Instituto de Ciências Matemáticas e de Computação

ISSN - 0103-2569

Utilização de Técnicas Inteligentes em Bioinformática

Ana Carolina Lorena
André C. P. L. F. de Carvalho

Nº 219

RELATÓRIOS TÉCNICOS DO ICMC

São Carlos
Novembro/2003

Utilização de Técnicas Inteligentes em Bioinformática*

Ana Carolina Lorena
André C. P. L. F. de Carvalho

Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Ciências de Computação e Estatística
Laboratório de Inteligência Computacional
Caixa Postal 668, 13560-970 - São Carlos, SP, Brasil
e-mail: {aclorena, andre}@icmc.sc.usp.br

Resumo. A Bioinformática, área de pesquisa que emprega técnicas computacionais na análise e gerenciamento de dados biológicos, tem apresentado uma forte expansão nos últimos anos. Isto se deve em grande parte à necessidade de análise do dos grandes volumes de dados gerados em laboratórios de Biologia Molecular atualmente. Técnicas inteligentes pertencentes ao campo de Aprendizado de Máquina (AM) estão entre as mais empregadas nessas análises. Este campo da Inteligência Computacional estuda o desenvolvimento de algoritmos capazes de extrair descrições de conceitos a partir de exemplos (Mitchell, 1997), permitindo assim a extração de descrições automáticas de conceitos biológicos a partir de amostras de dados. Este relatório apresenta um levantamento bibliográfico sobre a utilização de técnicas inteligentes em alguns dos principais problemas tratados em Bioinformática.

Palavras-Chave: Bioinformática, Aprendizado de Máquina supervisionado.

Novembro 2003

*Trabalho realizado com auxílio da FAPESP.

Este documento foi preparado com o formatador de textos LaTeX. O sistema de citações de referências bibliográficas utiliza o padrão *Apalike* do sistema bibTeX.

Sumário

1 Introdução **1**

2 Conceitos de Biologia Molecular **2**
2.1 DNA, Genes e Proteínas 2
2.2 Expressão Gênica 5
2.3 Técnicas para Medida de Expressão Gênica 7

3 Algumas Aplicações em Bioinformática **8**

4 Aprendizado de Máquina **10**
4.1 Conceitos Básicos 10
4.2 Algumas Técnicas de AM 11
4.2.1 k-Vizinhos Mais Próximos 12
4.2.2 Raciocínio Baseado em Casos 14
4.2.3 Árvores de Decisão 15
4.2.4 Redes Neurais Artificiais 17
4.2.5 Algoritmos Genéticos 19
4.2.6 *Support Vector Machines* 21

5 Aplicações de AM em Bioinformática **24**
5.1 k-Vizinhos Mais Próximos 24
5.2 Raciocínio Baseado em Casos 25
5.3 Árvores de Decisão 25
5.4 Redes Neurais Artificiais 26
5.5 Algoritmos Genéticos 28
5.6 *Support Vector Machines* 29
5.7 Outras Técnicas 31

6 Conclusão **31**

Referências Bibliográficas **32**

A Glossário **40**

Lista de Figuras

1 Célula, cromossomo, DNA, gene e suas relações 3
2 Estrutura de uma molécula de DNA (Souto et al., 2003) 3
3 Conformações estruturais das proteínas; (a) estrutura primária; (b) estrutura secundária; (c) estrutura terciária; (d) estrutura quaternária 4
4 O processo de expressão gênica 5
5 *Splicing* em molécula de mRNA 6
6 Ilustração do processamento realizado pelo algoritmo k-NN (Mitchell, 1997) . . 12
7 Ciclo do RBC (Aamot and Plaza, 1994) 14
8 AD simplificada para o diagnóstico de pacientes (Baranauskas and Monard, 2000) 15
9 Representação da AD da Figura 8 por meio de regras de decisão (Baranauskas and Monard, 2000) 16
10 Exemplo de estrutura de uma RN 17
11 Operadores genéticos de cruzamento (a) e mutação (b) 19
12 Exemplo simplificado de classificação de promotores por uma SVM linear . 21
13 Transformação do conjunto de dados no espaço de entrada para o espaço de características. 23
14 Ilustração de uma AD para a identificação de regiões doadoras (Craven and Shavlik, 1994) 26
15 Parte de RN Perceptron utilizada no reconhecimento de sítios de início de tradução no DNA de bactérias *E. coli* (Craven and Shavlik, 1994) 27
16 Conjunto de seqüências similares 29

Lista de Tabelas

1 Código Genético (Lewis, 2001) 6

Lista de Algoritmos

1 Algoritmo ID3 (Mitchell, 1997) 16
2 Algoritmo *back-propagation* (Haykin, 1999) 18
3 Algoritmo Genético (Goldberg, 1989) 20

1 Introdução

A Bioinformática é uma área de pesquisa que vem despertando crescente interesse na comunidade científica. Este novo campo interdisciplinar consiste na aplicação de técnicas matemáticas, estatísticas e computacionais para a resolução de problemas provenientes da Biologia (Setubal and Meidanis, 1997). Dentre as diversas áreas da Biologia, aquela em que a aplicação de técnicas computacionais tem se mostrado mais frutífera é a Biologia Molecular, principalmente sua parte relacionada à genética.

Os estudos em genética possibilitam ganhos em várias áreas, entre elas Medicina, Farmácia e Agricultura. Pode-se, por exemplo, obter uma melhor compreensão da origem de diversas doenças, o que permite a formulação de tratamentos médicos mais eficazes, assim como o desenvolvimento de novas drogas. Como muitos alimentos atualmente são provenientes de organismos geneticamente modificados, as investigações em Bioinformática também permitem um maior controle e melhoria no desenvolvimento de novos organismos.

Atualmente, as pesquisas em Bioinformática estão se concentrando na análise dos grandes volumes de dados gerados em laboratórios, visando tratar problemas como, por exemplo, a identificação de genes[1] em seqüências genômicas (Craven and Shavlik, 1994).

Com o objetivo de obter procedimentos automáticos para tal, recursos e técnicas computacionais vêm sendo largamente empregados na análise de dados biológicos. Técnicas inteligentes pertencentes ao campo de Aprendizado de Máquina (AM) estão entre as mais utilizadas. Este campo da Inteligência Computacional estuda o desenvolvimento de algoritmos capazes de extrair descrições de conceitos a partir de exemplos (Mitchell, 1997), permitindo assim a extração de descrições automáticas de conceitos biológicos a partir de amostras de dados.

Este relatório tem por objetivo apresentar um levantamento bibliográfico sobre o uso de técnicas inteligentes em problemas da área de Bioinformática. São descritas diversas técnicas empregadas, assim como aplicações destas reportadas na literatura.

Iniciando este relatório, na Seção 2 apresenta-se uma introdução a conceitos relacionados à Biologia Molecular, necessários à compreensão das aplicações investigadas em Bioinformática. Na Seção 3 a Bioinformática é descrita em maiores detalhes, sendo então apresentados os principais problemas desta área. A Seção 4 descreve o campo de Aprendizado de Máquina e as técnicas citadas neste trabalho. Na Seção 5 são apresentadas algumas aplicações de técnicas inteligentes em Bioinformática reportadas na literatura científica. Finalizando, na Seção 6 as considerações finais deste trabalho são destacadas.

[1] O gene é a estrutura fundamental determinante da hereditariedade, codificando as proteínas, componentes essenciais de todo ser vivo.

Como material complementar, no Apêndice A é apresentado um glossário dos principais termos biológicos citados neste trabalho.

2 Conceitos de Biologia Molecular

Nesta Seção é apresentada uma breve introdução aos principais conceitos de Biologia Molecular. Pretende-se apresentar apenas definições necessárias ao entendimento das aplicações de técnicas computacionais neste campo e, portanto, esta caracterização é realizada de forma simplificada.

A **Biologia Molecular** retrata o estudo das células e moléculas, blocos básicos utilizados na construção de todas as formas de vida (Casley, 1992). Em particular, estuda-se o **genoma** dos organismos, definido como o conjunto de informações genéticas dos mesmos.

Esta seção encontra-se dividida em três partes. Na Seção 2.1 é apresentada uma visão geral de conceitos importantes para compreensão deste texto, abrangendo descrições sobre DNA, genes e proteínas. A seguir, o processo de expressão gênica, um dos principais processos celulares estudados em Biologia Molecular, é descrito na Seção 2.2. Na Seção 2.3 são apresentadas duas das principais técnicas laboratoriais utilizadas na medida da expressão de um gene. A análise de dados gerados por este tipo de experimento tem sido amplamente explorada, auxiliando tarefas como, por exemplo, o diagnóstico de tumores (Ben-Dor et al., 2000; Yeang et al., 2001).

2.1 DNA, Genes e Proteínas

Gregor Mendel, em seus experimentos realizados no século XVII, foi o primeiro a identificar fatores responsáveis pela hereditariedade nos organismos vivos (Silva, 2001). A estes fatores, que "codificam" a informação genética, foi dada, posteriormente, a denominação de **genes**. Na busca pela localização dos genes foram identificados os **cromossomos**. Estas estruturas possuem capacidade de replicação (reprodução) e estão presentes em todas células. Os estudos acerca dos cromossomos, por sua vez, levaram à descoberta de que estes são compostos por moléculas de **Ácido Desoxirribonucléico** (DNA) e que os genes são seqüências contíguas de DNA. A Figura 1 ilustra as estruturas citadas e suas relações.

Uma molécula de DNA consiste de duas fitas entrelaçadas em forma de dupla hélice, conforme pode ser visualizado na Figura 2. Cada fita é composta por uma seqüência de **nucleotídeos**, que podem ser de quatro tipos: Adenina (A), Guanina (G), Citosina (C) e Timina (T). Cada nucleotídeo de uma fita se liga a outro complementar da segunda, conforme a regra: A se liga com T, e C com G.

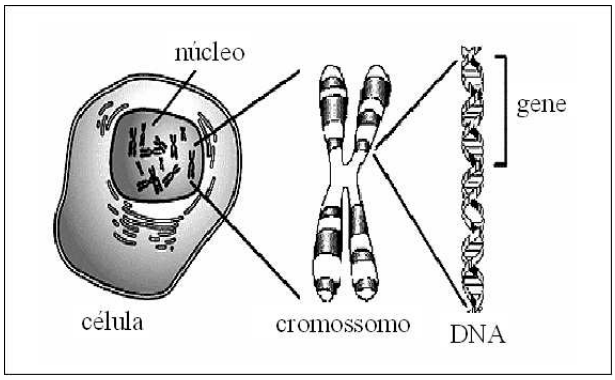

Figura 1: Célula, cromossomo, DNA, gene e suas relações

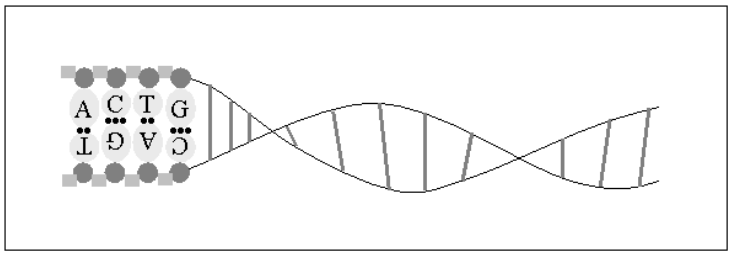

Figura 2: Estrutura de uma molécula de DNA (Souto et al., 2003)

Um fragmento de DNA pode conter diversos genes. A propriedade mais importante dos genes está no fato de que estes codificam **proteínas**, componentes essenciais de todo ser vivo. Ao processo de geração de proteínas a partir da informação contida nos genes dá-se o nome de **expressão gênica**.

As proteínas possuem diversas funções biológicas (Lewis, 2001). Elas podem ter papel estrutural, como no caso do colágeno presente nos tendões, ou estar ligadas a atividades regulatórias, como no caso das enzimas, que catalizam diversas reações químicas nas células.

Estas moléculas também são seqüências lineares, sendo compostas de conjuntos de **aminoácidos**. Existe um total de 20 tipos de aminoácidos (listados na Tabela 1), que se combinam em uma seqüência característica para cada proteína. A composição desta seqüência é denominada **estrutura primária** da proteína (Minatti, 2003).

Ações mútuas entre estes aminoácidos, por meio de interações moleculares, formam outros dois tipos de estruturas: secundária e terciária.

A **estrutura secundária** é definida pelos ângulos formados por ligações peptídicas entre os aminoácidos da proteína. Quando estas ligações fazem a proteína assumir uma forma helicoidal, tem-se uma estrutura denominada hélice alfa. Outras duas estruturas secundárias são as folhas beta, em que segmentos da cadeia de aminoácidos interagem paralelamente, e as voltas, que revertem a cadeia peptídica (Vouet et al., 2002).

Dobras da cadeia proteica sobre ela mesma formam a **estrutura terciária** da proteína (Minatti, 2003). Esta corresponde à estrutura tridimensional da proteína, definindo sua forma e, consequentemente, sua função, expondo sítios de contato com outras cadeias proteicas e genômicas. O que determina a estrutura terciária são as cadeias laterais dos aminoácidos. Algumas destas são tão longas e hidrofóbicas que perturbam a estrutura secundária, provocando dobras na proteína (Minatti, 2003).

Existe ainda uma quarta conformação de proteínas. Algumas delas são compostas por mais de uma unidade (cadeia) proteica. Um exemplo é a imunoglobulina, que é constituída por 4 cadeias proteicas (Minatti, 2003). A conformação espacial conjunta dessas cadeias constitui a **estrutura quaternária** da proteína (Vouet et al., 2002).

A Figura 3 ilustra as quatro conformações estruturais anteriormente descritas.

