Técnicas de Aprendizado de Máquina para Problemas de Biologia Molecular

M. C. P. de Souto, A. C. Lorena, A. C. B. Delbem, A. C. P. L. F. de Carvalho

[1]Laboratório de Inteligência Computacional (LABIC)
Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo-São Carlos
Av. do Trabalhador São-Carlense, 400
Cx. Postal 668 - CEP 13566-590
São Carlos - São Paulo - Brasil

{marcilio,aclorena,acbd,andre}@icmc.usp.br

Abstract. *In the last years, the amount of biological data stored in databases has been growing exponentially. As a consequence, the current conventional techniques used to analyze data have been shown limited. Furthermore, the rich information contained in this kind of data and its broad biological implication require new techniques for data analysis. Among the strategies employed for such analysis, one can mention Machine Learning, which can learn automatically from the available data, yielding useful hypothesis. The aim of this tutorial is to introduce Machine Learning techniques in the context of their applications to Molecular Biology problems (gene prediction, analysis of gene expression data and phylogeny construction).*

Resumo. *Nos últimos anos, o acúmulo de dados biológicos vem aumentando exponencialmente. Com isso, meios convencionais para a análise de dados mostram-se restritos. Além disso, a rica informação contida nesses dados e sua vasta implicação biológica requerem novas técnicas para sua análise. Dentre as estratégias utilizadas para tal, pode-se destacar o Aprendizado de Máquina, que provê técnicas capazes de aprender automaticamente a partir dos dados disponíveis e produzir hipóteses úteis. O objetivo deste tutorial é introduzir técnicas de Aprendizado de Máquina aplicadas a três problemas de Biologia Molecular: predição de genes, análise de dados de expressão gênica e construção de filogenia.*

1. Introdução

Dados biológicos estão sendo disponibilizados a uma taxa muito elevada, fazendo com que os bancos de dados atuais cresçam exponencialmente [Baldi e Brunak, 2001]. Esse fenômeno vem sendo causado pela utilização de novas e eficientes técnicas na análise de seqüências de genoma e proteoma. Manipular e analisar os dados acumulados nessas bases de dados tornou-se um dos maiores desafios da Bioinformática.

A **Bioinformática** ou **Biologia Computacional** diz respeito à utilização de técnicas e ferramentas de computação para a resolução de problemas da Biologia [Baldi e Brunak, 2001]. Dentre as diversas áreas da Biologia, aquela em que a aplicação de técnicas computacionais tem se mostrado mais promissora é a Biologia Molecular [Setúbal e Meidanis, 1997]. Nesse contexto, a computação pode ser aplicada na resolução de problemas como comparação de seqüências (DNA, RNA e proteínas), montagem de

fragmentos, reconhecimento de genes, identificação e análise da expressão de genes e determinação da estrutura de proteínas [Setúbal e Meidanis, 1997, Baldi e Brunak, 2001].

O emprego de métodos computacionais na Biologia iniciou-se na década de 1980, quando biólogos experimentais, em conjunto com cientistas da computação, físicos e matemáticos, começaram a aplicar esses métodos na modelagem de sistemas biológicos. Durante esse período, ferramentas computacionais foram desenvolvidas por essa comunidade para análise dos dados, utilizando algoritmos convencionais da Ciência da Computação.

No entanto, as ferramentas que usam computação convencional têm se mostrado limitadas para abordar problemas biológicos complexos. Isto vem ocorrendo, entre outras razões, devido à ausência de uma teoria fundamental em nível molecular. Outra razão para essa dificuldade é a ineficiência das ferramentas convencionais em lidar com grandes quantidades de dados. Técnicas de Aprendizado de Máquina (AM) [Mitchell, 1997] são assim cada vez mais empregadas para tratar problemas em Biologia Molecular, por sua capacidade de aprender automaticamente a partir de grandes volumes de dados e produzir hipóteses úteis [Baldi e Brunak, 2001].

Um dos objetivos desse tutorial é propiciar uma visão ampla das principais abordagens de AM utilizadas em problemas específicos e relevantes da Biologia Molecular. Os problemas cobertos são reconhecimento de genes, análise de dados de expressão gênica e construção de filogenia. As técnicas empregadas são representativas dos diferentes paradigmas de AM: redes neurais artificiais (aprendizado conexionista), máquinas de vetores suporte (aprendizado estatístico), algoritmos de agrupamento (aprendizado estatístico), algoritmos genéticos (aprendizado evolutivo) e árvores de decisão (aprendizado simbólico).

É importante ressaltar que há uma infinidade de outros problemas da Biologia Molecular que podem ser abordados com técnicas de AM, por exemplo, a predição de estrutura de proteínas [Baldi e Brunak, 2001]. Da mesma forma, existem diversas outras técnicas de AM que vêm sendo aplicadas com sucesso aos problemas descritos neste tutorial, como os algoritmos de agrupamento baseados em modelos [Molla et al., 2003, Ji et al., 2003]. O enfoque escolhido para este tutorial, porém, foi o de apresentar um conjunto de problemas relevantes para a Biologia Molecular, com um esforço em descrevê-los de maneira concisa, por meio de exemplos concretos de como pesquisadores vêm solucionando-os com algoritmos de AM.

Esse tutorial está dividido em seis seções. A Seção 2 descreve alguns conceitos e definições fundamentais da Biologia Molecular, para tornar mais claro o entendimento dos tipos de problema biológicos que serão abordados. Na Seção 3 são apresentados os principais conceitos de AM e sua terminologia básica, como também uma introdução às técnicas aqui utilizadas para abordar os problemas biológicos.

As principais seções do tutorial são as Seções 4, 5 e 6. Na Seção 4, o tema tratado é o reconhecimento e previsão da estrutura de genes, tarefa que geralmente envolve diversos passos e técnicas. Nesse contexto, algoritmos de AM podem ser aplicados em uma ou mais etapas preditivas, por exemplo, na identificação de sítios de *splicing* e de regiões codificadoras. A Seção 5 aborda o problema de análise de dados de expressão gênica, dividindo-o em três sub-problemas: identificação de sub-classes de doenças, identificação e predição da funcionalidade dos genes, e classificação de doenças. Na Seção 6 é investigado o uso de algoritmos evolutivos na construção de árvores filogenéticas. Por fim, na Seção 7 é apresentado um resumo do tutorial apontando direções futuras para cada um dos problemas abordados.

2. Principais Conceitos de Biologia Molecular

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 suas informações genéticas. Gregor Mendel, em seus experimentos realizados no século XVII, foi o primeiro a identificar fatores responsáveis pela hereditariedade nos organismos vivos [Silva, 2001].

Esses fatores foram, posteriormente, denominados de **genes**, os quais codificam a informação genética. Na busca pela localização dos genes foram identificados os **cromossomos**, que são estruturas que possuem capacidade de replicação (reprodução) e estão presentes em todas as células. Estudos acerca dos cromossomos, por sua vez, levaram à descoberta de que eles são compostos por moléculas de **Ácido Desoxirribonucléico** (DNA) e que genes são seqüências contíguas de DNA.

Esta seção está dividida em duas partes. Na Seção 2.1, um resumo de conceitos importantes, como DNA, expressão gênica e proteínas, é apresentado. A compreensão desses conceitos é fundamental para o entendimento deste tutorial. Posteriormente, na Seção 2.2, são introduzidos conceitos mais específicos à expressão gênica, conhecimento importante para a compreensão da Seção 5.

2.1. DNA, Expressão Gênica e Proteínas

Uma molécula de DNA consiste de duas fitas anti-paralelas entrelaçadas em forma de dupla hélice, conforme pode ser visualizado na Figura 1. Cada fita é composta por uma seqüência de nucleotídeos (bases), 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: $A = T$, $T = A$, $C \equiv G$ e $G \equiv C$, em que cada "$-$" representa uma ponte de hidrogênio.

