis para estes atributos, e de folhas, que representam as diferentes classes de um conjunto de treinamento. Um nó folha representa uma única classe, mas uma classe pode estar representada em mais de um nó folhas. Um nó interno é chamado de nó-decisão, pois representa um teste sobre o valor de um atributo. Cada aresta que sai de um nó-decisão até um de seus nós filhos representa um dos possíveis resultados do teste sobre o valor do atributo. Podemos considerar o conjunto de todos os objetos (n atributos) possíveis como pontos em um espaço n -dimensional com um eixo para cada atributo, enumerando todos os valores possíveis para o mesmo. Assim, nota-se que cada nó decisão particiona o espaço de objetos em k partições, onde k é o número de arestas que partem do nó. A interseção entre os particionamentos efetuados por cada nó-decisão da árvore resulta em um espaço totalmente dividido em partições ainda menores, para os quais é atribuído uma única classe. Dada esta analogia, podemos verificar que um nó da árvore representa um subespaço. O nó-raiz representa o próprio espaço de objetos. Os nós-filho de um nó representam as partições da partição do espaço representada pelo nó-pai. Os nós-folha

representam partições em que, pelo menos teoricamente, só estão contidos elementos de uma mesma classe.

Uma árvore de decisão tem a função de particionar recursivamente um conjunto de treinamento, até que cada subconjunto obtido deste particionamento contenha casos de uma única classe. Para atingir esta meta, a técnica de árvores de decisão examina e compara a distribuição de classes durante a construção da árvore. Os resultados obtidos, após a construção de uma árvore de decisão, são dados organizados de maneira compacta, que são utilizados para classificar novos casos.

A Figura 1 apresenta um exemplo de árvore de decisão. Neste exemplo, são trabalhados objetos que relatam as condições propícias de uma pessoa receber ou não um empréstimo. É considerada a probabilidade do montante do empréstimo ser médio, baixo ou alto. Alguns objetos são exemplos positivos de uma classe *sim*, ou seja, os requisitos exigidos a uma pessoa, por um banco, são satisfatórios à concessão de um empréstimo, e outros são negativos, onde os requisitos exigidos não são satisfatórios à concessão de um empréstimo. Classificação, neste caso, é a construção de uma estrutura de árvore, que pode ser usada para classificar corretamente todos os objetos do conjunto.

Figura 1. Exemplo de uma árvore de decisão.

Após a construção de uma árvore de decisão é importante avaliá-la. Esta avaliação é realizada através da utilização de dados que não tenham sido usados no treinamento. Esta estratégia permite estimar como a árvore generaliza os dados e se adapta a novas situações, podendo, também, se estimar a proporção de erros e acertos ocorridos na construção da árvore.

A partir de uma árvore de decisão é possível derivar regras. As regras são escritas considerando o trajeto do nó raiz até uma folha da árvore. Estes dois métodos são geralmente utilizados em conjunto. Devido ao fato das árvores de decisão tenderem a crescer muito, de acordo com algumas aplicações, elas são muitas vezes substituídas pelas regras. Isto acontece em virtude das regras poderem ser facilmente modularizadas. Uma regra pode ser compreendida sem que haja a necessidade de se referenciar outras regras.

Com base na árvore de decisão apresentada na Figura 1, pode-se exemplificar a derivação de regras. Dois exemplos de regras obtidas a partir desta árvore são mostrados a seguir:

- Se montante = médio e salário = baixo
então classe = não
- Se montante = médio e salário = alto
então classe = sim