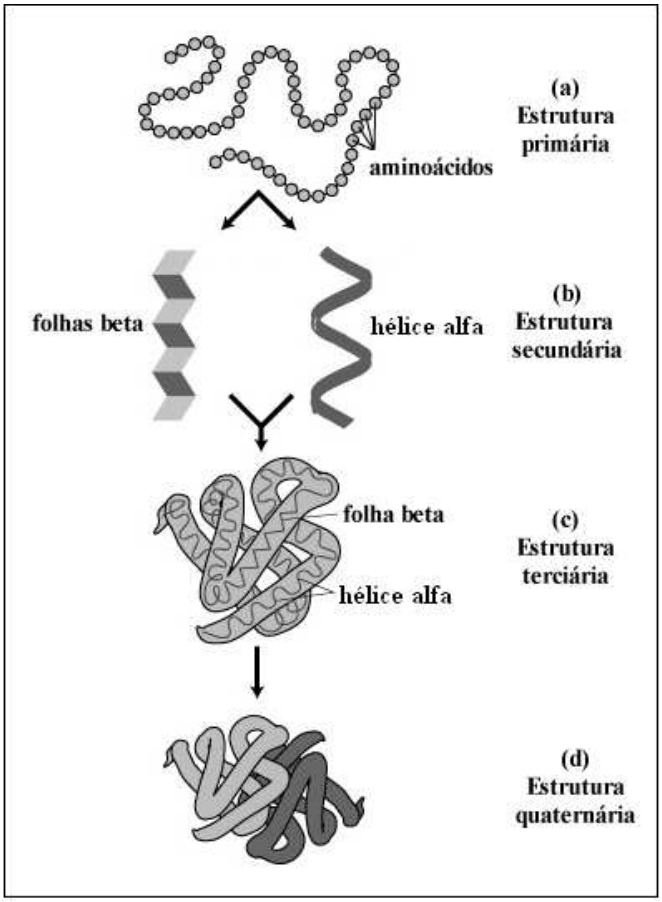

Figura 3: Conformações estruturais das proteínas; (a) estrutura primária; (b) estrutura secundária; (c) estrutura terciária; (d) estrutura quaternária

2.2 Expressão Gênica

A **expressão gênica** constitui o processo pelo qual as seqüências de nucleotídeos dos genes são interpretados na produção de proteínas. Ela é composta por duas etapas, conforme pode ser visualizado na Figura 4.

Na primeira, denominada **transcrição**, a enzima RNA (**Ácido Ribonucléico**) polimerase se liga a uma região do DNA denominada **promotora** e inicia a síntese de um RNA mensageiro (mRNA). O mRNA é bastante similar ao DNA, com exceção de duas características: é composto por apenas uma fita e possui o nucleotídeo Uracila (U) no lugar da Timina (T).

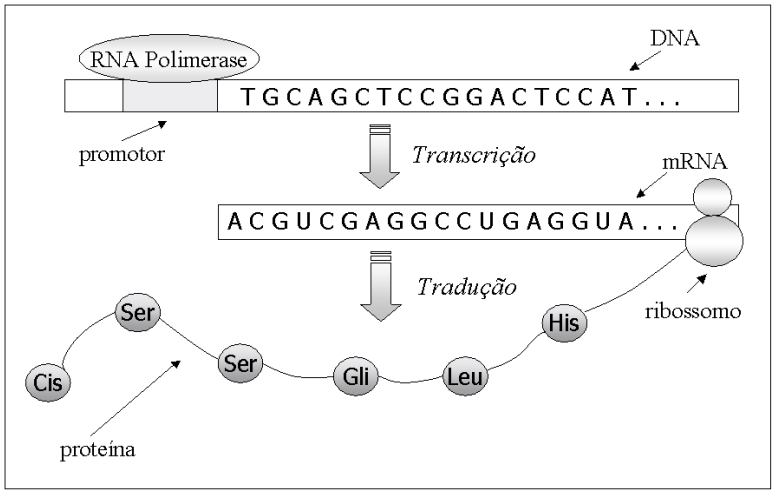

Figura 4: O processo de expressão gênica

Na segunda etapa da expressão é realizada, a partir do mRNA, a síntese da molécula de proteína. Este processo é denominado **tradução**. Cada grupo de três nucleotídeos do mRNA representa um aminoácido, constituinte de uma proteína. O código genético consiste no mapeamento destes grupos, também referenciados por **códons**, nos aminoácidos correspondentes. Há 64 possíveis combinações de triplas de nucleotídeos, ou seja, 64 códons. Existem, porém, apenas 20 aminoácidos. Alguns aminoácidos são, portanto, mapeados por mais de um códon. Além disso, 3 códons são responsáveis por indicar o final da tradução, e são denominados códons de parada. As diferentes codificações podem ser visualizadas na Tabela 1.

Existem algumas diferenças na forma como os procedimentos descritos anteriormente são realizados em organismos **eucariotos** (seres vivos complexos, tais como os humanos), que possuem o material genético em um núcleo delimitado por uma membrana, e **procariotos** (seres unicelulares, como por exemplo as bactérias), que possuem o material genético difuso em suas células. Uma das mais importantes e que merece destaque é

	U	**C**	**A**	**G**	
U	Phelinanina (Phe) Phe Leucina (Leu) Leu	Serina (Ser) Ser Ser Ser	Tirosina (Tir) Tir Parada Parada	Cisteína (Cis) Cis Parada Tritophan (Tri)	**U** **C** **A** **G**
C	Leu Leu Leu Leu	Prolina (Pro) Pro Pro Pro	Histidina (His) His Glutamina (Glu) Glu	Arginina (Arg) Arg Arg Arg	**U** **C** **A** **G**
A	Isoleucina (Iso) Iso Iso Metionina (Met)	Treonina (Tre) Tre Tre Tre	Aspargina (Asp) Asp Lisina (Lis) Lis	Ser Ser Arg Arg	**U** **C** **A** **G**
G	Valina (Val) Val Val Val	Alanina (Ala) Ala Ala Ala	Ácido Áspártico (Aca) Aca Ác. Glutamínico (Acg) Acg	Glicina (Gli) Gli Gli Gli	**U** **C** **A** **G**

Tabela 1: Código Genético (Lewis, 2001). O primeiro, segundo e terceiro nucleotídeos dos códons são representados, respectivamente, pela coluna mais à esquerda, pela primeira linha e pela coluna mais à direita da Tabela.

a de que, em organismos eucariotos, algumas partes da molécula de mRNA não são traduzidas em proteínas. O material genético dos organismos eucariotos possui, portanto, seqüências de nucleotídeos que são codificadas em proteínas, os **exons**, e seqüências que não participam deste processo, os **introns**. As fronteiras entre estas seqüências são denominadas **sítios de *splicing***, nome decorrente do processo de *splicing*, em que os introns são removidos da molécula de mRNA (Figura 5).

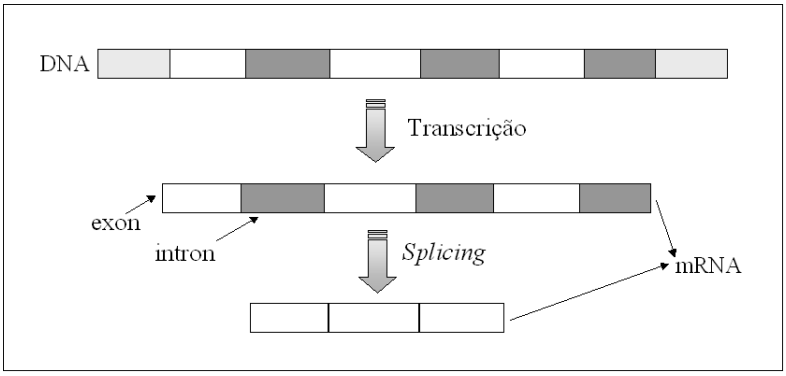

Figura 5: *Splicing* em molécula de mRNA

Todo o código genético de um organismo é frequentemente comparado a um projeto ou planta de construção deste ser vivo. Isto porque cada gene contém um plano para codificação de proteínas, os principais blocos constituintes de todo organismo. Estes planos,

por sua vez, encontram-se organizados em seqüências. O entendimento da formação e distribuição dessas seqüências é uma grande fonte de conhecimento sobre os seres vivos. Análises sobre esses dados, permitem, por exemplo, a compreensão de doenças determinadas geneticamente, formando uma base para estudos médicos e farmacêuticos.

2.3 Técnicas para Medida de Expressão Gênica

O nível de expressão de um gene constitui uma medida da quantidade de proteínas produzidas pelo mesmo em um determinado intervalo de tempo. Experimentos para medida do nível de expressão gênica são usualmente realizados visando monitorar um conjunto de genes em diferentes condições (de pacientes com e sem tratamento por uma nova droga, por exemplo) ou comparar a expressão de um gene em diferentes tipos de tecidos (normal e canceroso, por exemplo) (Furey et al., 2000).

Com respeito à obtenção de dados de expressão gênica, duas técnicas laboratoriais freqüentemente empregadas são os *microarrays* e o SAGE (*Serial Analysis of Gene Expression*) (D'haeseleer et al., 1999).

Os ***microarrays*** são placas de vidro com pontos onde são fixados cDNAs (DNAs complementares produzidos a partir de mRNA) ou oligonucleotídeos (seqüências de DNA curtas) dos genes que se deseja investigar (Souto et al., 2003). A seguir, moléculas de cDNA provenientes do tecido a ser analisado (canceroso, por exemplo) e de um tecido de referência (normal, por exemplo), marcados com cores diferentes, são colocados sobre a placa. Ocorre então um processo denominado **hibridização**, em que as moléculas de cDNA complementares da placa e dos tecidos analisados se ligam. Por meio de um exame computacional do nível de fluorescência em cada ponto da placa pode-se calcular a taxa de expressão dos genes do tecido investigado em relação à do tecido de referência.

Os *microarrays* são largamente utilizados. Seu custo é, porém, em geral alto. Além disso, experimentos com *microarray* são sucetíveis a erros, que podem ocorrer em diversas fases do experimento.

O **SAGE** usa uma abordagem diferente para medida do nível de expressão (D'haeseleer et al., 1999). Inicialmente, são criados milhares de cDNAs dupla-hélice a partir das moléculas de mRNA coletadas do tecido de interesse. De cada cDNA é extraída uma seqüência de 10 a 14 pares de base (longas o suficiente para identificar unicamente cada gene), denominada ***tag***. Por meio de uma série de passos, essas *tags* são concatenadas e podem ser então eficientemente amplificadas e seqüenciadas (Stanton, 2001). Um aplicativo computacional é utilizado na contagem das *tags*, fornecendo assim o nível de expressão de cada gene.

As principais vantagens do SAGE são a não necessidade de se conhecer as seqüências

de mRNA a priori, o que possibilita a descoberta de novos genes, e a utilização da tecnologia de seqüenciamento, já dominada por muitos laboratórios. Este método, porém, é complexo, exigindo uma grande quantidade de seqüenciamento, além de também ser sucetível a erros.

3 Algumas Aplicações em Bioinformática

A Bioinformática pode ser considerada, na prática, um novo ramo das áreas de Biologia e de Computação. Seu principal objetivo é automatizar a obtenção, distribuição e análise de dados genéticos (Baldi and Brunak, 1998).

Estes dados apresentam-se em grande parte na forma de seqüências de nucleotídeos (ou aminoácidos, no caso de proteínas), as quais vêm sendo armazenadas em bases de dados, em geral públicas. Devido ao crescente número de trabalhos relacionados ao seqüenciamento do genoma de diferentes espécies, aliados a resultados de outros tipos de experimentos, como medidas de expressão gênica, estes repositórios vêm acumulando uma grande quantidade de dados com formatos semelhantes.

Atualmente, grande atenção tem se voltado à análise desse grande volume de dados, visando a obtenção, por exemplo, de evidências que relacionem as informações genéticas a determinadas doenças. A realização desse tipo de tarefa de forma manual, em laboratórios, é em geral impraticável e muito custosa. É no exame desses dados que o uso de técnicas computacionais tem se mostrado mais promissor.

Alguns dos problemas da Biologia em que métodos computacionais vêm sendo aplicados com sucesso são listados a seguir:

- **Análise da forma de proteínas:** as funções e propriedades de uma proteína são determinadas por sua estrutura tridimensional (Casley, 1992). Esta estrutura, por sua vez, é definida pela seqüência de genes que codificam a proteína em questão. Uma área de pesquisa que tem despertado grande interesse é a análise dos dados genéticos seqüenciados com o fim de determinar a estrutura das proteínas codificadas.

- **Alinhamento de seqüências:** o objetivo neste caso é alinhar seqüências relacionadas, dado uma função de custo (Setubal and Meidanis, 1997). Com isto permite-se explorar o grau de similaridade entre cadeias de DNA, mRNA ou proteínas. Um grau de similaridade elevado indica uma alta probabilidade das funções executadas pelas moléculas comparadas serem semelhantes. Pode-se então inferir, por exemplo, funções de novas proteínas até então desconhecidas, ou identificar pos-

síveis mutações genéticas. O alinhamento pode ser utilizado também para medir a distância evolutiva entre duas ou mais espécies (Shamir et al., 2001).

- **Identificação de genes em seqüências de DNA:** compreende o reconhecimento e a localização de cada gene em seqüências de DNA. São utilizadas duas abordagens na realização desta tarefa (Craven and Shavlik, 1994). A primeira envolve determinar sinais que podem ser identificados nas seqüências e que indicam situações especialmente adequadas à localização de genes, tais como posições (também denominados sítios) de início de transcrição. O segundo método, entitulado busca por conteúdo, procura por padrões nas seqüências que determinem regiões do DNA que são codificadas em proteínas.

- **Localização de proteínas na célula:** neste caso, o objetivo está em, a partir da seqüência de aminoácidos de uma proteína, prever sua localização na célula (membrana, núcleo, etc.) (Nakai, 2000).

- **Análise de todo genoma:** esta é uma área de atuação em crescimento. Um exemplo de tarefa realizada neste caso está na predição de eventos que determinaram a evolução de dois genomas distintos a partir de um ancestral comum (Shamir et al., 2001).