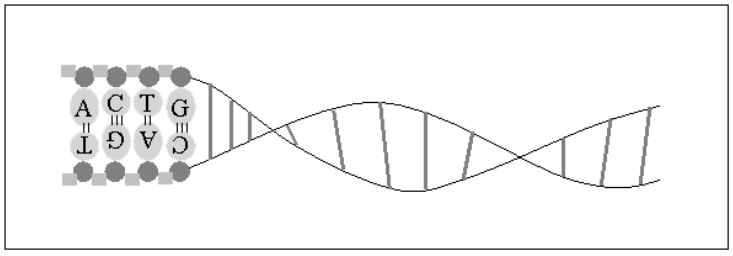

Figura 1: Estrutura de uma molécula de DNA

Como resultado dessas regras de ligação, as duas seqüências de bases que formam as moléculas de DNA são complementares entre si. Moléculas de DNA de fita simples, que são encontradas apenas em condições especiais, têm a capacidade de se ligar a seqüências complementares em um processo chamado de **hibridização**. Esse processo é utilizado em muitas das técnicas da Biologia Molecular.

Um fragmento de DNA pode conter diversos genes. A propriedade mais importante dos genes está no fato de que eles codificam proteínas, componentes essenciais de todo ser vivo. 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 catalisam diversas reações químicas nas células.

As **proteínas** também são seqüências lineares, compostas de conjuntos de aminoácidos. O processo pelo qual as seqüências de nucleotídeos dos genes são interpretadas na produção de proteínas é denominado **expressão gênica** (Figura 2). A expressão

é composta por duas etapas: na primeira, denominada **transcrição**, um 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 do nucleotídeo Timina (T).

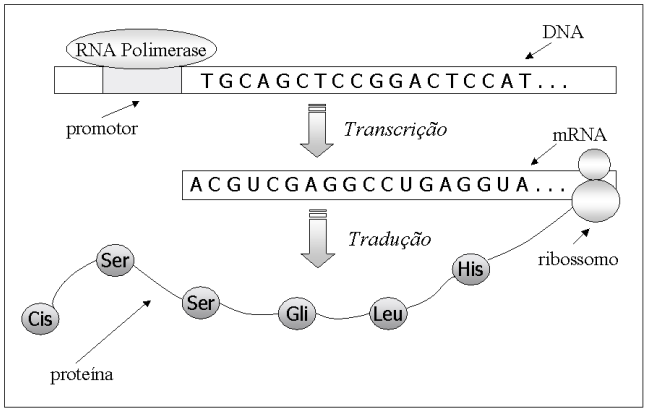

Figura 2: Processo de expressão gênica

Na segunda etapa da expressão, chamada **tradução**, é realizada a síntese da molécula de proteína, a partir do mRNA. Cada grupo de três nucleotídeos do mRNA representa um **aminoácido**, constituinte de uma proteína. O **código genético** consiste no mapeamento desses grupos, também referenciados por **códons**, nos aminoácidos correspondentes. Há 64 possíveis combinações de triplas de nucleotídeos, ou seja, 64 códons. Porém, existem apenas 20 aminoácidos. Portanto, muitos deles são mapeados por mais de um códon.

Desses 64 códons, 3 são responsáveis por indicar o final da tradução, sendo denominados códons de parada. As diferentes codificações podem ser visualizadas na Tabela 1. O primeiro, segundo e terceiro nucleotídeos dos códons são representados, respectivamente, pela coluna mais à esquerda, a primeira linha e a coluna mais à direita da tabela.

	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]

Existem algumas diferenças na forma como os procedimentos descritos anteriormente ocorrem em organismos **eucariotos** (seres vivos complexos, tais como os huma-

nos), 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 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 desse processo, os **introns**. As fronteiras entre essas seqüências são denominadas **sítios de *splicing***, nome decorrente do processamento no qual os introns são removidos da molécula de mRNA (Figura 3).

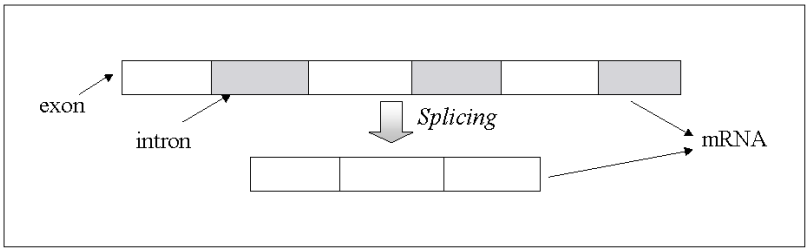

Figura 3: *Splicing* em moléculas de mRNA

Todo o código genético de um organismo é usualmente comparado a um projeto ou planta de construção desse ser vivo. Isto porque cada gene contém um plano para a codificação de proteínas, os principais blocos constituintes de todo o organismo. Esses planos, por sua vez, encontram-se organizados em seqüências. O entendimento da formação, relação e distribuição dessas seqüências é uma grande fonte de conhecimento sobre os seres vivos.

Por exemplo, como será visto na Seção 6, o uso de seqüências de DNA ou de aminoácidos vem contribuindo bastante para o desenvolvimento de filogenias (estimativa dos relacionamentos evolutivos entre um conjunto de objetos que tenham uma mesma origem).

2.2. Experimentos com Expressão Gênica

A análise da expressão dos genes é de grande interesse para as Ciências Biológicas. Esse tipo de análise pode fornecer informações importantes sobre as funções de uma célula, uma vez que as mudanças na fisiologia de um organismo são geralmente acompanhadas por mudanças nos padrões de expressão dos genes [Alberts et al., 1997].

Diversas técnicas têm sido propostas para obtenção da expressão dos genes: MPSS (*Massively Parallel Signature Sequence technology*), SAGE (*Serial Analysis of Gene Expression*), Real-time RT-PCR (*Reverse-Transcription Polymerase Chain Reaction*) e *microarray* de DNA [Brenner et al., 2000, Velculescu et al., 1995, Freeman et al., 1999, Harrington et al., 2000]. Muitas dessas técnicas podem ser utilizadas em estudos de genomas inteiros, da expressão de genes ativos, no ordenamento e sequenciamento dos genes, na determinação de variantes genéticas, em diagnósticos de doenças e várias outras aplicações [Slonim, 2002].

No caso de *microarray* de DNA, o principio básico empregado é o seguinte: moléculas de DNA complementar (cDNA[1]) ou oligonucleotídeos[2] correspondentes aos genes cuja expressão deve ser analisada (sondas) são afixadas, de uma maneira ordenada

[1]Molécula de DNA produzida a partir de um mRNA e, portanto, sem introns [Alberts et al., 1997].
[2]Seqüências de DNA curtas de 20 a 30 nucleotídeos [Alberts et al., 1997].

(*arrays*), a um suporte sólido que pode ser uma lâmina (*slide*) de vidro. A miniaturização e automação da criação dessas lâminas com o uso de robôs (ou síntese *in situ* de oligonucleotídeos) tornou possível a sua produção com milhares de genes (isto é, uma parte substancial do genoma) representados em poucos centímetros quadrados - *microarrays*.

Ainda no caso específico de *microarray* de cDNA, primeiro as sondas são replicadas um grande número de vezes. Em seguida, um robô fixa essas sondas em determinados pontos (*spots*) da lâmina de vidro. Ao final, a lâmina conterá milhares de pontos com DNA, colocados lado a lado, cada ponto contendo milhares de sondas de cDNA que foram projetadas para hibridizar com o mRNA de um certo gene.

Em um próximo passo, para medir a abundância relativa dos transcritos correspondentes a uma determinada célula, as suas moléculas de mRNA são também transcritas para moléculas de cDNA. Essa transcrição é necessária porque moléculas de RNA são muito instáveis, tendendo a degradar rapidamente. Posteriormente as moléculas de cDNA produzidas são, em geral, marcadas com rótulos fluorescentes verdes (Cy3). Da mesma forma, as moléculas de mRNA da célula de controle ou referência também são separadas e transcritas, sendo que nesse caso são marcadas com rótulos vermelhos (Cy5).

As moléculas de cDNA de ambas as células são então despejadas na lâmina. Depois de um certo tempo, a lâmina é lavada, removendo as moléculas de cDNA que não hibridizaram com as sondas. Em seguida, a lâmina é escaneada, produzindo como resultado uma imagem com as intensidades de todos os pontos (todo o processo é ilustrado na Figura 4). A imagem digital da lâmina é, por fim, processada por meio de métodos computacionais, com o objetivo de calcular a intensidade obtida para cada mRNA.

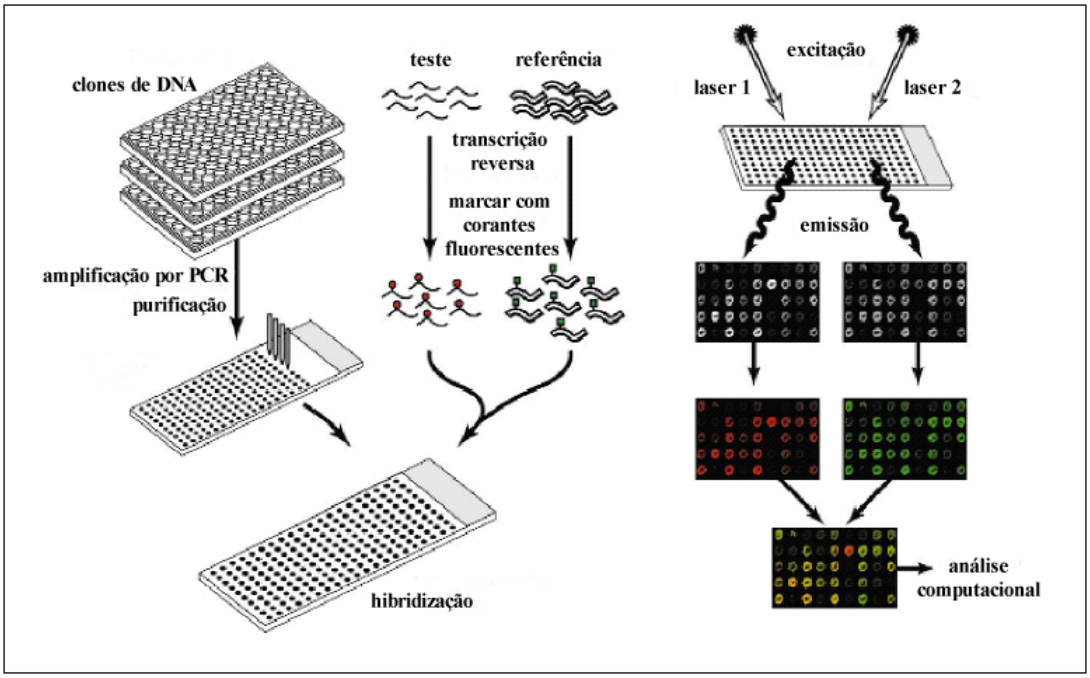

Figura 4: Esquema de um *microarray* de cDNA [Duggan et al., 1999]

Devido ao avanço das tecnologias utilizadas na obtenção de dados de expressão gênica, o volume desses dados vem aumentando exponencialmente. Como conseqüência, meios convencionais de armazenamento, análise e comparação de dados têm se tornado limitados. Além disso, a rica informação contida nesses dados e sua vasta implicação biológica requerem novas tecnologias para sua análise. Dentre dessas novas tecnologias, merece destaque o desenvolvimento de algoritmos sofisticados baseados em AM, discutidos na Seção 5.

3. Técnicas de Aprendizado de Máquina

Aprendizado de Máquina (AM) estuda como construir programas de computador que melhoram seu desempenho em alguma tarefa por meio de experiência. Aprender, nesse contexto, pode ser definido como a seguir (para situações em que o desempenho em alguma tarefa pode ser medido): diz-se que um programa computacional aprende a partir da experiência E, em relação a uma classe de tarefas T, com medida de desempenho P, se seu desempenho nas tarefas T, medida por P, melhora com a experiência E [Mitchell, 1997].

O objetivo principal desta seção é introduzir as principais técnicas de AM utilizadas neste tutorial: redes neurais artificiais, máquinas de vetores suporte, algoritmos genéticos, árvores de decisão e algoritmos de agrupamento. Com o propósito de facilitar a apresentação dessas técnicas, alguns conceitos básicos de AM são inicialmente apresentados.

3.1. Conceitos Básicos

Técnicas de AM podem ser divididas, de maneira geral, em aprendizado supervisionado e aprendizado não supervisionado. Se antes do processo de aprendizado o indutor recebe um conjunto de exemplos, cada exemplo sendo formado por um conjunto de atributos de entrada e um conjunto de atributos de saída (rótulos), então esse tipo de aprendizado pode ser classificado como **aprendizado supervisionado**. Os seguintes algoritmos de aprendizado supervisionado são abordados neste tutorial: redes neurais artificiais to tipo *multilayer perceptron*, máquinas de vetores de suporte, algoritmos genéticos e árvores de decisão.

Em contraste, **aprendizado não supervisionado** é realizado quando, para cada exemplo, apenas os atributos de entrada estão disponíveis. Essas técnicas de aprendizado são utilizadas quando o objetivo for encontrar em um conjunto de dados padrões ou tendências (aglomerados) que auxiliem o entendimento desses dados. Este tutorial aborda os seguintes algoritmos de aprendizado não supervisionado: redes neurais do tipo mapa auto-organizáveis, algoritmo k-médias e algoritmos de agrupamento hierárquico.

Também com o propósito de facilitar o entendimento dos termos utilizados em AM, é apresentada a seguir uma lista dos conceitos mais usados no tutorial [Mitchell, 1997, Monard e Baranauskas, 2003a].

- **Exemplo (padrão, instância)**: um objeto único do mundo a partir do qual um modelo será aprendido, ou sobre o qual um modelo será usado (por exemplo, para predição). Na maioria dos trabalhos em AM, exemplos são descritos por vetores de características. Um exemplo de padrão poderia ser uma amostra de tecido de um paciente, que poderia estar associada à presença ou ausência de câncer.
- **Característica (atributo, variável)**: uma quantidade descrevendo um exemplo. Um atributo tem um domínio definido pelo seu tipo, que denota os valores que ele pode assumir. No caso dos padrões (pacientes) do item anterior, cada atributo poderia ser o valor do nível de expressão de um gene.
- **Vetor de características**: uma lista de características que descreve um exemplo. Para o exemplo apresentado no item anterior, poderia ser um vetor m-dimensional descrevendo todos os m genes medidos para o tecido de um determinado paciente.
- **Classe**: no aprendizado supervisionado, todo exemplo possui pelo menos um atributo especial denominado rótulo ou classe, que descreve o fenômeno de interesse. No caso em que os exemplos são tecidos de pacientes, as classes poderiam ser presença de câncer e ausência de câncer.
- **Conjunto de exemplos (conjunto de dados)**: é composto por um número de exemplos (padrões) com seus respectivos valores de atributos. No caso de apren-

dizado supervisionado, a cada exemplo também é associada uma classe. Usualmente, o conjunto de exemplos é dividido em dois subconjuntos disjuntos: o **conjunto de treinamento**, utilizado para o aprendizado do conceito e o **conjunto de teste**, utilizado para medir o grau de efetividade do conceito aprendido.

- **Acurácia (taxa de erro)**: a taxa de predições corretas (ou incorretas) realizada pelo modelo para um determinado conjunto de dados. A acurácia é, em geral, estimada utilizando um conjunto independe de teste, que não foi usado em nenhum momento durante o processo de aprendizado. Existem outros meios de estimar a acurácia por meio de técnicas mais complexas, como *cross-validation* e *bootstrap* [Mitchell, 1997].
- **Falso positivo**: dado um classificador para discriminar classes A e B (supondo que A é a classe mais relevante ou positiva), o número de falsos positivos é a quantidade de exemplos da classe B classificados como da classe A. Do mesmo modo, o número de **falsos negativos** é a quantidade de exemplos da classe A classificados como da classe B.
- **Ruído**: é comum, no mundo real, trabalhar com dados imperfeitos. Eles podem ser derivados do próprio processo que gerou o dados, do processo de aquisição de dados, do processo de transformação ou mesmo de classes rotuladas incorretamente. Nesses casos, diz-se que existe ruído nos dados - que é bastante comum em dados obtidos com *microarrays* [Slonim, 2002].
- **Overfitting (super-ajustamento)**: ocorre quando o modelo se especializa nos dados utilizados no seu treinamento, apresentando uma taxa de acurácia baixa para novos dados.

3.2. Redes Neurais Artificiais

Uma rede neural artificial é um modelo de computação inspirado na forma como a estrutura paralela e densamente conectada do cérebro dos mamíferos processa informação. Mais formalmente, **Redes Neurais Artificiais** (RNs) são sistemas paralelos distribuídos compostos por unidades de processamento simples (nodos) que calculam determinadas funções matemáticas (normalmente não-lineares). Essas unidades são dispostas em uma ou mais camadas e interligadas por um grande número de conexões. As conexões estão associadas a pesos, os quais armazenam o conhecimento representado no modelo e servem para ponderar a entrada recebida por cada nodo da rede. A Figura 5 ilustra uma típica RN com mais de uma camada, chamada RN multi-camadas.