- **Análise de expressão gênica:** o objetivo neste caso é analisar o nível de expressão de diferentes genes (a "quantidade" de proteínas que este produz). Com isto pode-se, por exemplo, comparar mudanças na expressão de alguns genes frente a alguma doença, possibilitando assim identificar alvos para futuros medicamentos e/ou terapias gênicas (Furey et al., 2000).

Na resolução das tarefas citadas, diversas metodologias computacionais, matemáticas e estatísticas podem ser empregadas. Uma alternativa bastante utilizada em alinhamento de seqüências, por exemplo, é a **programação dinâmica**, que procura particionar um problema em vários subproblemas menores, porém similares, e resolvê-los de forma recursiva (Setubal and Meidanis, 1997).

Técnicas de Aprendizado de Máquina vêm recebendo crescente atenção na resolução dos problemas identificados. Na próxima seção é apresentada uma breve conceituação à respeito dessa área da Inteligência Computacional, assim como descrições de algoritmos de AM atualmente empregados na solução de algumas das tarefas citadas.

4 Aprendizado de Máquina

Aprendizado de Máquina (AM) é um campo de pesquisa da Inteligência Computacional que estuda o desenvolvimento de técnicas capazes de extrair conceitos (conhecimento) a partir de amostras de dados (Mitchell, 1997).

Esta seção apresenta uma introdução ao AM, assim como às principais técnicas desta área empregadas em Bioinformática. Inicialmente, alguns conceitos próprios desta área são introduzidos na Seção 4.1. Na Seção 4.2 são apresentadas algumas das principais técnicas de AM, formando uma base para a compreensão de seu uso na solução de problemas em Bioinformática (Seção 5).

4.1 Conceitos Básicos

As diversas técnicas de AM podem ser divididas, de maneira geral, segundo o paradigma empregado pelas mesmas na análise dos dados. Pode-se dizer que o **paradigma de aprendizado** determina a maneira como o algoritmo de AM se relaciona com seu meio ambiente, ou seja, o modo como se dará o seu aprendizado por meio dos dados (Haykin, 1999). Os dois principais paradigmas de aprendizado existentes são o supervisionado e o não-supervisionado.

No paradigma de aprendizado **supervisionado** tem-se a figura de um "professor externo", o qual representa o conhecimento do ambiente por conjuntos de exemplos na forma (entrada, saída). O algoritmo de AM é então treinado a partir de conjuntos de exemplos rotulados visando aprender uma função desejada. Este é o tipo de paradigma focado neste trabalho.

No paradigma de aprendizado **não-supervisionado** não há a presença de um professor, ou seja, não existem instâncias rotuladas da função a ser aprendida. O algoritmo de AM aprende a representar (ou agrupar) as entradas submetidas segundo uma medida de qualidade. Essas técnicas são utilizadas principalmente quando o objetivo for encontrar padrões ou tendências que auxiliem no entendimento dos dados (Souto et al., 2003).

No contexto de aprendizado supervisionado, dado um conjunto de exemplos rotulados, deve-se produzir um **classificador** (**modelo** ou **hipótese**) capaz de predizer precisamente a classe de novos padrões para os quais não se conhece a classificação. Este classificador representa uma aproximação da **função objetivo** que descreve os dados. Esta é a função que mapeia todos os dados do domínio em que está ocorrendo o aprendizado, incluindo os disponíveis para geração do classificador, em seus respectivos rótulos. Os **rótulos** ou **classes** representam o fenômeno de interesse sobre o qual se deseja fazer previsões.

Cada **exemplo** (também referenciado por **dado**, **padrão** ou **instância**) é tipicamente representado por um vetor de características. Cada **característica**, também denominada

atributo, expressa um determinado aspecto do exemplo. Normalmente, há dois tipos básicos de atributos: nominal e contínuo (Monard and Baranauskas, 2003). Um atributo é definido como **nominal** (ou **categórico**) quando não existe uma ordem entre os valores que este pode assumir (por exemplo, entre cores). No caso de atributos **contínuos**, é possível definir uma ordem linear nos valores assumidos (por exemplo, entre pesos $\in \Re$).

Visando avaliar a taxa de predições corretas ou incorretas (também denominadas **taxa de acurácia** e **taxa de erros**, respectivamente) obtidas por um classificador sobre dados não apresentados em seu treinamento, o conjunto de exemplos é, em geral, dividido em dois subconjuntos disjuntos: de treinamento e de teste. O subconjunto de treinamento é utilizado no aprendizado do conceito e o subconjunto de teste é utilizado para medir o grau de efetividade do conceito aprendido sobre novos dados.

Estimado o erro ou acurácia de um classificador, a **generalização** deste é definida como sua capacidade de prever corretamente a classe de novos padrões, sendo avaliada por seu desempenho em conjuntos de teste. No caso em que o modelo se especializa nos dados utilizados em seu treinamento, apresentando baixa acurácia quando confrontado com novos dados, tem-se a ocorrência de ***overfitting*** (super ajustamento). É também possível induzir hipóteses que apresentam baixa acurácia mesmo no conjunto de treinamento. Tem-se então uma condição de ***underfitting***. Esta situação pode ocorrer, por exemplo, quando os exemplos de treinamento disponíveis são pouco representativos ou o modelo obtido é muito simples (Monard and Baranauskas, 2003).

A seguir, algumas técnicas de AM são apresentadas, com enfoque no paradigma de aprendizado supervisionado.

4.2 Algumas Técnicas de AM

De forma geral, as técnicas de AM pertencentes ao paradigma supervisionado podem ser subdivididas em duas classes gerais: **baseadas em instâncias** e **indutivas** (Mitchell, 1997). Enquanto técnicas indutivas constróem uma descrição geral (um modelo formal) do conceito a ser aprendido a partir das amostras de treinamento, técnicas baseadas em instâncias simplesmente armazenam todos exemplos disponíveis para treinamento. A classe de um novo padrão é então dada de acordo com seu relacionamento com os dados armazenados.

Como exemplos de técnicas baseadas em instâncias, destacam-se o algoritmo k-vizinhos mais próximos e os Sistemas de Raciocínio Baseado em Casos. Seguindo a abordagem indutiva, tem-se as Árvores de Decisão, as Redes Neurais Artificiais e as *Support Vector Machines*. Estas técnicas representam alguns dos algoritmos de AM supervisionado mais utilizados em Bioinformática.

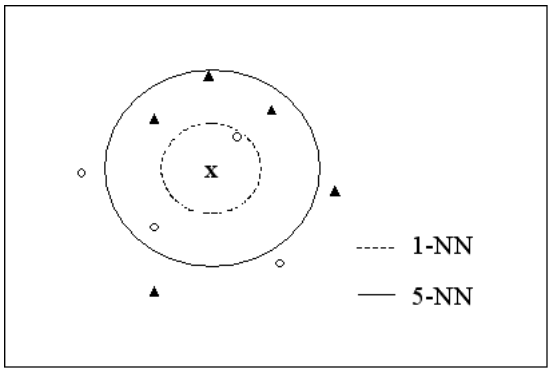

Figura 6: Ilustração do processamento realizado pelo algoritmo k-NN (Mitchell, 1997)

Os Algoritmos Genéticos, que seguem um paradigma de aprendizado referenciado como evolutivo, também constituem um representante de AM empregado em diversas aplicações da área de Bioinformática. O paradigma de aprendizado **evolutivo** engloba algoritmos cujos conceitos e princípios se baseam na teoria da evolução e genética.

A seguir, cada uma das técnicas listadas é descrita.

4.2.1 k-Vizinhos Mais Próximos

O algoritmo **k-vizinhos mais próximos** (*k-Nearest Neighbor* - k-NN) é o representante mais simples da classe de técnicas de AM supervisionado baseadas em instâncias (Mitchell, 1997).

Seu funcionamento reside em armazenar todas as instâncias de treinamento e produzir saídas baseadas em medidas de similaridade com parte destes padrões. A fração de exemplos consultados em cada nova previsão são os k vizinhos mais próximos ao novo padrão submetido. A classificação retornada corresponde à classe mais comum entre os rótulos desses k exemplos.

A Figura 6 ilustra a operação do algoritmo k-NN em um conjunto de dados cujas instâncias são pontos no espaço bidimensional. Este exemplo também ilustra o efeito da variação do valor do parâmetro k, cuja escolha é determinante no desempenho do algoritmo k-NN. Neste exemplo, deseja-se classificar o padrão representado por **x**. Se k é escolhido como 1 (em que tem-se o algoritmo 1-NN), **x** é classificado como pertencente à classe "círculo". Empregando um valor maior de k, por exemplo, 5 (5-NN), **x** é classificado como "triângulo".

Para determinação dos vizinhos mais próximos, o k-NN utiliza uma medida para computação da distância entre pares de dados. Em geral, a medida de distância empregada é

a Euclidiana, representada na Equação 1.

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{q=1}^{m} (x_{iq} - x_{jq})^2} \tag{1}$$

em que $\mathbf{x}_i$ e $\mathbf{x}_j$ são dois padrões pertencentes ao espaço $\Re^m$, e x_{iq} e x_{jq} correspondem aos valores de suas coordenadas (atributos).

Um problema da medida Euclidiana está em pressupor que os dados correspondem a pontos no espaço m-dimensional ($\Re^m$), ou seja, que seus atributos são numéricos (contínuos).

Na computação da distância entre dados com atributos categóricos, normalmente conta-se o número de posições em que os valores dos atributos diferem (Batista et al., 2000). Porém, uma medida de distância mais adequada é a VDM (*Value Difference Metric*) (Stanfill and Waltz, 1986), apresentada nas Expressões 2 e 3.

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{q=1}^{m} (d_{\text{VDM}}(x_{iq}, x_{jq}))^2} \tag{2}$$

$$d_{\text{VDM}}(x_{iq}, x_{jq}) = \sum_{a=1}^{C} \left| \frac{N_a(x_{iq})}{N(x_{iq})} - \frac{N_a(x_{jq})}{N(x_{jq})} \right|^b \tag{3}$$

Na Equação 3, C é o número de classes presentes no conjunto de dados, $N_a(x_{iq})$ é o número de instâncias da classe a com valor de atributo x_{iq}, $N(x_{iq})$ é o número total de exemplos com atributo x_{iq} e b é uma constante, frequentemente igual a 1 ou 2. Pela interpretação desta expressão, tem-se que o VDM considera dois valores de atributos similares se eles ocorrem com freqüência relativa aproximada em todas as classes.

Para o caso em que os dados possuem atributos categóricos e contínuos, pode-se utilizar uma composição das medidas de distância Euclidiana e VDM.

Por meio de seu processamento, o k-NN possui a habilidade de construir aproximações locais da função objetivo, diferentes para cada nova instância a ser classificada. Esta característica, compartilhada por todas técnicas baseadas em instâncias, pode ser vantajosa quando a função objetivo é muito complexa, mas ainda pode ser descrita por uma coleção de aproximações locais de menor complexidade (Mitchell, 1997). Uma vantagem particular do k-NN é sua simplicidade.

Como desvantagem principal do k-NN, assim como de outras técnicas baseadas em instâncias, tem-se o custo de classificar novos padrões, que pode ser bastante elevado.

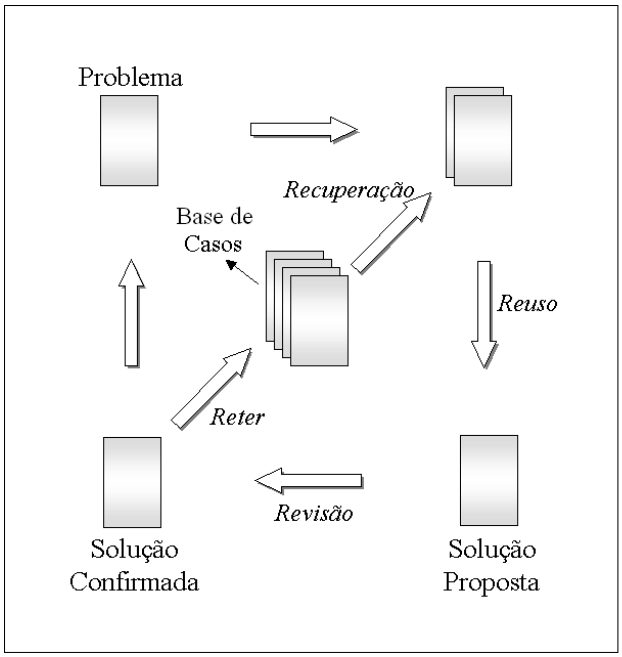

Figura 7: Ciclo do RBC (Aamot and Plaza, 1994)

4.2.2 Raciocínio Baseado em Casos

O **Raciocínio Baseado em Casos** (RBC) se originou da constatação de que especialistas humanos, quando confrontados com novos problemas, em geral utilizam o mesmo raciocínio empregado em situações semelhantes solucionadas no passado. Neste sentido, sistemas de RBC são definidos como processos computacionais que solucionam problemas novos por analogia com antigos, os quais encontram-se armazenados em uma base usualmente denominada **memória** ou **base de casos** (Rich and Knight, 1994).

Os **casos** são informações que representam as experiências passadas (Watson and Marir, 1994). Eles são compostos tipicamente por três partes básicas: uma descrição do problema atacado, a solução utilizada em sua resolução e o resultado alcançado por sua aplicação.

O processamento realizado em um sistema de RBC ao ser apresentado a um novo problema encontra-se esquematizado na Figura 7 (Aamot and Plaza, 1994). Inicialmente, há a recuperação de casos semelhantes armazenados na base. Em seguida, as soluções dos casos recuperados são reusadas ou revistas (adaptadas) para o problema consultado. Se o caso solucionado for relevante e diferente dos demais casos armazenados, este pode ser incluído na base de casos.

Uma preocupação importante em sistemas de RBC diz respeito a como indexar os casos de forma a facilitar sua recuperação. Para realizar esta tarefa, outras técnicas inteligentes podem ser empregadas. Uma Árvore de Decisão, por exemplo, pode ser utilizada para

particionar os dados de acordo com o valor de seus atributos e indexá-los em uma estrutura de árvore.