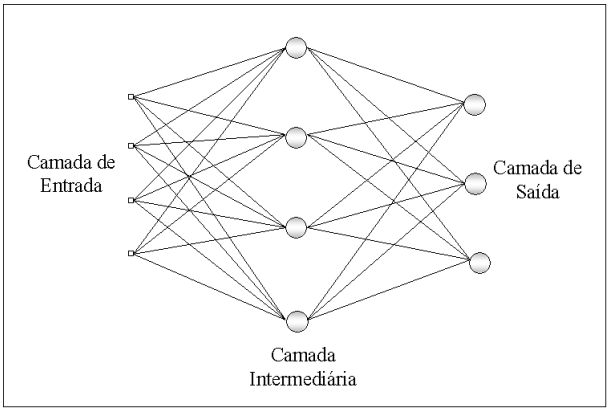

Figura 5: Exemplo de uma rede neural multi-camadas

O aprendizado em sistemas biológicos envolve ajustes nas sinapses que existem entre os neurônios. De forma similar, o aprendizado em RNs ocorre por meio da apresentação de um conjunto de padrões (representando um dado problema) à rede. Estes

padrões são utilizados por por um algoritmo de treinamento para, iterativamente, ajustar os pesos das conexões (sinapses). O objetivo do processo de treinamento é extrair o conhecimento necessário para a resolução do problema em questão. O conhecimento armazenado nos pesos é usado, posteriormente, para gerar a resposta da rede para novos padrões.

A primeira RN utilizada em problemas de reconhecimento de padrões foi a rede Perceptron, proposta por [Rosenblatt, 1958]. A rede Perceptron original utiliza apenas um nodo cujos pesos podem ser ajustados durante o treinamento da rede. Uma limitação desse tipo de RN é sua incapacidade de lidar com problemas que não sejam linearmente separáveis[3] [Minsky e Papert, 1969].

Problemas não-linearmente separáveis podem ser tratados por RNs com uma camada intermediária (entre as camadas de entrada e saída). Porém, apenas em 1986 foi apresentado um algoritmo de treinamento, chamado *backpropagation*, para essa classe de redes [Rumelhart et al., 1986]. O tipo de RN multi-camada usada com o *backpropagation* é geralmente denominada de rede *Multi-Layer Perceptron* (MLP) [Rumelhart et al., 1986]. As redes MLPs treinadas com o algoritmo *backpropagation* têm sido um dos modelos de RNs mais usado em aplicações do mundo real [Haykin, 1999], inclusive em problemas de Biologia Molecular [Towell et al., 1990, Rampone, 1998, Xu et al., 2002, Faceli et al., 2003].

O algoritmo *backpropagation* utiliza pares (entrada, saída desejada) para ajustar os pesos da rede por meio de um mecanismo de correção de erros. O treinamento usando esse algoritmo ocorre em duas fases, cada uma percorrendo a rede em um sentido: fase *forward*, em que é produzida a saída da rede para um dado padrão de entrada, e fase *backward*, em que os pesos das conexões da rede são atualizados de acordo com o erro, calculado a partir da diferença entre a saída desejada e a saída produzida pela rede. Existem diversas variações desse algoritmo que têm como objetivo acelerar o processo de treinamento e reduzir as taxas de erros obtidas. O Algoritmo 1 apresenta os principais passos do algoritmo *backpropagation*.

As RNs apresentam uma série de vantagens, como tolerância a dados ruidosos, habilidade de representar qualquer função (linear ou não) e capacidade de lidar com padrões de entrada representados por vetores de alta dimensão, em que os valores dos atributos podem ser contínuos ou discretos. Os principais problemas são a dificuldade de definição de seus parâmetros, como por exemplo, no caso das redes MLP, o número de nodos em suas camadas intermediárias, o tipo de função de ativação e o valor da taxa de aprendizado, além da dificuldade de compreensão dos conceitos aprendidos pela rede, codificados nos valores finais dos pesos da rede.

Por fim, é importante ressaltar que existem vários modelos de RNs, cada um deles mais adequado para um dado tipo de problema. Alguns dos modelos mais populares incluem as redes MLP com o *backpropagation* que foram revisados nesta seção, as redes do tipo mapa auto-organizável (que serão introduzidas na Seção 3.6), redes com funções de base radial, redes de Hopfield, entre outras [Haykin, 1999].

3.3. Máquinas de Vetores Suporte

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 popularizaram seu uso em Bioinformática estão sua boa capacidade de generalização e robustez diante de dados de

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

Algoritmo 1 Algoritmo Backpropagation [Haykin, 1999]

```
1: Inicializar pesos da rede com valores aleatórios;
2: repita
3:    erro_total = 0;
4:    para todo cada padrão de treinamento faça
5:       para todo cada camada i da rede, i = 1, 2, ..., n faça
6:          para todo cada nodo n_ij da i-ésima camada faça
7:             Calcular saida_produzida do nodo;
8:          fim-para
9:       fim-para
10:      erro = saida_desejada - saida_produzida;
11:      para todo cada camada i da rede, i = n, n − 1, ..., 1 faça
12:         para todo cada nodo n_ij da i-ésima camada faça
13:            Ajustar pesos do nodo;
14:         fim-para
15:      fim-para
16:      erro_total = erro_total + erro;
17:   fim-para
18: até que erro_total > valor_desejado
```

grande dimensão, como os presentes em grande parte das aplicações envolvendo o reconhecimento de genes e a análise de dados de expressão gênica.

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 E 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 $g(\mathbf{x}) = \mathrm{sgn}(\mathbf{w} \cdot \mathbf{x} + b)$ capaz de separar os dados pertencentes a E com erro mínimo e maximizar a margem ρ de separação entre as classes presentes em E (Figura 6).

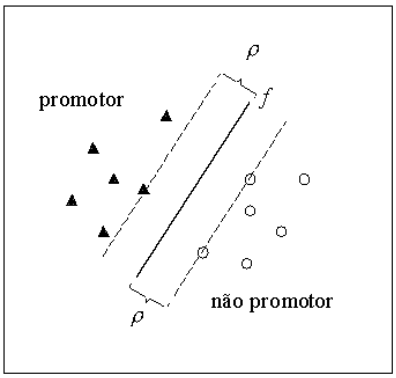

Figura 6: Exemplo simplificado de problema de classificação de promotores por meio de uma SVM linear

Dada uma função linear $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$, a margem $\rho(\mathbf{x}_i, y_i)$ utilizada para classificar um padrão $\mathbf{x}_i$ 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 e 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. Para o caso de conjuntos mais gerais, pode-se 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:

$$\textbf{Minimizar: } \|\mathbf{w}\|^2 + C \sum_{i=1}^{n} \varepsilon_i$$
$$\textbf{Sob as restrições: } \begin{cases} \varepsilon_i \geq 0 \\ y_i \left(\mathbf{w} \cdot \mathbf{x_i} + b\right) \geq 1 - \varepsilon_i \end{cases}$$

em que C 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 [Hearst et al., 1998]. Por meio da resolução deste problema, obtém-se o classificador representado na Equação 1, em que as variáveis α_i são 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.

$$g(\mathbf{x}) = \text{sgn}(f\left(\mathbf{x}\right)) = \text{sgn}(\sum_{\mathbf{x_i} \in \text{ SVs}} \alpha_i y_i \mathbf{x_i} \cdot \mathbf{x} + b) = \begin{cases} +1 \ se \sum\limits_{\mathbf{x_i} \in \text{ SVs}} \alpha_i y_i \mathbf{x_i} \cdot \mathbf{x} + b > 0 \\ -1 \ se \sum\limits_{\mathbf{x_i} \in \text{ SVs}} \alpha_i y_i \mathbf{x_i} \cdot \mathbf{x} + b < 0 \end{cases} \tag{1}$$

em que

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

O classificador apresentado na Equação 1 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, mapeia-se cada padrão do conjunto de treinamento E 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. SVMs lineares podem então ser utilizadas sobre o conjunto de treinamento mapeado no espaço de características [Cristianini e 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 calculado. 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 defini-la sem conhecer-se

explicitamente o mapeamento Φ. Alguns dos Kernels mais utilizados são os Polinomiais, os Gaussianos ou RBF (*Radial-Basis Function*) e os Sigmoidais.