Como principal vantagem dos sistemas de RBC, tem-se a possibilidade de contínua atualização do conhecimento, por meio do procedimento de inserção de novos casos relevantes na base de casos.

O modelo obtido, que na realidade constitui a base de casos, possui, porém, custo de armazenagem geralmente superior ao das outras técnicas apresentadas.

4.2.3 Árvores de Decisão

As **Árvores de Decisão** (ADs) organizam as informações em uma estrutura composta de nós e ramificações (Baranauskas, 2001). Cada **nó** representa um atributo e está associado a um teste relevante para classificação dos dados, ou a uma classe, no caso do nó ser folha (pertencente à base da AD).

Os **ramos** representam cada um dos possíveis resultados do teste aplicado. A classificação de uma instância se dá seguindo os nós e as ramificações formados até que uma folha seja alcançada. Um exemplo simplificado de uma AD para o diagnóstico de pacientes é ilustrado na Figura 8 (Baranauskas and Monard, 2000).

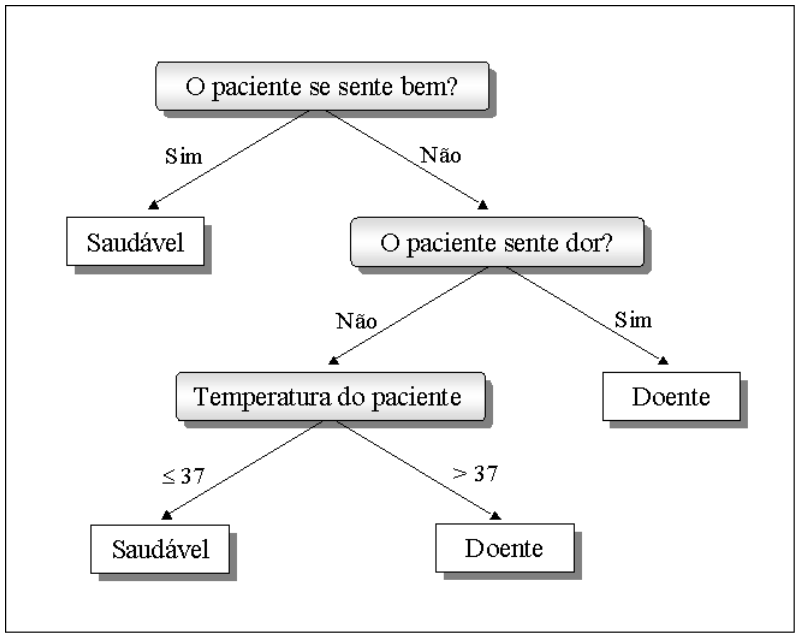

Figura 8: AD simplificada para o diagnóstico de pacientes (Baranauskas and Monard, 2000)

As ADs também podem ser representadas por um conjunto de **regras de decisão**[2] disjuntas. A representação da AD da Figura 8 por meio de regras de decisão é apresentada

[2]Regras de Decisão são formulações do tipo: **Se** (teste = verdadeiro) **então** (classe $\leftarrow x$) **senão** (<nova regra> ou classe $\leftarrow y$), em que o **senão** é opcional.

- **Se** paciente se sente bem **então** classe = Saudável
- **Se** paciente não se sente bem **e** paciente não tem dor **e** temperatura do paciente ≤ 37 **então** classe = Saudável
- **Se** paciente não se sente bem **e** paciente não tem dor **e** temperatura do paciente > 37 **então** classe = Doente
- **Se** paciente não se sente bem **e** paciente tem dor **então** classe = Doente

Figura 9: Representação da AD da Figura 8 por meio de regras de decisão (Baranauskas and Monard, 2000)

na Figura 9.

Há diversos algoritmos para a indução de ADs. Entre os mais utilizados estão o ID3 (Quinlan, 1986) (cujas versões mais recentes denominam-se C4.5 (Quilan, 1988) e C5.0 (Rulequest-Research, 2000)) e o CART (Breiman et al., 1984). O Algoritmo 1 apresenta um pseudo-código do ID3. De maneira simplificada, o ID3 seleciona para teste em cada nó um atributo capaz de separar os dados de diferentes classes de acordo com um critério, como maior ganho de informação. Ele é aplicado recursivamente até que todos dados do conjunto de treinamento sejam corretamente classificados (separados).

Algoritmo 1 Algoritmo ID3 (Mitchell, 1997)

```
Selecionar atributo dos dados de treinamento que gere maior ganho de informação
se primeira vez que o algoritmo é chamado então
    Utilizar atributo selecionado como nó raiz
fim se
se todos os exemplos pertencem a mesma classe então
    Rotular o nó com a classe dos exemplos
    Encerrar execução dessa chamada
senão
    para todo valor que o atributo pode assumir faça
        Criar um ramo saindo da nó atual para um novo nó
        Aplicar o algoritmo ID3 ao novo nó sobre os exemplos por ele cobertos
    fim para
fim se
```

Devido à AD ser gerada de forma a classificar corretamente todos dados de treinamento, ela pode ser sujeita a *overfitting* (Mitchell, 1997). Para evitar este efeito, uma etapa de **poda** é geralmente aplicada à árvore treinada. A poda elimina ramos da árvore que são considerados pouco representativos. Os algoritmos C4.5 e C5.0, sucessores do

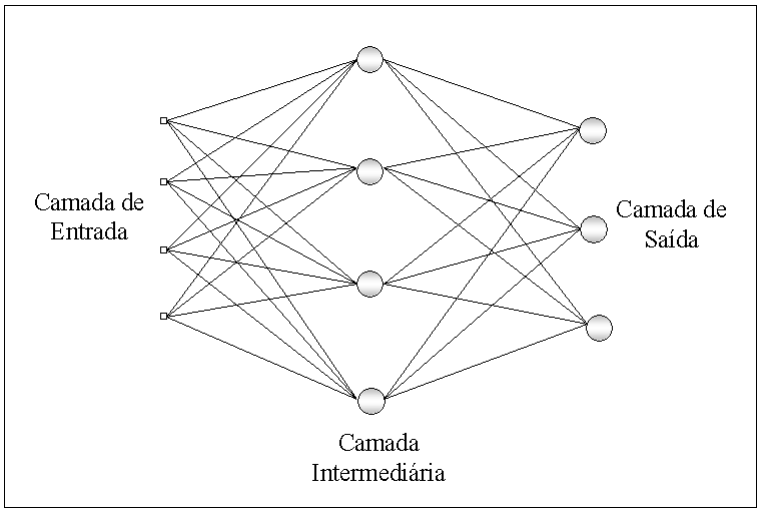

Figura 10: Exemplo de estrutura de uma RN

ID3, implementam este procedimento.

Uma das vantagens das ADs está na compreensibilidade das regras de classificação geradas. Em aplicações médicas, por exemplo, é necessário não somente predizer uma doença, mas também especificar quais sintomas determinaram seu diagnóstico.

As ADs são, porém, pouco robustas a exemplos de grandes dimensões (com muitos atributos), como dados de expressão gênica. O desempenho obtido neste caso mostra-se, em geral, inferior ao de outras técnicas de AM, como as Redes Neurais Artificiais e as *Support Vector Machines*.

4.2.4 Redes Neurais Artificiais

As **Redes Neurais Artificiais** (RNs) são sistemas computacionais inspirados na estrutura, método de processamento e habilidade de aprendizado de um "cérebro biológico" (Cybenko, 1996).

Estas estruturas são compostas de vários elementos de processamento simples, simulando os neurônios biológicos. Estes elementos são os neurônios artificiais, também denominados **nodos** de uma RN, e encontram-se organizados em uma ou mais camadas (Haykin, 1999). Cada elemento de processamento é conectado a um ou vários outros por ligações ponderadas, que procuram simular as sinapses biológicas. Um exemplo esquemático de uma RN é ilustrado na Figura 10.

O conhecimento é adquirido e armazenado em uma RN por meio de ajustes nos pesos das conexões, em um processo denominado treinamento. Existem diversos algoritmos para treinamento de RNs. Seu funcionamento, em geral, consiste em ajustar os pesos da rede de forma a aproximar os valores de saída produzidos pela RN dos valores de saídas

desejados para os padrões do conjunto de treinamento.

Um dos primeiros modelos de RNs foi a rede **Perceptron** (Rosemblatt, 1962). Esta rede possui apenas um nodo. Este encontra-se conectado às entradas por meio de pesos ajustados pela minimização dos erros cometidos no processo de aproximação da função objetivo. Uma limitação das redes Perceptron está em sua incapacidade em lidar com problemas não linearmente separáveis (Minsky and Paper, 1969), ou seja, com conjuntos de dados cujas classes não podem ser separadas por fronteiras lineares.

Para solução de problemas não linearmente separáveis, uma ou mais camadas de nodos intermediários (entre as camadas de entrada e saída) são necessárias. Entre as RNs multicamadas, destacam-se as redes ***Multilayer Perceptron*** (MLP).

As redes MLP possuem uma ou mais camadas de nodos intermediários que computam funções não lineares a partir de entradas ponderadas pelos valores dos pesos das conexões associadas às mesmas. Seu treinamento é geralmente realizado por intermédio do algoritmo *back-propagation* (Rumelhart et al., 1986). Como no caso das redes Perceptron, o algoritmo *back-propagation* utiliza os dados de treinamento para ajustar os pesos da rede por um mecanismo de correção de erros. O *back-propagation* consiste na iteração de duas etapas: *forward* e *backward*. Na fase *forward*, um exemplo de treinamento é apresentado à rede, que produz uma saída. Na fase *backward*, que percorre a RN da camada de saída em direção à camada de entrada, os pesos das conexões da rede são modificados a partir do erro calculado entre a saída desejada e a produzida pela RN. O Algoritmo 2 apresenta um pseudo-código do *back-propagation*.

Algoritmo 2 Algoritmo *back-propagation* (Haykin, 1999)

```
Inicializar pesos da rede com pequenos valores aleatórios
repita
    erro_total = 0
    para cada padrão de treinamento faça
        Apresentar padrão à rede
        Calcular saída_produzida pela rede
        erro = saída_produzida - saída_desejada
        erro_total = erro_total + erro
        para todo camada i da rede, i = n, n − 1, ..., 1 faça
            para cada nodo n_ij da i-ésima camada faça
                Ajustar pesos associados ao nodo
            fim para
        fim para
    fim para
até erro_total < limiar
```

De forma geral, entre as vantagens no uso de RNs estão sua robustez a ruídos e capacidade de representar funções lineares e não-lineares de formas e complexidades variadas.

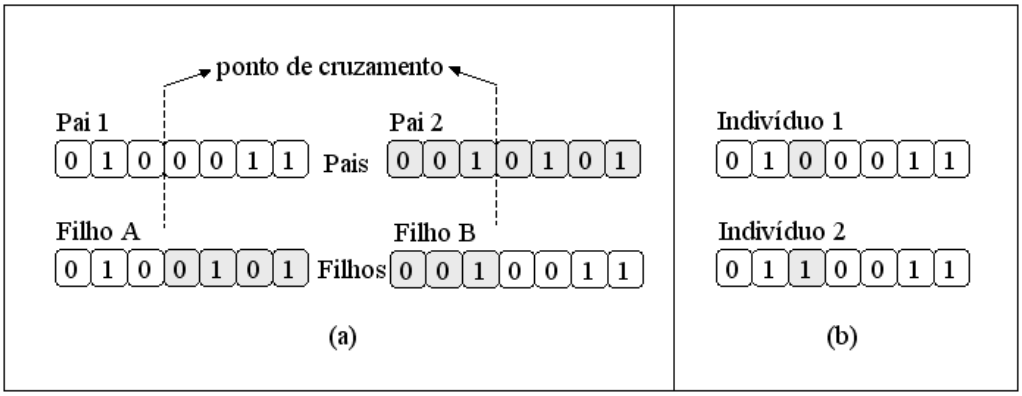

Figura 11: Operadores genéticos de cruzamento (a) e mutação (b)

Como principal desvantagem, tem-se a necessidade do ajuste de diversos parâmetros pelo usuário, os quais influenciam diretamente no desempenho do classificador obtido. Há também problemas quanto à dificuldade de compreensão dos conceitos aprendidos pela rede, os quais encontram-se codificados em seus pesos.

4.2.5 Algoritmos Genéticos

Os **Algoritmos Genéticos** (AGs) são técnicas inteligentes baseadas nos mecanismos do processo de evolução biológica e da genética (Michaelewicz, 1996). Segundo a teoria da evolução, os organismos que se adaptam melhor a seu ambiente possuem maiores chances de terem suas características reproduzidas em uma nova geração.

A utilização desses conceitos em uma forma computacional se dá da seguinte maneira: é gerada inicialmente uma **população** de possíveis soluções ao problema, e busca-se atingir a solução global a partir de constantes evoluções dos melhores indivíduos de acordo com uma **função de aptidão**, que mede a qualidade dos mesmos. Para a definição de cada nova população a partir dos indivíduos escolhidos, três **operadores genéticos** são geralmente aplicados: elitismo (cópias simples dos melhores indivíduos), cruzamento (combinação de estruturas de pares de indivíduos) e mutação (altera a estrutura de alguns indivíduos, visando manter uma variabilidade dentro da nova população). Os dois últimos operadores citados são ilustrados na Figura 11.

Um aspecto importante em relação aos AGs é a forma de representação do problema a ser solucionado. Freqüentemente, os **indivíduos**, que constituem possíveis soluções do problema investigado, são representados por vetores binários. Nestes vetores cada elemento, denominado gene, representa a presença (valor 1) ou ausência (valor 0) de alguma característica (Michaelewicz, 1996). Esta é a codificação ilustrada na Figura 11.

Embora diferentes implementações de AGs variem em alguns aspectos, seu funcionamento básico consiste na alternância de vários ciclos de: geração de população, avaliação

de seus indivíduos e aplicação dos operadores genéticos sobre aqueles considerados mais aptos. O Algoritmo 3 ilustra um exemplo de AG.