A escolha da função Kernel e seus parâmetros, assim como da constante C no problema de otimização, tem influência direta no desempenho do classificador gerado por uma SVM [Müller et al., 2001]. Essa sensibilidade a escolhas de parâmetros representa uma das deficiências das SVMs. Outra deficiência diz respeito à dificuldade de interpretação do modelo gerado, como no caso das RNs.

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 multiclasses. Duas abordagens usuais para tal são a decomposição "um-contra-todos" (1ct) e "todos-contra-todos" (tct) [Smola e Schölkopf, 2002].

3.4. Algoritmos Genéticos

Algoritmos Genéticos (AGs) constituem uma das linhas de pesquisa da Computação Evolutiva (CE), área de AM que estuda algoritmos inspirados na teoria da evolução e na genética. Os AGs foram introduzidos em meados de 1976 por John Holland e seus colaboradores da Universidade de Michigan [Holland, 1992]; mas seu pleno desenvolvimento só ocorreu a partir da década de 80, por meio do trabalho de [Goldberg, 1989]. Além de AGs, a CE engloba três linhas de pesquisa tradicionalmente conhecidas como: Programação Evolutiva, Estratégia Evolutiva, Sistemas Classificadores e Programação Genética [Bäck et al., 1997].

Pode-se definir AGs como técnicas de otimização global que empregam uma estratégia de busca paralela e estruturada que, embora aleatória, consegue, em geral, melhorar a qualidade das soluções produzidas a cada a nova iteração (geração). Por sua capacidade de explorar simultaneamente vários pontos do espaço de busca, o risco de estacionamento em mínimos locais é reduzido. Esses algoritmos conseguem lidar bem com situações em que o espaço de busca é amplo e complexo [Pham e Karaboga, 2000].

Os AGs atuam sobre uma população de indivíduos, baseados no fato de que indivíduos com boas características genéticas têm maiores chances de sobrevivência e de produzirem indivíduos cada vez mais aptos, enquanto indivíduos menos aptos tendem a desaparecer. Cada **indivíduo** da população, chamado cromossomo, corresponde a uma solução para um dado problema. Um mecanismo de reprodução, baseado em processo evolutivo, é aplicado sobre a população atual com o objetivo de explorar o espaço de busca e encontrar melhores soluções para um dado problema.

O primeiro passo para a utilização de um AG é a geração de uma população de indivíduos, em geral determinada aleatoriamente, que podem ser vistos como possíveis soluções para o problema apresentado. Durante o processo evolutivo, esta população é avaliada atribuindo para cada indivíduo uma nota, por meio de uma **função de aptidão**, que indica quão boa é a resposta a ele associada para a solução do problema investigado. Indivíduos com melhores índices de aptidão têm maiores chances de sobreviver para a próxima geração, produzindo assim indivíduos cada vez mais aptos, enquanto indivíduos menos aptos tendem a desaparecer.

Uma porcentagem dos mais adaptados é escolhida, enquanto os outros são descartados. Os membros selecionados podem sofrer modificações em suas características fundamentais por meio de operadores genéticos, formando uma nova população ou geração. Este processo, chamado de reprodução, é repetido até que um conjunto de soluções satisfatórias seja encontrado [Goldberg, 1989].

Os operadores genéticos mais utilizados são o *crossover*, a mutação e o elitismo. Os dois primeiros têm seu funcionamento ilustrado nas figuras 7 e 8, respectivamente. O operador de elitismo faz com que um ou mais indivíduos mais aptos sejam automaticamente passados à geração seguinte. Para finalizar a execução de algoritmo, é utilizado um critério de parada. Este critério pode ser, por exemplo, após um número fixo de gerações, quando a aptidão do melhor indivíduo de uma população superar um dado valor ou quando os indivíduos de uma população se tornarem muito semelhantes. O Algoritmo 2 mostra o pseudo-código de um AG típico.

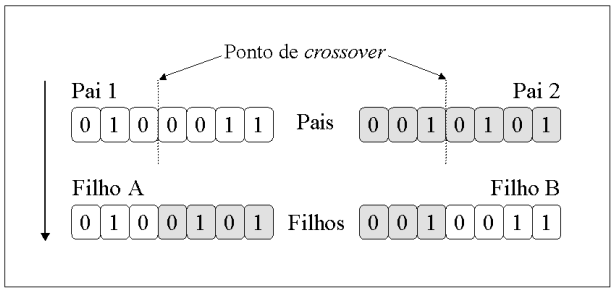

Figura 7: Operador genético de *crossover*

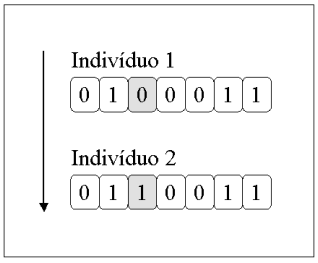

Figura 8: Operador genético mutação

Algoritmo 2 Algoritmo Genético [Goldberg, 1989]

```
1: Escolher população inicial de indivíduos;
2: repita
3:     Selecionar indivíduos mais aptos;
4:     Aplicar operadores genéticos aos indivíduos selecionados;
5:     para todo indivíduo da população faça
6:         Avaliar aptidão do indivíduo;
7:     fim-para
8: até que critério de parada seja atingido
```

É crescente o uso de AGs para a resolução de problemas em Biologia Molecular, como, por exemplo, filogenia e alinhamento de sequências [Matsuda, 1996, Lewis, 1998, Brauer et al., 2002, Katoh et al., 2001, Cancino et al., 2003].

3.5. Árvores de Decisão

Na sua forma mais simples, uma **Árvore de Decisão** (AD) é uma lista de perguntas com respostas do tipo “sim” ou “não”, hierarquicamente arranjadas, que levam a uma decisão. Por exemplo, com o propósito de determinar se um trecho de DNA é um gene, pode-se

construir uma árvore como a mostrada na Figura 9. Uma árvore como essa é fácil de manipular, por ter um número finito de possibilidades em cada ramo, e qualquer caminho pela árvore levar a uma decisão. A estrutura da árvore e as regras em cada um dos ramos são determinadas a partir de um conjunto de dados por meio de um processo de aprendizado.

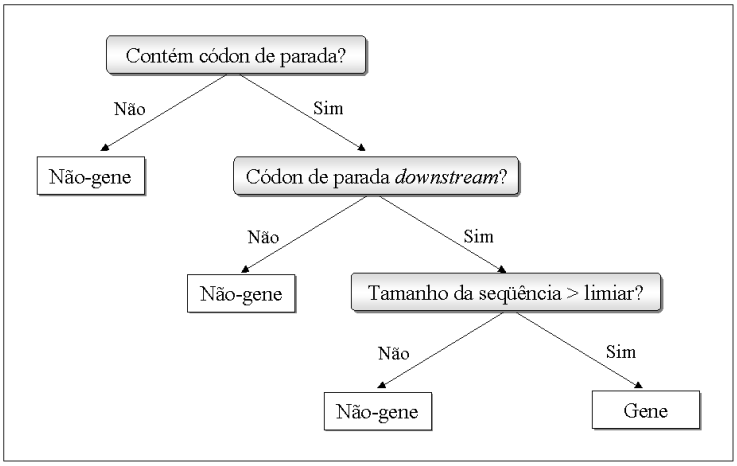

Figura 9: Árvore de decisão simples

De maneira mais formal, uma AD encontra regras que recursivamente bifurcam o conjunto de dados a fim de produzir sub-conjuntos que sejam homogêneos intra subconjuntos e heterogêneos inter sub-conjuntos. O conteúdo desses sub-conjuntos pode ser descrito por um conjunto de regras que usa um ou mais atributos dos dados [Mitchell, 1997, Monard e Baranauskas, 2003b]. O Algoritmo 3 ilustra de maneira simplificada o funcionamento de um algoritmo indutor de AD, chamado ID3 [Mitchell, 1997].

Algoritmo 3 Algoritmo ID3 [Mitchell, 1997]

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

Uma vantagem das ADs sobre outras técnicas de AM, como as revisadas anteriormente, é que elas produzem modelos que podem ser facilmente interpretados por humanos [Mitchell, 1997, Monard e Baranauskas, 2003b]. Esta é uma característica importante, visto que especialistas humanos podem analisar um conjunto de regras aprendidas por uma AD e determinar se o modelo aprendido é plausível, dadas as restrições do mundo real. Na Biologia, ADs têm sido usadas em problemas de reconhecimento de padrões, por exemplo, encontrar sítios de *splicing* em genes, como será visto na Seção 4.