Algoritmo 3 Algoritmo Genético (Goldberg, 1989)

1: Gerar população inicial de indivíduos
2: **repita**
3: Avaliar cada membro da população
4: Selecionar indivíduos que participarão da próxima geração
5: Aplicar operadores genéticos aos indivíduos selecionados
6: **até que** critério de parada seja atingido

No Algoritmo 3, para obtenção da população inicial, geralmente utiliza-se o procedimento de atribuir aos indivíduos valores de genes aleatórios (inicializá-los como cadeias de bits aleatórios). A avaliação de cada membro de uma população é realizada segundo a função de aptidão, que mede a efetividade de cada indivíduo na solução do problema investigado. Uma próxima população é então gerada selecionando-se probabilisticamente os indivíduos mais aptos. Indivíduos com maior índice de aptidão têm maiores chances de sobreviver para a próxima geração. Com isto, são produzidos indivíduos cada vez mais aptos, enquanto indivíduos menos adequados tendem a desaparecer (Souto et al., 2003). A aplicação dos operadores genéticos sobre os indivíduos selecionados representa o processo de geração de sucessores em um AG. Os procedimentos de avaliação-seleção-aplicação de operadores genéticos são iterados até que um critério de parada seja atingido. Este critério pode ser após um número fixo de gerações, quando a aptidão do melhor indivíduo exceder um limiar pré-estabelecido, ou quando os indivíduos da população forem muito semelhantes (Michaelewicz, 1996).

Algumas das vantagens dos AGs residem no fato destes realizarem buscas simultâneas em várias regiões do espaço de soluções (Michaelewicz, 1996). Isto permite que se encontre diversas soluções e os torna métodos de busca global. Por este motivo os AGs também são definidos como técnicas de otimização global. Além disso, os AGs são capazes de realizar buscas em espaços de hipóteses (soluções) contendo várias partes complexas, em que o impacto de cada parte na solução do problema pode ser difícil de modelar por meio de técnicas convencionais (Mitchell, 1997).

Por outro lado, uma das principais desvantagens dos AGs é seu custo computacional. Dependendo da complexidade da aplicação investigada, seu uso pode ser computacionalmente caro. Contudo, os AGs são facilmente paralelizáveis, podendo tirar proveito do crescente uso de tecnologias distribuídas.

4.2.6 *Support Vector Machines*

As Máquinas de Vetores Suporte (SVMs - do inglês *Support Vector Machines*) constituem uma técnica de aprendizado que vem recebendo grande atenção nos últimos anos (Hearst et al., 1998). Entre as principais características que tornam seu uso atrativo em Bioinformática estão sua boa capacidade de generalização e robustez diante de dados de grande dimensão, como os presentes em grande parte das aplicações envolvendo o reconhecimento de genes e a análise de dados de expressão.

Esses resultados são alcançados pelo emprego dos conceitos da Teoria de Aprendizado Estatístico (Vapnik, 1995), que apresenta diversos limites na capacidade de generalização de um classificador linear. Para tal, dado um conjunto de treinamento S com n pares $(\mathbf{x}_i, y_i)$, em que $\mathbf{x}_i \in \Re^m$ e $y_i \in \{-1, +1\}$, as SVMs buscam o classificador linear $g(\mathbf{x}) = \text{sgn}(\mathbf{w} \cdot \mathbf{x} + b)$ capaz de separar os dados pertencentes a S com erro mínimo e que também maximize a margem ρ de separação entre as classes presentes em S (Figura 12).

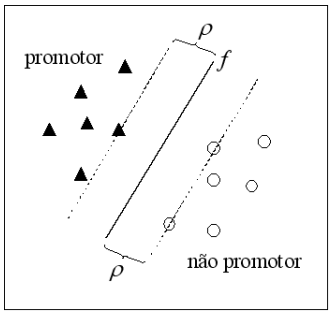

Figura 12: Exemplo simplificado de classificação de promotores por uma SVM linear

Dada uma função linear $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$, a margem $\rho(\mathbf{x}_i, y_i)$ com o qual um padrão $\mathbf{x}_i$ é classificado é fornecida por $y_i f(\mathbf{x}_i)$. Ela mede a distância do padrão $\mathbf{x}_i$ em relação ao hiperplano separador. A margem ρ do classificador linear f é então definida como a margem mínima observada em todo conjunto de treinamento.

Maximizar ρ equivale a minimizar a norma de $||\mathbf{w}||$ (Hearst et al., 1998). Logo, pode-se manter ρ fixo e buscar um hiperplano com $||\mathbf{w}||$ pequeno tal que não haja exemplos de treinamento com margem menor que ρ (Smola and Schölkopf, 2002). Fixando ρ em 1, tem-se o seguinte problema de otimização:

$$\textbf{Minimizar: } ||\mathbf{w}||^2$$
$$\textbf{Sob as restrições: } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geq 0, \text{ para } i = 1, \ldots, n$$

O hiperplano obtido na resolução deste problema é capaz de realizar a classificação de conjuntos linearmente separáveis[3]. Para generalizá-lo a conjuntos mais gerais, pode-se

[3]Um conjunto é linearmente separável se é possível separar os padrões de classes diferentes contidos no mesmo por pelo menos um hiperplano (Mitchell, 1997).

utilizar o artifício de permitir que alguns padrões tenham margem menor que ρ. Isto é obtido com o relaxamento das restrições do problema de otimização apresentado, que é reformulado da seguinte maneira:

$$\textbf{Minimizar: } \|\mathbf{w}\|^2 + B\sum_{i=1}^{n}\varepsilon_i$$

$$\textbf{Sob as restrições: } \begin{cases} \varepsilon_i \geq 0 \\ y_i\left(\mathbf{w}\cdot\mathbf{x}_i + b\right) \geq 1 - \varepsilon_i \end{cases}$$

em que B é uma constante que impõe um peso diferente para o treinamento em relação à generalização e ε representa a variável responsável pela imposição do relaxamento.

Este é um problema clássico em otimização denominado **programação quadrática**, para o qual há uma ampla e estabelecida teoria existente (Hearst et al., 1998). Por meio da resolução do mesmo, obtém-se o classificador representado na Equação 4, em que os α_i correspondem a variáveis determinadas no processo de otimização. Pode-se verificar que a determinação do classificador final se dá unicamente em função de padrões denominados **vetores suporte** (SVs - do inglês *Support Vectors*). Esses padrões correspondem aos exemplos de treinamento mais próximos ao hiperplano separador e são considerados os dados mais informativos do conjunto de treinamento.

$$g(\mathbf{x}) = \operatorname{sgn}(f(\mathbf{x})) = \operatorname{sgn}(\sum_{\mathbf{x}_i \in \mathrm{SV}} \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b) = \begin{cases} +1 \ se \sum\limits_{\mathbf{x}_i \in \mathrm{SV}} \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b > 0 \\ -1 \ se \sum\limits_{\mathbf{x}_i \in \mathrm{SV}} \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b < 0 \end{cases} \tag{4}$$

em que

$$b = -\frac{1}{2}\left[\max_{\{i|y_i=-1\}}(\alpha_i y_i \mathbf{x}_i) + \min_{\{i|y_i=+1\}}(\alpha_i y_i \mathbf{x}_i)\right] \tag{5}$$

O classificador apresentado na Equação 4 ainda tem, porém, utilização limitada. Há muitos casos em que não é possível dividir satisfatoriamente os dados de treinamento por um hiperplano e uma fronteira não linear é mais adequada.

Para generalizar as SVMs para lidar com essas situações mapea-se cada padrão do conjunto de treinamento S para um novo espaço, denominado **espaço de características**. Uma característica singular desse espaço é que a escolha de uma função de mapeamento $\mathbf{\Phi}$ apropriada torna o conjunto de treinamento mapeado linearmente separável. A Figura 13 ilustra este processo. SVMs lineares podem então ser utilizadas sobre o conjunto de treinamento mapeado no espaço de características (Cristianini and Shawe-Taylor, 2000). Para isto, basta aplicar a função de mapeamento $\mathbf{\Phi}$ a cada padrão nas equações listadas para o caso linear.

Por meio desse procedimento, percebe-se que a única informação necessária sobre o mapeamento é uma definição de como o produto interno $\mathbf{\Phi}(\mathbf{x}_i)\cdot\mathbf{\Phi}(\mathbf{x}_j)$ pode ser realizado.

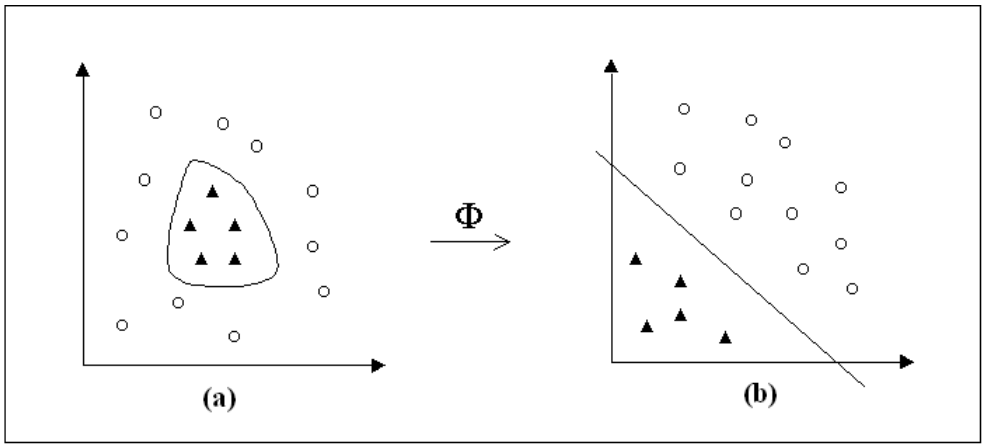

Figura 13: Transformação do conjunto de dados no espaço de entrada (a) para o espaço de características (b)

Isto é obtido com a introdução do conceito de **Kernels**, funções que recebem dois pontos $\mathbf{x}_i$ e $\mathbf{x}_j$ do espaço de entradas e computam o produto escalar $\mathbf{\Phi}(\mathbf{x}_i) \cdot \mathbf{\Phi}(\mathbf{x}_j)$ no espaço de características (Haykin, 1999). De maneira geral, a função Kernel é mais simples que a do mapeamento $\mathbf{\Phi}$. Por este motivo, é comum definí-la sem conhecer-se explicitamente o mapeamento $\mathbf{\Phi}$. Alguns dos Kernels mais utilizados são os Polinomiais, os Gaussianos ou RBF (*Radial-Basis Function*) e os Sigmoidais.

Como já apontado, além de sua boa capacidade de generalização, as SVMs também se destacam por sua robustez em relação a padrões de grandes dimensões.

Suas principais deficiências, semelhante ao caso das RNs, encontram-se em sua sensibilidade a escolhas de parâmetros pelo usuário (embora em geral o número de parâmetros das SVMs seja menor que o das RNs) e na dificuldade de interpretação do modelo gerado por esta técnica.

Deve-se destacar também que as SVMs realizam originalmente classificações binárias. Diversas aplicações em Bioinformática, porém, envolvem mais de duas classes. Existem diversos métodos para generalizar as SVMs a problemas desse tipo. Duas abordagens usuais para tal são a decomposição "um-contra-todos" (1ct) e "todos-contra-todos" (tct) Smola and Schölkopf (2002). Sendo C o número de classes, no primeiro caso produz-se C classificadores, cada um separando uma classe i das $C-1$ restantes. A classe de um novo padrão é dada pelo índice da SVM que produz a maior saída. No caso da decomposição tct, são produzidos $C(C-1)/2$ classificadores para separação de cada classe i de outra j, em que $i, j = 1, \ldots, k$ e $i \neq j$. A saída de um padrão é então usualmente dada por um voto de maioria entre as SVMs.

Um estudo mais detalhado a respeito das SVMs pode também ser consultado em (Lorena and Carvalho, 2003).

5 Aplicações de AM em Bioinformática

O uso de sistemas computacionais no armazenamento e análise do grande volume de dados gerado pelo Projeto Genoma e outros experimentos laboratoriais em Biologia é cada vez mais intenso. Técnicas computacionais convencionais, tais como algoritmos de programação dinâmica, têm sido úteis na realização de um grande número de tarefas de Bioinformática, como, por exemplo, no alinhamento de seqüências e na montagem de fragmentos de DNA (Setubal and Meidanis, 1997).

Porém, a complexidade dos sistemas biológicos, assim como o conhecimento incompleto sobre a Biologia em nível Molecular, restrigem a utilidade de tais métodos em diversas análises. O uso de técnicas de AM mostra-se uma alternativa promissora, já que estas são capazes de lidar com grandes quantidades de dados, mesmo na presença de ruídos (muito comuns nas bases de dados de Biologia Molecular) e na ausência de teorias gerais (Baldi and Brunak, 1998). Esta Seção tem por objetivo apresentar alguns trabalhos da literatura científica em Bioinformática que utilizaram os algoritmos de AM descritos na Seção 4.2 em suas soluções.

5.1 k-Vizinhos Mais Próximos

Diversos trabalhos em Bioinformática reportam resultados do uso do algoritmo k-NN em comparação com aqueles obtidos por outras técnicas de AM. O enfoque neste caso não está no uso particular do k-NN, mas na avaliação do desempenho de diferentes técnicas de AM na solução de um mesmo problema. Este tipo de estratégia foi aplicada, por exemplo, em (Lapedes et al., 1989), (Noordewier et al., 1991) e (Michie et al., 1994), na classificação de sítios de *splicing*. Nos três trabalhos listados, o algoritmo k-NN apresentou desempenho (em termos de acurácia) inferior ao de outras técnicas de AM, como ADs e RNs. Em (Towell et al., 1990), o algoritmo k-NN, com medida de distância que consiste em contar o número de nucleotídeos que diferem entre pares de seqüências, também foi inferior às ADs e RNs no domínio de reconhecimento de promotores em seqüências de DNA.

Em (Horton and Nakai, 1997), o algoritmo k-NN se mostrou eficaz na tarefa de localização de proteínas no ambiente celular. Cada proteína era representada por um conjunto de atributos físico-químicos derivados de sua seqüência de aminoácidos. Para o cálculo da distância entre padrões, utilizou-se a medida Euclidiana. Os resultados obtidos com o k-NN foram comparados ao de ADs e de um classificador Naïve Bayes (descrições sobre esta técnica podem ser consultadas em (Good, 1965)). O k-NN se mostrou superior às outras técnicas investigadas, excedendo acurácias previamente reportadas na localização de proteínas em leveduras e em bactérias *E. coli*.

5.2 Raciocínio Baseado em Casos

Shavlik (1990, 1991) empregaram sistemas de RBC para o reconhecimento de genes. Os casos neste contexto são as seqüências de nucleotídeos de um gene, ou de aminoácidos codificados pelos mesmos. No reconhecimento de uma nova seqüência, esta é submetida ao sistema de RBC, que busca cadeias semelhantes em sua memória. Um alto grau de similaridade entre a cadeia apresentada e uma presente na memória indica a predição de uma região codificadora na primeira seqüência, ou seja, de um gene.

A mesma metodologia pode ser empregada no problema de alinhamento de seqüências, e com isto permitir a descoberta e classificação de novos genes e/ou proteínas. Harris et al. (1993) desenvolveram uma ferramenta de RBC, denominada *ClassX*, que permite a classificação de novas seqüências de aminoácidos em domínios de proteínas[4].

5.3 Árvores de Decisão

Assim como o algoritmo k-NN, as ADs foram aplicadas na solução de diversos problemas em Bioinformática com o intuito de realizar estudos comparativos com outras técnicas de AM (ex.: (Lapedes et al., 1989; Towell et al., 1990; Noordewier et al., 1991; Michie et al., 1994; Craven et al., 1995)). Como as aplicações em Bioinformática em geral se caracterizam por possuir um número elevado de atributos, os resultados das ADs geralmente se mostram inferiores aos de RNs e SVMs, técnicas mais robustas para padrões de grandes dimensões.

Em (Lapedes et al., 1989), por exemplo, ADs, RNs e o algoritmo k-NN foram aplicados na identificação de sítios de *splicing*. Uma terminologia comumente utilizada neste domínio referencia as bordas entre exons e introns como "doadoras" e as bordas introns/exons como "receptoras". Na identificação de doadores e receptores, comumente são utilizadas algumas evidências reportadas em estudos da Biologia. Uma delas é a de que ambas as regiões caracterizam-se por possuir um par de nucleotídeos que se conserva no intron da junção, sendo GT para doadores e AG para receptores. Isto não significa, porém, que estes mesmos pares não estão presentes nos exons, o que dificulta sua utilização para a classificação dessas regiões. Técnicas de AM são então úteis na obtenção de descrições gerais capazes de diferenciar esses sítios.

Lapedes et al. (1989) geraram classificadores diferentes para o reconhecimento de regiões doadoras e receptoras, respectivamente. Observou-se que as RNs produziram os melhores resultados. Entretanto, foi destacado pelos autores que as ADs apresentam a vantagem de organizarem o conhecimento adquirido de forma mais facilmente interpretável. As regras obtidas se mostraram relativamente pequenas e interpretáveis biologi-

[4]Domínios são seqüências de aminoácidos que agem como componentes modulares das proteínas.

camente. Uma ilustração simplificada de uma AD para a identificação de regiões doadoras pode ser visualizada na Figura 14.

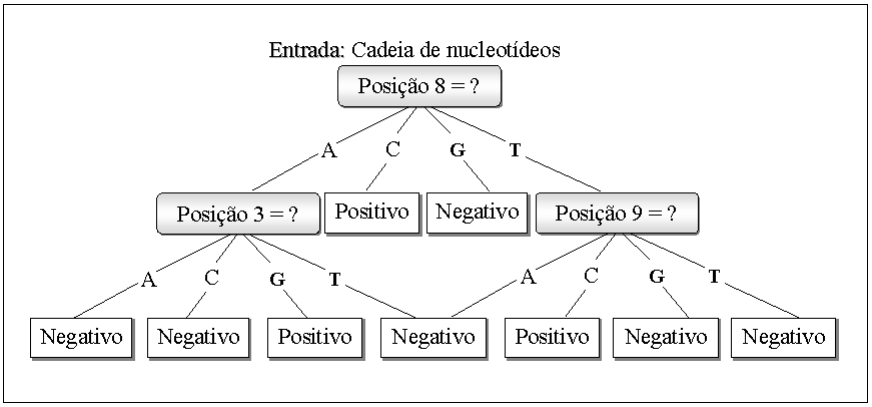

Figura 14: Ilustração de uma AD para a identificação de regiões doadoras (Craven and Shavlik, 1994)

Huss et al. (2001) estudaram a aplicação de ADs no reconhecimento de proteínas específicas do cérebro. O estudo da especificidade de proteínas em relação a um determinado tecido é um problema ainda não satisfatoriamente resolvido. A importância do reconhecimento da especificidade a partir da seqüência da proteína está no fato que com isto permite-se o desenvolvimento de drogas mais eficazes para problemas relacionados a um determinado tecido. Diversas técnicas para geração de ADs e combinações (comitês[5]) de ADs foram investigadas, entre as quais o *Bagging* e o *Adaboost*. Embora o *Adaboost* tenha demonstrado um melhor desempenho, em geral os resultados alcançados pelos diversos algoritmos foram semelhantes.

Para o treinamento desses algoritmos, Huss et al. (2001) não utilizaram as seqüências das proteínas diretamente. Essas seqüências foram antes submetidas a uma série de análises, por meio da utilização de programas e servidores disponíveis na Web. Os resultados desses exames foram então fornecidos às técnicas de aprendizado investigadas. Com isto, os autores pretenderam fazer uso da vasta gama de informações disponíveis atualmente, e obtiveram ganhos em termos de acurácia.

5.4 Redes Neurais Artificiais

Uma das primeiras aplicações de AM em Bioinformática se deu com o emprego de RNs no reconhecimento de sítios de início de tradução (SITs) em seqüências de DNA de bactérias *E. coli*.

[5]Um comitê consiste de um conjunto de classificadores cujas saídas são combinadas na previsão da classe de novas instâncias (Baranauskas and Monard, 2000). Seu estudo é motivado pelo fato de que a combinação da saída de diversos preditores pode melhorar o desempenho obtido por um único preditor.

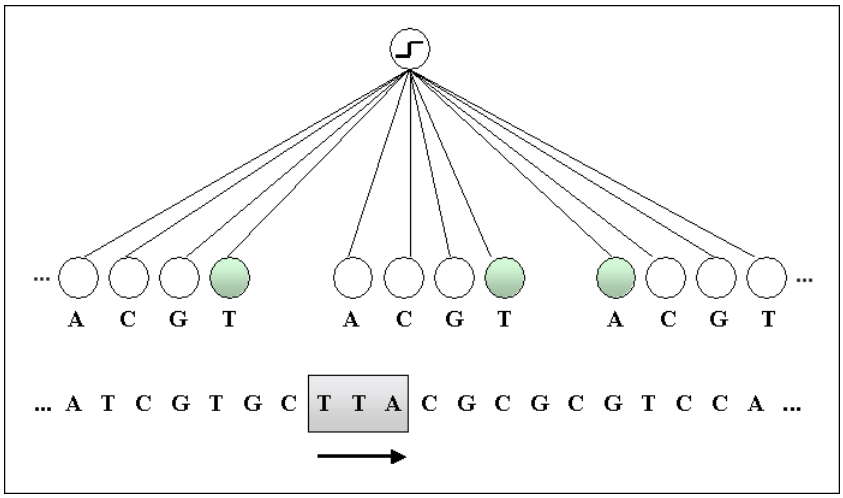

Figura 15: Parte de RN Perceptron utilizada no reconhecimento de sítios de início de tradução no DNA de bactérias *E. coli* (Craven and Shavlik, 1994)

Estes locais iniciam-se geralmente pelo códon AUG, que codifica o aminoácido *Metionina*. Porém, nem todo AUG é um SIT.

Em procariotos, outra informação que pode ser utilizada na identificação de SITs é a presença de uma seqüência denominada *Shine-Dalgarno* (Stormo, 2000). Esta cadeia precede o códon inicial e é complementar à parte do ribossomo que se liga ao mRNA durante a tradução. Sua localização é dificultada pelo fato dessas regiões apresentarem grandes variações na composição de seus nucleotídeos.

Para verificar se a identificação da seqüência *Shine-Dalgarno* em conjunto com o códon iniciador (geralmente um AUG) podem ser considerados suficientes na identificação dos SITs de bactérias *E. coli*, Stormo et al. (1982) utilizaram uma rede do tipo Perceptron nessa aplicação. A rede foi empregada na determinação dos pesos de uma MPP, ou seja, na geração de um modelo para a identificação dos SITs. Os pesos da matriz correspondem então aos pesos da rede, que foi treinada de forma a determinar uma MPP e um limiar capaz de distinguir seqüências com e sem um SIT.

Uma ilustração de parte da rede gerada para esse problema é apresentada na Figura 15. A entrada da RN utilizada é composta de quatro nodos por nucleotídeo na seqüência lida, representando os quatro possíveis valores (A, T, G, C) que este pode assumir (representação também referenciada como canônica). Os valores das ativações da camada de entrada da rede refletem o trecho da seqüência coberto em um dado instante. A saída da rede indica se a seqüência possui ou não um SIT em seu centro. Os autores utilizaram como entrada seqüências de 51, 71 e 101 nucleotídeos. O melhor desempenho foi verificado no último caso.

Segundo Stormo et al. (1982), a RN obtida foi mais precisa que diversos métodos

existentes na época para identificação de SITs. Porém, como destacado na Seção 4.2.4, as redes Perceptron possuem uma grande deficiência pelo fato de só conseguirem classificar padrões linearmente separáveis (Haykin, 1999). Posteriormente ao trabalho de Stormo et al. (1982), houve o advento de diversas técnicas para o treinamento e uso de redes multicamadas, capazes de classificar dados não linearmente separáveis. A partir de então, tais redes vêm sendo aplicadas ao problema descrito, como pode ser visto em (Futschik et al., 1999).

As RNs foram aplicadas a diversos outros problemas da Bioinformática. Elas constituem uma das técnicas de AM mais empregadas neste domínio. Merecem destaque na área de reconhecimento de genes o uso do sistema KBANN (*Knowledge-Based Artificial Neural Network*) (Towell et al., 1990), que combina a abordagem neural e regras de decisão nos problemas de se encontrar regiões promotoras e sítios de *splicing* em seqüências de DNA, os trabalhos de Farber et al. (1992) e o servidor de análise de seqüências GRAIL (Uberbacher et al., 1993), que identificam genes em seqüências de DNA. Outras aplicações de RNs incluem a predição de estruturas de proteínas (Riis, 1995; Polyac et al., 1992; Guan et al., 1994; Ding and Dubchak, 2001; Chung et al., 2003), a classificação de proteínas (Wu et al., 1997; Blekas et al., 2003) e a análise de dados de expressão gênica (Azuaje, 2000; Khan et al., 2001; Toure and Basu, 2001).

5.5 Algoritmos Genéticos

Uma das aplicações na qual os AGs são amplamente empregados em Bioinformática é a predição da estrutura tridimensional de proteínas (Alander, 1995; Krasnogor et al., 1999; Unger and Moult, 1993; Yap and Cosic, 1999). Este problema é bastante complexo, sendo considerado um dos maiores desafios da Biologia MolecularEste problema é bastante complexo e é considerado um dos maiores desafios da Biologia Molecular, pois métodos tradicionais de laboratório desenvolvidos para realização desta tarefa (como, por exemplo, a cristalografia) são custosos e consomem muito tempo para encontrar uma solução.

No problema de alinhamento de seqüências também é reportado o uso desta técnica, como em (Zhang and Wong, 1997). Neste artigo os autores relatam a utilização de AGs em associação a programação dinâmica para realizar buscas por múltiplas seqüências similares. Este problema envolve, além de identificar seqüências similares, apontar suas diferenças, que podem ser de três tipos: inserção, remoção e substituição. Estes conceitos podem ser melhor compreendidos por meio da observação do conjunto de seqüências ilustrado na Figura 16.

Na Figura 16, as colunas 1, 2, 5 e 7 são iguais, e é dito que seus valores "combinam". A coluna 3 possui uma situação de substituição na primeira seqüência, a coluna 4 de que

G	A	C	G	T	G	C
G	A	A	∅	T	∅	C
G	A	A	∅	T	G	C

Figura 16: Conjunto de seqüências similares

uma unidade foi "inserida" na primeira seqüência, e na coluna 6 se verifica o caso em que uma unidade foi "removida" da segunda seqüência. A representação da presença de inserção ou remoção é indicada pelo caractere $\emptyset$.

No sistema proposto por Zhang and Wong (1997), o AG é utilizado na descoberta de blocos que são iguais, e o algoritmo de programação dinâmica lida com a diferença entre as cadeias de nucleotídeos dos blocos identificados. Foi utilizada uma abordagem híbrida pelo fato dos AGs serem eficientes em encontrar blocos iguais de nucleotídeos, porém possuírem dificuldades em apontar as diferenças entre as seqüências. Já o algoritmo dinâmico produz bons resultados na tarefa geral de alinhamento, mas é ineficiente quando confrontado com seqüências múltiplas grandes, que por sua vez são bastante comuns no domínio de Biologia Molecular.

5.6 *Support Vector Machines*

Em um trabalho pioneiro envolvendo o uso de SVMs na identificação de sinais em seqüências genômicas, Zien et al. (2000) aplicaram esta técnica no reconhecimento de sítios de início de tradução (SITs) de organismos vertebrados. O desempenho das SVMs foi comparado ao de RNs, obtidas por Pedersen and Nielsen (1997), e ao de um método probabilístico Markoviano de Salzberg (1997). Os melhores resultados foram observados com a aplicação das primeiras. Foram utilizadas nos experimentos seqüências de mRNA com 200 nucleotídeos.