3.6. Técnicas de Agrupamento

Técnicas exploratórias podem prestar um grande auxílio à compreensão da natureza complexa das relações multivariadas. Dentre essas técnicas, incluem-se as construções gráficas e os algoritmos para agrupar padrões. O agrupamento, em particular, é feito com base em similaridades ou distâncias (dissimilaridades) entre os padrões.

Nesse contexto, a escolha da medida de similaridade, como também da técnica de agrupamento, é em geral subjetiva [Brazma e Vilo, 2000, Slonim, 2002, Costa et al., 2003]. Os critérios de escolha devem levar em consideração a natureza da variável (discreta, contínua ou binária), as escalas de medida (nominal, ordinal ou intervalar) e o tipo do problema investigado.

Dentre as medidas de similaridade mais difundidas na literatura de análise de dados de expressão gênica, destacam-se a distância euclidiana e o coeficiente de Pearson. Para o mesmo problema, as técnicas de agrupamento mais difundidas são o agrupamento hierárquico, o k-médias e as redes neurais do tipo mapa auto-organizável [Alizadeh et al., 2000, Brazma e Vilo, 2000, Eisen et al., 1998, Costa et al., 2003, Michaels et al., 1998, Tamayo et al., 1999, Toronen et al., 1999, Wen et al., 1998], que são introduzidas a seguir.

Algoritmos de Agrupamento Hierárquico

Algoritmos de agrupamento hierárquico comprendem uma técnica bastante familiar para a maioria dos biólogos, por causa de seu uso na geração de árvores filogenéticas. Relacionamentos entre genes, por exemplo, são representados por uma árvore, ou **dendograma**, cujos comprimentos dos ramos refletem o grau de similaridade entre os genes (padrões de seqüências ou expressão gênica).

Esses relacionamentos são úteis porque eles podem representar graus variados de similaridades, além de requererem poucas suposições sobre a natureza dos dados [Eisen et al., 1998]. A árvore gerada pode ser usada para ordenar genes na tabela original de dados de tal forma a agrupar genes com expressões similares.

Entre as técnicas de agrupamento hierárquico, três variações são amplamente usadas: ligação máxima, média e simples. Essas variações estão associadas à maneira como a proximidade entre aglomerados é calculada [Jain e Dubes, 1988]. Neste tutorial é apresentada a técnica de agrupamento hierárquico implementando a ligação média ou UPGMA (do inglês *Unweighed Pair Group Method Average*). Essa variação foi escolhida por ser extensivamente usada na literatura de expressão gênica [Eisen et al., 1998, Alizadeh et al., 2000, Costa et al., 2003]. A técnica UPGMA também é usada para a resconstrução de árvores filogenéticas [Sneath e Sokal, 1973].

Quando a técnica UPGMA é utilizada, a proximidade entre dois aglomerados é calculada pela proximidade média entre entre os padrões de um grupo com os padrões de um outro grupo. Dado um conjunto de padrões E e uma medida de proximidade D, o Algoritmo 4, que implementa a técnica UPGMA, funciona da seguinte maneira [Jain e Dubes, 1988]:

Dependendo da variação utilizada, algoritmos de agrupamento hierárquico podem encontrar aglomerados não-isotrópicos, por exemplo, aglomerados em forma de cadeias bem separadas ou concêntricos [Jain et al., 1999]. Também é importante salientar que os métodos hierárquicos retornam hierarquias, e não partições. No entanto, as hierarquias podem ser transformadas em partições, por exemplo, cortando o dendograma gerado em um certo nível, como ilustrado na Figura 10. Nessa figura, duas linhas pontilhadas repre-

Algoritmo 4 Algoritmo UPGMA [Jain e Dubes, 1988]

1: Calcular matriz de proximidade, contendo as proximidades entre todos os padrões;
2: numero_aglomerados = numero_padroes;
3: **enquanto** numero_aglomerados > 1 **faça**
4: Selecionar o par de aglomerados mais similar;
5: Combinar os aglomerados selecionados um único aglomerado;
6: Atualizar a matriz de proximidade, recalculando as proximidades do novo aglomerado formado pela fusão;
7: **fim-enquanto**

sentam, respectivamente, cortes com três e quatro aglomerados.

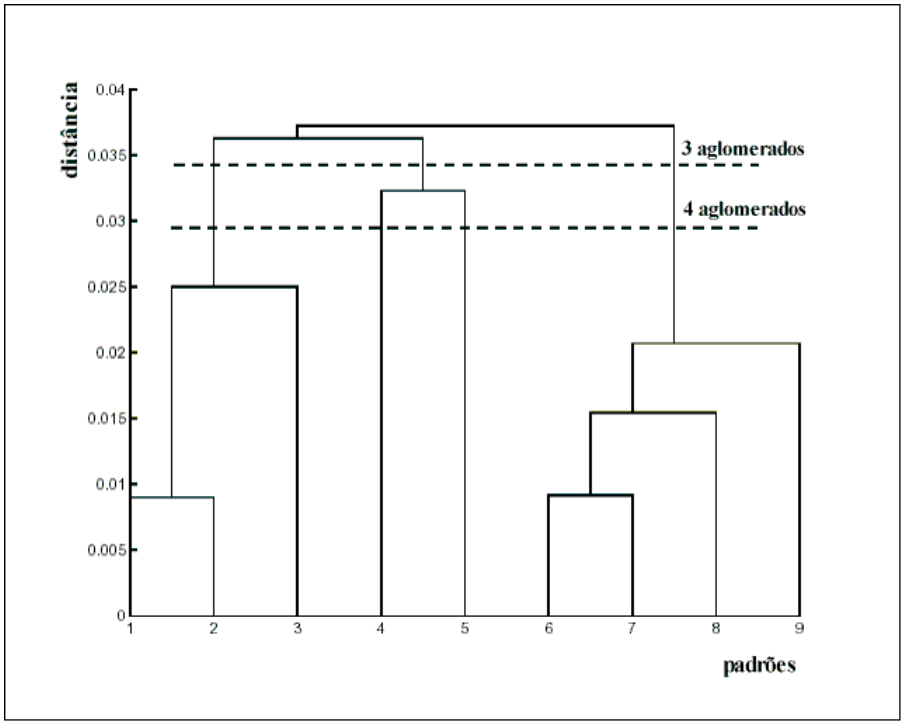

Figura 10: Exemplo de dois cortes em um dendograma com nove objetos

Apesar de sua ampla utilização, os métodos hierárquicos possuem uma série de deficiências quando aplicados ao estudo de dados de expressão gênica. Árvores hierárquicas não são projetadas para refletir as maneiras distintas em que padrões de expressão gênica podem ser similares - esse problema torna-se mais evidente na medidas que o conjunto de dados cresce [Tamayo et al., 1999]. Além disso, esses métodos são determinísticos. Portanto, padrões só podem ser agrupados baseando-se em decisões locais, as quais, uma vez tomadas, não podem ser re-avaliadas. Como conseqüência, esses métodos não são robustos a ruído [Mangiameli et al., 1996].

k-médias

Quando na análise de dados de expressão gênica há informações sobre o número de aglomerados que devem ser gerados, o algoritmo k-médias é uma boa alternativa às técnicas de agrupamento hierárquico [Tavazoie et al., 1999]. No algoritmo ***k*-médias**, objetos são particionados em um número fixo (k) de aglomerados. Não são geradas hierarquias, mas essas poderiam ser produzidas usando o resultado do k-médias como entrada para uma técnica de agrupamento hierárquico.

Uma deficiência do algoritmo k-médias é sua dependência da escolha de k e sua sensibilidade à escolha da partição inicial, essa última podendo levá-lo a ficar preso em mínimos locais. O algoritmo k-médias se comporta melhor com dados que contenham aglomerados esféricos. Portanto, aglomerados com outra geometria podem não ser encontrados [Jain et al., 1999].

Intuitivamente, o algoritmo k-médias funciona da seguinte maneira [Quackenbush, 2001]. Após atribuir aleatoriamente os padrões do conjunto de treinamento a cada um dos k aglomerados (k é especificado pelo usuário), um conjunto de vetores, chamados protótipos, contendo a média dos vetores (padrões) pertencentes a cada aglomerado é calculado. Esses protótipos são utilizados para calcular as distâncias entre os aglomerados.