Como as RNs e as SVMs requerem que os dados estejam em formato numérico, aplicou-se uma codificação canônica de cinco bits a cada nucleotídeo que compõe as seqüências. A posição dos bits indica se o nucleotídeo é A, C, G ou T ou N, quando seu valor for desconhecido.

Outro estudo conduzido neste mesmo trabalho envolveu a incorporação de informações específicas do domínio biológico (conhecimento *a priori*) às funções Kernel. A modificação consistiu em privilegiar correlações locais entre nucleotídeos, enquanto dependências entre nucleotídeos de posições distantes foram consideradas de pouca importância ou inexistentes. A realização dessas simples modificações melhorou consideravelmente os resultados alcançados pelas SVMs.

Motivado pelos bons resultados obtidos pela técnica probabilística de Salzberg (1997), que se mostrou melhor que as RNs para o reconhecimento de SITs, Zien et al. (2000) também investigaram uma reformulação da função Kernel das SVMs considerando as informações providas por esta técnica. Essa alteração levou aos melhores resultados obtidos até então na aplicação considerada.

Outro trabalho em Bioinformática envolvendo o emprego de SVMs é apresentado por Ding and Dubchak (2001). A aplicação considerada foi o reconhecimento de tipos de dobras de proteínas. Os autores compararam os resultados de SVMs aos de RNs na solução desse problema. As RNs também foram utilizadas como classificadores binários, a exemplo das SVMs. É reportado que as SVMs obtiveram os melhores resultados.

Bons resultados também são apresentados no uso de SVMs na análise de dados provenientes de experimentos de *microarray*. Esses dados caracterizam-se por possuir uma grande dimensionalidade (possuem medidas de milhares de genes), o que dificulta sua análise.

Furey et al. (2000) aplicaram SVMs sobre esse tipo de dados para a classificação de tecidos cancerosos. Nos estudos realizados, os autores também buscaram determinar exemplos do conjunto de treinamento que pudessem conter erros em sua classificação, que é usualmente feita a priori por um Biólogo. Os resultados observados se mostraram comparáveis aos de versões modificadas de RNs do tipo Perceptron (Helmbold and Warmuth, 1995; Freund and Schapire, 1998; Grove et al., 1997).

Existem diversos outros trabalhos em que SVMs foram aplicadas na classificação de tumores por meio da análise de dados de expressão gênica. Alguns exemplos são: (Golub et al., 1999), (Yeang et al., 2001), (Ramaswamy et al., 2001) e (Rifkin et al., 2003).

Brown et al. (1999), por sua vez, utilizaram as SVMs sobre dados de *microarray* para classificação de genes. Neste caso, o conjunto de dados é caracterizado por possuir um grande número de exemplos (representados pelos genes) e um número menor de atributos (as amostras de tecidos). Experimentos sobre esses dados sugerem que genes com função similar possuem padrão de expressão semelhante. Nos estudos conduzidos, as SVMs se mostraram superiores às ADs, à janelas de *Parzen* (Bishop, 1995) e ao *Kernel Fisher Discriminant*[6] (Mika et al., 1999).

Hua and Sun (2001) aplicaram SVMs na localização de proteínas no meio celular. Seus resultados foram comparados aos de uma RN, a uma técnica discriminante estatística e a uma abordagem Markoviana. Tanto no domínio de procariotos quanto de eucariotos as SVMs exibiram acurácias superiores às das outras técnicas avaliadas.

Outro trabalho em que SVMs foram utilizadas com bons resultados no campo da

[6]O *Kernel Fisher Discriminant* é uma técnica baseada na teoria Estatística *Fisher Discriminant* e no uso de funções Kernel, similar ao realizado em relação às SVMs (Mika et al., 1999).

Biologia Molecular é reportado em (Karchin et al., 2002). Neste caso, as SVMs foram aplicadas ao problema de reconhecimento de subfamílias de proteínas do tipo "receptor agrupado de proteínas G" (RAPG). As RAPGs são responsáveis por diversos processos fisiológicos das células, entre eles o metabolismo celular, secreção, e a geração de respostas do sistema imunológico e de visão (Hebert and Bouvier, 1998).

5.7 Outras Técnicas

Além das metodologias citadas, técnicas estatísticas como redes Bayesianas (Friedman et al., 2000) e Modelos Markovianos Escondidos (*Hidden Markov Models*) (Grundy et al., 1997; Eddy, 1998; Krogh, 1998) vêm sendo aplicados com êxito em Bioinformática.

Em muitos trabalhos também se verifica a utilização de abordagens híbridas, que combinam diferentes técnicas de AM na solução de um ou mais problemas. Exemplos incluem o sistema KBANN (Towell et al., 1990), que integra aprendizado simbólico (regras de decisão) e conexionista (RNs) para encontrar seqüências promotoras no DNA de bactérias *E. coli* e sítios de *splicing* no DNA de primatas, e o trabalho de Guan et al. (1994), que integra RNs, AGs, busca k-NN e regras heurísticas na predição de estruturas 3D de proteínas a partir de sua cadeia de aminoácidos.

6 Conclusão

Estudos sobre a estrutura genética representam um grande ganho científico, já que pouco se conhece sobre este campo da Biologia Molecular. Como conseqüência permitem, por exemplo, a identificação de doenças determinadas geneticamente, formando uma base para estudos médicos e farmacêuticos.

No momento, os genomas de vários organismos já foram mapeados, e o seqüenciamento do genoma humano se encontra em fase final. Isto representa a aquisição de um grande volume de dados a serem analisados. É nessa etapa que o uso de técnicas computacionais se mostra mais promissor, sendo hoje o foco principal de um campo de pesquisa denominado Bioinformática, que agrega conhecimentos biológicos, computacionais, matemáticos e estatísticos.

Muitas das técnicas utilizadas são inteligentes, pertencentes a um campo de estudo em Inteligência Computacional denominado Aprendizado de Máquina (AM). Este relatório destacou alguns trabalhos de Bioinformática em que vários algoritmos de AM foram empregados com sucesso.

Deve-se ressaltar, porém, que ainda há muitos resultados a serem obtidos e métodos a serem investigados, o que faz o uso de AM em Bioinformática um campo de pesquisa

promissor.

Referências

Aamot, A. and Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1):39–59.

Alander, J. (1995). An indexed bibliography of genetic algorithms and proteins. Technical Report Report 94-1-PROTEIN, University of Vaasa, Department of Information Technology and Production Economics.

Azuaje, F. (2000). Making genome expression data meaningful: Prediction and discovery of classes of cancer through a connectionist learning approach. In *IEEE International Conference on Bioinformatics and Biomedical Engeneering*, pages 208–213.

Baldi, P. and Brunak, S. (1998). *Bioinformatics - The Machine Learning Approach*. The MIT Press.

Baranauskas, J. A. (2001). Extração de conhecimento por múltiplos indutores. Tese de Doutorado, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, Brasil.

Baranauskas, J. A. and Monard, M. C. (2000). Reviewing some machine learning concepts and methods. Technical Report 102, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, Brasil. Disponível em: `ftp://ftp.icmc.usp.br/pub/BIBLIOTECA/rel_tec/RT_102.ps.zip`.

Batista, G. E. A. P. A., Carvalho, A. C. P. L. F., and Monard, M. C. (2000). Applying one-sided selection to unbalanced datasets. In *Mexican International Conference on Artificial Intelligence (MICAI) Lecture Notes in Artificial Intelligence*, volume 1793, pages 315–325. Springer-Verlag.

Ben-Dor, A., Bruhn, L., Friedman, N., and Nachman, I. (2000). Tissue classification with gene expression profiles. *Journal of Computational Biology*, 7(3-4):559–583.

Bishop, C. (1995). *Neural Networks for pattern recognition*. Oxford University Press.

Blekas, K., Fotiadis, D., and Likas, A. (2003). Protein sequence classification using probabilistic motifs and neural networks. In *Joint 13th International Conference on Artificial Neural Networks (ICANN) and 10th International Conference on Neural Information*

Processing (ICONIP) Lecture Notes in Computer Science, volume 2714, pages 702–709, Istambul, Turkey. Springer Verlag.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). Classification and regression trees. Chapman and Hall.

Brown, M., Grundy, W., Lin, D., Christianini, N., Sugnet, C., Jr, M., and Haussler, D. (1999). Support vector machine classification of microarray gene expression data. Technical Report UCSC-CRL 99-09, Department of Computer Science, University California Santa Cruz, Santa Cruz, CA.

Casley, D. (1992). Primer on molecular biology. Technical report, U. S. Department of Energy, Office of Health and Environmental Research.

Chung, I.-F., Huang, C.-D., Shen, Y.-H., and Lin, C.-T. (2003). Recognition of structure classification of protein folding by NN and SVM hierarchical learning architecture. In *Joint 13th International Conference on Artificial Neural Networks (ICANN) and 10th International Conference on Neural Information Processing (ICONIP) Lecture Notes in Computer Science*, volume 2714, pages 1159–1167, Istambul, Turkey. Springer Verlag.

Craven, M. W., Mural, R. J., Hauser, L. J., and Uberbacher, E. C. (1995). Predicting protein folding classes without overly relying on homology. In Rawlings, C. J., Clarck, D. A., Altman, R. B., Hunter, L., Lengauer, T., and Wodak, S. J., editors, *3th International Conference on Intelligent Systems in Molecular Biology*, pages 98–106, Cambridge, UK.

Craven, M. W. and Shavlik, J. W. (1994). Machine learning approaches to gene recognition. *IEEE Expert*, 9(2):2–10.

Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press.

Cybenko, G. (1996). Neural Networks in Computational Science and Engineering. *IEEE Computational Science and Engineering*, 3(1):36–43.

D'haeseleer, P., Liang, S., and Somogyi, R. (1999). Gene expression data analysis and modeling. Pacific Symposium on Biocomputing'99, Tutorial session on Gene Expression and Genetic Networks.

Ding, C. H. Q. and Dubchak, I. (2001). Multi-class protein fold recognition using Support Vector Machines and Neural Networks. *Bioinformatics*, 4(17):349–358.

Eddy, S. R. (1998). Profile Hidden Markov Models. *Bioinformatics*, 14(9):755–763.

Farber, R., Lapedes, A., and Sirotkin, K. (1992). Determination of eukaryotic protein coding regions using neural networks and information theory. *Journal of Molecular Biology*, 226:471–479.

Freund, Y. and Schapire, R. E. (1998). Large margin classification using the perceptron algorithm. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pages 209–217.

Friedman, N., Linial, M., Nachman, I., and Peér, D. (2000). Using Bayesian Networks to analyse expression data. In *Proceedings of the 4th Annual International Conference on Computational Molecular Biology (RECOMB)*, pages 127–135.

Furey, T., Cristianini, N., Duffy, N., Bednarski, D., Schummer, M., and Haussler, D. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):906–914.

Futschik, M., Schreiber, M., Brown, C., and Kasabov, N. (1999). Comparative studies of Neural Network models for mRNA analysis. In *Poster and Software Demonstrations, Proceedings of the International Conference on Intelligent Systems for Molecular Biology*, page 43, Neidelberg, Germany.

Goldberg, D. (1989). *Genetic algorithms in search, optimization, and machine learning.* Addison-Wesley, MA.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D., and Lander, E. S. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537.

Good, I. J. (1965). *The estimation of probabilities: an essay on modern bayesian methods.* MIT Press.

Grove, A. J., Littlestone, N., and Schuurmans, D. (1997). General convergence results for linear discriminant updates. In *Proceedings of the 10th Annual Conference on Computational Learning Theory*, pages 171–183, New York. ACM Press.

Grundy, W. N., Bailey, T. L., Elkan, C. P., and Baker, M. E. (1997). Hidden Markov Model analysis of motifs in steroid dehydrogenases and their homologs. *Biochemical and Biophysical Research Communications*, 231(6):760–766.

Guan, X., Mural, R. J., and Uberbacher, E. C. (1994). Protein structure prediction using hybrid AI methods. In *Proceedings of the 10th Conference on Artificial Intelligence for Applications*, pages 471–473. IEEE Computer Society Press.

Harris, N. L., States, D. J., and Hunter, L. (1993). ClassX: A browsing tool for protein sequence mega classification. In *Proceedings of the 26th Havwaii International Conference on System Sciences*, pages 554–563. IEEE Computer Society Press, Los Alamitos, California.

Haykin, S. (1999). *Neural Networks - A Compreensive Foundation*. Prentice-Hall, New Jersey, 2 edition.

Hearst, M. A., Schölkopf, B., Dumais, S., Osuna, E., and Platt, J. (1998). Trends and controversies - support vector machines. *IEEE Intelligent Systems*, 13(4):18–28.

Hebert, T. E. and Bouvier, M. (1998). Structural and functional aspects of G protein-coupled receptor oligomerization. *Biochemical Cell Biology*, 76(1):1–11.

Helmbold, D. and Warmuth, M. K. (1995). On weak learning. *Journal of Computer and System Sciences*, 50(3):551–573.

Horton, P. and Nakai, K. (1997). Better prediction of protein cellular localization sites with k-nearest neighbor classifiers. In *Proceedings of the Conference in Intelligent Systems and Molecular Biology*, volume 5, pages 147–152.

Hua, S. and Sun, Z. (2001). Support vector machine approach for protein subcellular localization prediction. *Bioinformatics*, 17(8):721–728.

Huss, M., Boström, H., Asker, L., and Coster, J. (2001). Learning to recognize brain specific proteins based on low-level features from on-line prediction servers. In Zaki, M. J., Toivonen, H., and Wang, J. T.-L., editors, *Proceedings of the ACM SIGKDD Workshop on Data Mining in Bioinformatics (BIOKDD 2001)*, pages 45–49.