Os padrões são iterativamente deslocados entre aglomerados, de acordo com as distâncias intra- e inter- aglomerados. Padrões passam para um novo aglomerado se estiverem mais próximos desse novo aglomerado do que do seu aglomerado atual. Após cada deslocamento, os protótipos para cada aglomerado são recalculados. O processo continua até que não haja mudança no conjunto de protótipos calculados - Figura 11.

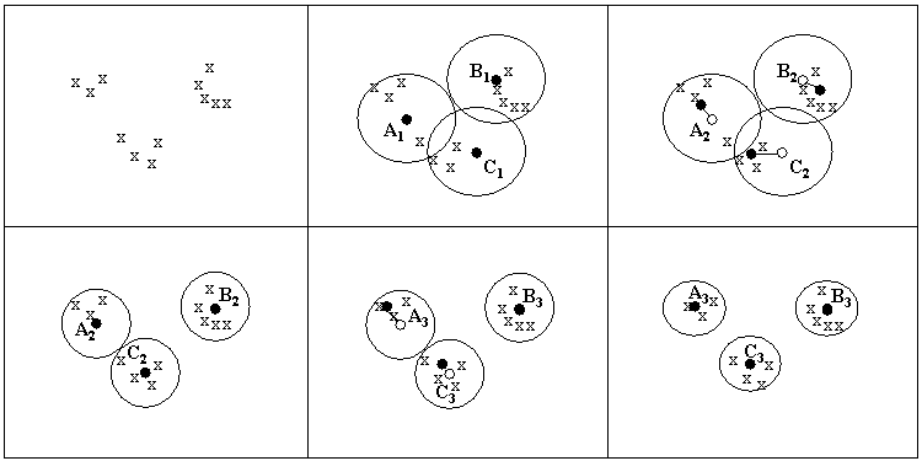

Figura 11: Estágios sucessivos do algoritmo *k*-médias

A seguir é apresentado uma versão simplificada do algoritmo k-médias - Algoritmo 5:

Algoritmo 5 Algoritmo *k*-médias [Jain e Dubes, 1988]

1: Escolher aleatoriamente k padrões do conjunto de treinamento para serem os k protótipos iniciais;
2: **enquanto** protótipos mudarem de valor **faça**
3: Usar os protótipos para agrupar os padrões nos aglomerados atuais;
4: **para todo** cada agrupamento $i, i = 1, 2, ..., k$ **faça**
5: Substituir protótipo $\mathbf{p}_i$ pela média de todos os padrões do i-ésimo aglomerado;
6: **fim-para**
7: **fim-enquanto**

Redes Neurais do tipo Mapa Auto-Organizáveis

Redes neurais do tipo **Mapa Auto-Organizável** (SOM - do inglês *Self-Organizing Map*) são redes apropriadas para aprendizado não supervisionado [Kohonen, 1997]. Elas combinam aprendizado competitivo com redução de dimensionalidade por meio da suavização dos aglomerados com respeito a uma grade (mapa) definida *a priori* - Figura 12.

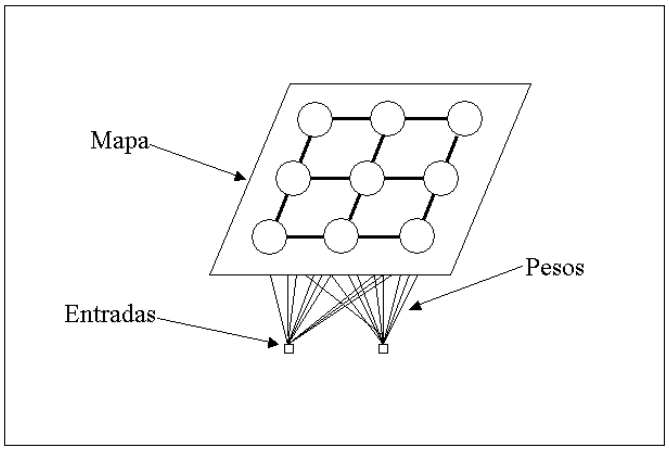

Figura 12: Exemplo de um topologia SOM 3x3 com duas entradas

Uma das principais características dessas redes é a propriedade de ordenação topológica dos aglomerados gerados. Ao contrário dos métodos hierárquicos, redes SOM são consideradas robustas na manipulação de dados com ruído [Mangiameli et al., 1996]. Por outro lado, elas são sensíveis a definição dos parâmetros iniciais, podendo ficar presas em mínimos locais.

De maneira intuitiva, o algoritmo treinamento das redes SOM funciona como segue. Inicialmente, escolhe-se uma grade (mapa) de nodos, por exemplo, um mapa 3×3. Cada elemento do vetor de entrada representa um atributo, por exemplo, o nível de expressão gênica de um dado gene. Todos os nodos do mapa são ligados aos atributos de entrada por meio de conexões ponderadas.

Os pesos são inicialmente escolhidos aleatoriamente e depois ajustados de forma iterativa. Cada iteração envolve a escolha aleatória de um padrão[4] do conjunto de treinamento, seguida de uma competição entre os nodos do mapa. Nessa competição, aquele nodo cujo vetor de pesos seja, de acordo com uma distância D, mais próximo ao vetor de entrada tem seus pesos (e dos nodos na vizinhança) ajustados na direção dos valores deste padrão de entrada. Assim, redes SOM impõem uma estrutura sobre os dados, em que nodos vizinhos tendem a definir aglomerados similares. A seguir, é apresentado o Algoritmo 6, uma versão simplificada do algoritmo utilizado para o treinamento de redes SOM.

Algoritmo 6 Algoritmo SOM [Haykin, 1999]

```
1:  Inicializar pesos da rede com valores aleatórios;
2:  Definir raio e taxa de aprendizado iniciais;
3:  enquanto ocorrerem mudanças significativas no mapa faça
4:      para todo padrão de entrada faça
5:          para todo cada nodo faça
6:              Calcular a distância entre padrão de entrada e pesos do nodo;
7:          fim-para
8:          Selecionar nodo n_k com menor distância;
9:          Atualizar pesos do nodo n_k e de seus nodos vizinhos;
10:         Reduzir taxa de aprendizado e raio;
11:     fim-para
12: fim-enquanto
```

[4]Um padrão pode ser, por exemplo, um vetor m-dimensional representando o nível de expressão gênica de m genes.

As redes SOM são apropriadas para análise exploratória dos dados quando não há informação *a priori* sobre sua distribuição. Um problema com essa técnica é o grande número de parâmetros ajustáveis, que incluem a topologia, a taxa de aprendizado, a função de vizinhança, o raio de vizinhança, entre outros. O sucesso do mapa é dependente desses parâmetros [Kohonen, 1997].

4. Reconhecimento de Genes

A identificação de genes em seqüências de DNA constitui uma tarefa muito custosa se realizada por meios laboratoriais. A obtenção de um procedimento (algoritmo) que automatize esta tarefa também é inviável, pois o conhecimento acerca da natureza dos genes ainda é incompleto. Há muitas variações na forma como os genes se apresentam, o que torna seu reconhecimento um problema complexo. O uso de técnicas de AM, capazes de extrair descrições de genes automaticamente a partir de amostras de dados conhecidas, se mostra uma alternativa adequada, e tem sido largamente explorada [Craven e Shavlik, 1994, Pavlidis et al., 2001, Shavlik, 1991, Uberbacher et al., 1993, Xu et al., 1996].

No reconhecimento de genes por meios computacionais, duas abordagens de busca são usualmente empregadas: por sinal e por conteúdo [Craven e Shavlik, 1994]. Essas abordagens diferem nas características das seqüências em que se concentram. A **busca por sinal** envolve a localização de sítios presentes na molécula de DNA que participam do processo de expressão gênica. Na **busca por conteúdo** procura-se padrões nas seqüências genômicas que indiquem a presença de um gene. Para obter melhores resultados, essas duas abordagens são usualmente aplicadas em conjunto. A seguir, essas considerações são apresentadas em maiores detalhes.