Karchin, R., Karplus, K., and Haussler, D. (2002). Classifying G-protein coupled receptors with Support Vector Machines. *Bioinformatics*, 18(1):147–159.

Khan, J., Wei, J. S., Ringnér, M., Saal, L. H., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Antoneuscu, C. R., Peterson, C., and Meltzer, P. S. (2001). Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine*, 7(6):673–679.

Krasnogor, N., Hart, W. E., Smith, J., and Pelta, D. A. (1999). Protein structure prediction with evolutionary algorithms. In Banzhaf, W., Daida, J., Eiben, A. E., Garzon, M. H., Honavar, V., Jakiela, M., and Smith, R. E., editors, *Proceedings of the Genetic and Evolutionary Computation Conference*, volume 2, pages 1596–1601, Orlando, Florida, USA. Morgan Kaufmann.

Krogh, A. (1998). An introduction to hidden markov models for biological sequences. In Salzberg, S. L., Searls, D. B., and Kasif, S., editors, *Computational Methods in Molecular Biology*, pages 45–63. Elsevier.

Lapedes, A., Barnes, C., Burks, C., Farber, R., and Sirotkin, K. (1989). Application of neural networks and other machine learning algorithms to DNA sequence analysis. In Bell, G. I. and Marr, T. G., editors, *Computers and DNA*, volume 7, pages 157–182.

Lewis, R. (2001). *Human Genetics - Concepts and Applications.* Mc Graw Hill, 4 edition.

Lorena, A. C. and Carvalho, A. C. P. L. F. (2003). Introdução às máquinas de vetores suporte (*Support Vector Machines*). Technical Report 192, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, Brasil. Disponível em: `ftp://ftp.icmc.usp.br/pub/BIBLIOTECA/rel_tec/RT_192.ps.zip`.

Michaelewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs.* Springer Verlag, 3 edition.

Michie, D., D.J.Spiegelhalter, and C.C.Taylor (1994). *Machine Learning, Neural and Statistical Classification.* Ellis Horwood.

Mika, S., Rätsch, G., Weston, J., Schölkopf, B., and Müller, K.-R. (1999). Fisher discriminant analysis with kernels. In Hu, Y.-H., Larsen, J., Wilson, E., and Douglas, S., editors, *Neural Networks for Signal Processing IX*, pages 41–48. IEEE Computer Society Press.

Minatti, E. (2003). Revista eletrônica de química, Universidade Federal de Santa Catarina. Disponível em `http://www.qmcweb.org`.

Minsky, M. and Paper, S. (1969). *Perceptrons.* MIT Press, Cambridge, MA.

Mitchell, T. (1997). *Machine Learning.* McGraw Hill.

Monard, M. C. and Baranauskas, J. A. (2003). Conceitos de aprendizado de máquina. In Rezende, S. O., editor, *Sistemas Inteligentes - Fundamentos e Aplicações*, pages 89–114. Editora Manole.

Nakai, K. (2000). Protein sorting signals and prediction of subcellular localization. *Advances in Proteins Chemistry*, 54:277–344.

Noordewier, O., G. G. Towell, G. G., and Shavlik, J. W. (1991). Training knowledge-based neural networks to recognize genes in DNA sequences. In *Advances in Neural Information Processing Systems*, volume 3, San Mateo, CA. Morgan Kaufmann Publishers.

Pedersen, A. G. and Nielsen, H. (1997). Neural network prediction of translation initiation sites in eukaryotes: Perspectives for EST and genome analysis. In *Proceedings of the 5th International Conference on Intelligent Systems for Molecular Biology (ISMB'97)*, pages 226–233.

Polyac, M. O., Wilcox, G. L., Xin, Y., Carmeli, T., and Liebman, M. (1992). Prediction of protein folding using the shift-learn method with a large scale neural network. Technical Report UMSI 92/15, University of Minessota Supercomputing Institute.

Quilan, J. R. (1988). *C 4.5 Programs for Machine Learning.* Morgan Kaufmann, CA.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1):81–106.

Ramaswamy, S., Tamayo, P., Rifkin, R., Mukherjee, S., Yeang, C. H., Angelo, M., Ladd, C., Reich, M., Latulippe, E., Mesirov, J. P., Poggio, T., Gerald, W., Loda, M., Lander, E. S., and Golub, T. R. (2001). Multiclass cancer diagnosis using tumor gene expression signatures. In *Proceedings of the National Academic Sciences of USA*, volume 98, pages 15149–15154.

Rich, E. and Knight, K. (1994). *Inteligência Artificial - versão traduzida.* McGraw Hill, 2 edition.

Rifkin, R., Mukherjee, S., Tamayo, P., Ramaswamy, S., Yeang, C.-H., Angelo, M., Reich, M., Poggio, T., Lander, E. S., Golub, T. R., and Mesirov, J. P. (2003). An analytical method for multi-class molecular cancer classification. *Society of Industrial and Applied Mathematics (SIAM) Reviews.* In press.

Riis, S. K. (1995). Combining Neural Networks for protein secondary structure prediction. In *Proceedings of the IEEE International Conference on Neural Networks*, volume 4, pages 1744–1748.

Rosemblatt, F. (1962). *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms.* Spartan Books, New York.

Rulequest-Research (2000). Rulequest research - data mining tools. Disponível em: `http://www.rulequest.com`.

Rumelhart, D. E., Hinton, G. E., and Willians, R. J. (1986). Learning internal representations by error propagation. In Rumelhart, D. E. and McClelland, J. L., editors, *Parallel Distributed Processing: Foundations*, volume 1, pages 318–362. MIT Press, Cambridge, MA.

Salzberg, S. L. (1997). A method for identifying splice sites and translational start sites in eukaryotic mRNA. *Computer Applications in Biosciences*, 13(4):365–376.

Setubal, J. and Meidanis, J. (1997). *Introduction to Computational Molecular Biology.* PWS Publishing Company.

Shamir, R., Kimmel, G., Farkash, A., Klinger, N., and Peles, R. (2001). Algorithms for molecular biology - lecture notes. Disponível em: `http://www.math.tau.ac.il/~rshamir/algmb/01/algmb01.html`.

Shavlik, J. W. (1990). Case-based reasoning with noisy case boundaries: an application in molecular biology. Technical Report CS-TR-1990-988, University of Wisconsin, Madison, WI.

Shavlik, J. W. (1991). Finding genes by case-based reasoning in the presence of noisy case boundaries. In Bareiss, E. R., editor, *Proceedings of the DARPA Cased-Based Reasoning Workshop*, pages 327–338, Washington, US. Morgan Kaufmann.

Silva, F. H. (2001). Apostila - curso de biologia molecular. INBIO - I Escola Brasileira de Inteligência Artificial e Bioinformática, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, Brasil. Disponível em: `http://www.icmc.usp.br/~inbio/material/Apostila_Inbio_Biomol.pdf`.

Smola, A. J. and Schölkopf, B. (2002). *Learning with Kernels.* The MIT Press, Cambridge, MA.

Souto, M. C. P., Lorena, A. C., Delbem, A. C. B., and Carvalho, A. C. P. L. F. (2003). *Técnicas de Aprendizado de Máquina para problemas de Biologia Molecular*, pages 103–152. Minicursos de Inteligência Artificial, Jornada de Atualização Científica em Inteligência Artificial, XXIII Congresso da Sociedade Brasileira de Computação.

Stanfill, C. and Waltz, D. (1986). Toward memory-based reasoning. *Communications of the ACM*, 29(12):1213–1228.

Stanton, L. W. (2001). Methods to profile gene expression. *Trends in Cardiovascular Medicine - Brief Notes*, 11(2):49–54.

Stormo, G. D. (2000). DNA biding sites: representation and discovery. *Bioinformatics*, 16(1):16–23.

Stormo, G. D., Schneider, T. D., and Gold, L. (1982). Use of the Perceptron algorithm to distinguish translation inition sites in E. coli. *Nucleic Acids Research*, 10(9):2997–3011.

Toure, A. and Basu, M. (2001). Application of neural networks to gene expression data for cancer classification. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, volume 1, pages 583–587, Washington, DC.

Towell, G., Shavlik, J., and Noordewier, M. (1990). Refinment of approximate domain theories by knowledge-based neural networks. In *Proceedings of the 8th National Conference on Artificial Intelligence*, pages 861–866. Menlo Park, California, AAAI Press.

Uberbacher, E. C., Einstein, J. R., Guan, X., and Mural, R. J. (1993). Gene recognition and assembly in the GRAIL system: Progress and challenges. In *Proceedings of the 2nd International Conference on Bioinformatics, Supercomputing and Complex Genome Analysis*, pages 465–476. World Scientific, Singapura.

Unger, R. and Moult, J. (1993). A genetic algorithm for 3D protein folding simulations. In *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 581–588. Morgan Kaufmann, San Mateo.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag, 2 edition.

Vouet, D., Vouet, J. G., and Pratt, C. W. (2002). *Fundamentos de Bioquímica - versão traduzida*. Editora Artmed.

Watson, I. and Marir, F. (1994). Case-based reasoning: a review. *The Knowledge Engineering Review*, 9(4):355–381.

Wu, C. H., Zhao, S., Simmons, K., and Shivakumar, S. (1997). Motif neural network design for large-scale protein family identification. In *Proceedings of the 1997 Conference on Neural Networks (ICNN/97)*, pages 86–89. IEEE Computer Society Press.

Yap, A. and Cosic, I. (1999). Application of genetic algorithm for predicting tertiary structures of peptide chains. In *IEEE Proceedings of the 1st Joint BMES/EMBS Conference*, volume 2, page 1214. IEEE Computer Society Press.

Yeang, C.-H., Ramaswamy, Tamayo, Mukherjee, Rifkin, Angelo, Reich, Lander, E., Mesirov, J., and Golub, T. (2001). Molecular classification of multiple tumor types. In *Intelligent Systems in Molecular Biology - Bioinformatics Discovery Note, Proceedings of the 9th International Conference on Intelligent Systems in Molecular Biology*, volume 1, pages 316–322, Copenhagen, Denmark.

Zhang, C. and Wong, A. K. C. (1997). Toward efficient multiple molecular sequence alignment: a system of genetic algorithm and dynamic programming. *IEEE Transactions on Systems, Man and Cybernetics*, 27(6):918–932.

Zien, A., Rätsch, G., Mika, S., Schölkopf, B., Lengaeuer, T., and Müller, K. R. (2000). Engineering support vector machine kernels that recognize translation initiation sites in DNA. *Bioinformatics*, 16(9):799–807.

A Glossário

Este Glossário apresenta um resumo dos principais conceitos definidos na Seção 2. As informações presentes são baseadas nas definições fornecidas em (Lewis, 2001; Casley, 1992; Vouet et al., 2002).

Adenina (A): nucleotídeo componente das moléculas de DNA e RNA.

Aminoácido: moléculas que são combinadas na formação de proteínas.

Citosina (C): um dos tipos de nucleotídeos constituintes das moléculas de DNA e RNA.

Código Genético: seqüência de triplas de nucleotídeos (códons) contidos no RNAm, que determina a seqüência de aminoácidos que forma uma proteína durante a sua codificação.

Cromossomo: estruturas que "guardam"o material genético, sendo constituídos por moléculas de DNA.

DNA (Ácido Desoxirribonucléico): molécula que armazena as informações genéticas de um organismo. É constituída por uma seqüência de nucleotídeos, que podem ser de quatro tipos: Adenina (A), Citosina (C), Guanina (G) e Timina (T). Esta molécula possui a forma de uma hélice dupla. Os nucleotídeos de cada fita constituinte desta estrutura são complementares, segundo a regra: A-T e C-G.

E. coli: bactéria estudada intensivamente por possuir um genoma de pequeno tamanho e ser de fácil reprodução em laboratório.

Expressão Gênica: processo pelo qual um gene é codificado em uma proteína. É composto por duas etapas: transcrição e tradução.

Eucarionte: organismo cujas células possuem o material genético em um núcleo, delimitado por uma membrana.

***Exons*:** regiões codificadoras de proteínas em seqüências de DNA/RNAm.

Gene: unidade fundamental da hereditariedade. Encontram-se nos cromossomos, codificados nas moléculas de DNA.

Genética: estudo dos padrões de herança de características em seres vivos.

Genoma: o genoma de uma espécie constitui todo o material genético da mesma.

Guanina (G): nucleotídeo constituinte das moléculas de DNA e RNA.

***Introns*:** seqüências de DNA/RNAm dos organismos eucariontes que não codificam proteínas.

Mutação: qualquer mudança brusca e não esperada na seqüência de um DNA.

Núcleo: estrutura que armazena o material genético dos organismos eucariontes.

Nucleotídeo: unidade componente das moléculas de DNA e RNA. Podem ser de cinco tipos: Adenina, Guanina, Citosina, Timina e Uracila.

Procariontes: organismos que possuem o material genético disperso em suas células.

Projeto Genoma: conjunto de pesquisas relacionadas ao seqüenciamento e análise do genoma humano e de espécies selecionadas.

Proteína: molécula composta por aminoácidos. Regulam a estrutura e função das células, tecidos e órgãos de todo ser vivo.

RNA (Ácido Ribonucléico): molécula que possui um papel importante na síntese de proteínas e outras atividades químicas das células. Há diferentes classes de RNA, entre eles o RNA mensageiro, cada uma com uma função distinta. O RNA possui uma estrutura de uma única fita, composta de nucleotídeos. Os nucleotídeos presentes nesta molécula são: Adenina (A), Citosina (C), Guanina (G) e Uracila (U).

RNA Mensageiro (RNAm): molécula de RNA utilizada como molde na codificação de uma proteína.

Timina (T): nucleotídeo componente das moléculas de DNA.

Transcrição: síntese de um RNAm a partir de um molde de DNA. É a primeira fase do processo de Expressão Gênica.

Tradução: Processo pelo qual as informações contidas nos nucleotídeos de um RNAm são codificadas em proteínas.

Uracila (U): nucleotídeo componente das moléculas de RNA.