4.1. Busca por Sinal

Na busca por sinal localiza-se os genes indiretamente, por meio da identificação de sinais associados ao processo de expressão gênica. Procura-se regiões do DNA (ou mRNA) que desempenhem alguma função específica durante a expressão. O reconhecimento desses sítios possibilita um maior entendimento das funções biológicas executadas em uma célula durante a expressão gênica. A detecção de sinais também é importante no auxílio à compreensão dos mecanismos regulatórios da expressão de um gene, uma vez que muitos sinais possuem função regulatória. Alguns sinais indicam, por exemplo, situações como a velocidade de expressão do gene e condições para que esta ocorra.

Muitos desses sítios apresentam regiões de consenso em suas seqüências. A localização dos mesmos por meio dessa propriedade, porém, se mostra uma técnica muito simples e imprecisa. Existem diversas variações nos consensos identificados. Na tentativa de acomodá-los, obtém-se uma taxa elevada de falsos positivos [Stormo, 2000].

Uma outra alternativa comumente aplicada na identificação de sinais consiste em construir uma **Matriz de Posições Ponderadas** (no inglês, *weighted position matrix*) (MPP) [Staden, 1984]. Por meio desse método, determina-se um modelo para o sinal. No entanto, sua solução é limitada pelo fato de se considerar que posições adjacentes nas cadeias são independentes estatisticamente. Existem variações nessa técnica que consideram a dependência entre nucleotídeos, o que normalmente é realizado por meio da introdução de modelos Markovianos [Zhang e Marr, 1993].

A tarefa de reconhecimento de genes pode também ser formulada como um problema de classificação e ser então solucionada por uma técnica de AM [Craven e Shavlik, 1994]. Dada uma janela de tamanho l em uma seqüência de DNA, determina-se se esta contém um sinal de interesse em uma posição particular da cadeia.

Em geral, a escolha do tamanho da janela influencia o desempenho das técnicas de AM aplicadas [Craven e Shavlik, 1993a]. Grande parte dos trabalhos nesta área testam diversos valores de l na busca pelo melhor valor. Gerado o classificador, ele pode ser utilizado na localização de sinais movendo-se a janela por toda seqüência. A Figura 13 ilustra este processo.

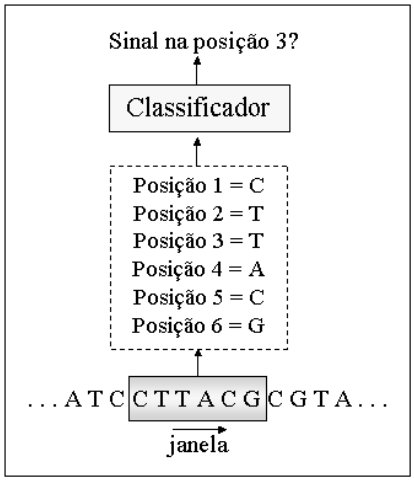

Figura 13: Busca por sinais como uma tarefa de classificação

Para o aprendizado do classificador, as instâncias do conjunto de treinamento que possuem o sítio de interesse devem ser alinhadas de forma a contê-lo em uma mesma posição da seqüência (geralmente seu centro). Freqüentemente, uma parte dos exemplos que não possuem o sinal procurado (exemplos negativos) também são alinhados de forma a ter na mesma posição uma região similar à típica do sinal . Este procedimento é realizado para evitar que o classificador gerado aprenda a diferenciar os sinais de forma trivial, como, por exemplo, unicamente por consensos presentes em suas seqüências.

Como exemplos de sinais tem-se os sítios de início de tradução, as regiões promotoras e os sítios de *splicing*. A seguir, os problemas de busca desses sítios são formalizados e soluções por meio de técnicas de AM são apresentadas.

Problema 4.1 *Identificação de sítios de início de tradução.*
Dado: Conjunto de seqüências de DNA (ou mRNA) de tamanho fixo com sítios de início de tradução identificados e conhecidos, além de cadeias sem a presença deste sinal.
Faça: Gerar um classificador capaz de identificar se uma janela de tamanho fixo de uma seqüência de DNA (ou mRNA) possui ou não um sítio de início de tradução.

Uma das primeiras aplicações de AM em Bioinformática se deu no reconhecimento de sítios de início de tradução (SITs) em seqüências de DNA da bactéria *E. coli* [Stormo et al., 1982]. 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 14. 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.

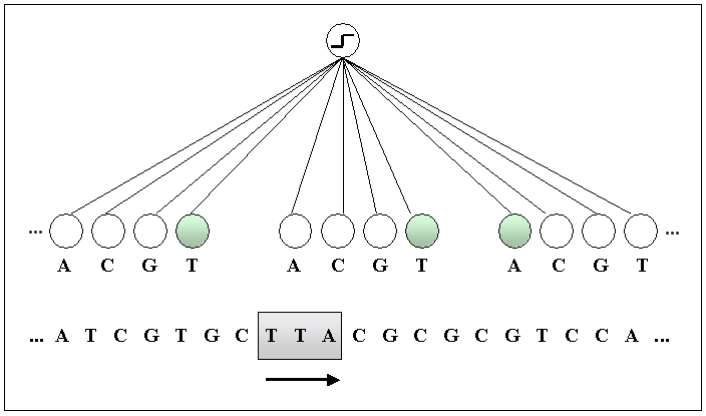

Figura 14: Esquema de parte de Perceptron utilizado no reconhecimento de sítios de início de tradução do DNA de bactérias *E. coli* [Craven e Shavlik, 1994]

Segundo [Stormo et al., 1982], a matriz obtida foi mais precisa que diversos métodos de consenso existentes na época. Verificou-se também que o uso da rede Perceptron fez com que pesos mais significativos correspondessem àqueles conectados às unidades de entrada representando o SIT e a região *Shine-Dalgarno*, confirmando as suspeitas iniciais dos autores.

Deve-se destacar, porém, que 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].

Em um estudo realizado por [Zien et al., 2000], SVMs foram aplicadas no reconhecimento de SITs de organismos vertebrados. O desempenho das SVMs foi comparado ao de RNs, obtidas por [Pedersen e Nielsen, 1997], e com um método probabilístico Markoviano de [Salzberg, 1997]. Resultados melhores são 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 con-

sideradas 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, [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 na aplicação considerada.

Problema 4.2 *Identificação de promotores.*
Dado: Conjunto de seqüências de DNA de tamanho fixo com regiões promotoras conhecidas e seqüências sem a presença desse sítio.
Faça: Gerar um classificador capaz de identificar se uma janela de tamanho fixo de uma seqüência de DNA possui ou não um promotor.

O início da transcrição gênica se dá com a ligação de uma molécula de RNA polimerase a uma região do DNA denominada promotora. As regras utilizadas pela RNA polimerase na identificação dos promotores ainda não são completamente conhecidas, o que dificulta a detecção dessas regiões em laboratórios.

[Towell et al., 1990] aplicaram uma abordagem híbrida de RNs e regras simbólicas na identificação de promotores em seqüências de bactérias *E. coli*. A rede empregada, denominada KBANN (*Knowledge Based Neural Network*), utiliza regras proposicionais formuladas por um biólogo (conhecimento a priori) na determinação da topologia e pesos iniciais da RN. As regras utilizadas identificavam dois conjuntos de padrões consenso em promotores procariotos e outras regiões cuja significância é controversa. As regiões consenso correspodem ao *TATA box* (região rica nos nucleotídeos T e A, com cadeia típica TAAATTA) e a seqüência TTGACA, que se encontram aproximadamente 10 e 35 nucleotídeos a montante (*upstream*) do SIT procarioto, respectivamente.

Por meio deste procedimento, os autores verificaram uma redução no tempo de treinamento das RNs, assim como uma melhora na generalização das redes. É interessante mencionar que as RNs obtidas aprenderam a descartar as regras que correspondiam a regiões controversas, indicando que estas não representam aspectos salientes dos promotores. Além disso, as regras produzidas falharam no reconhecimento de todas instâncias com promotores quando utilizadas individualmente.

No treinamento do classificador, as instâncias com promotores foram alinhadas de forma que a região promotora ficasse sete nucleotídeos à direita da janela, a qual possuía 57 nucleotídeos. A codificação dos nucleotídeos para a RN se deu de forma canônica de quatro bits, similar à discutida no Problema 4.1. Nos experimentos conduzidos pelos autores, os resultados obtidos pela rede KBANN foram comparados aos de uma rede MLP, de uma AD induzida pelo algoritmo ID3, do algoritmo k-vizinhos mais próximos[5] (k-NN, do inglês *k-nearest neighbor*) e de uma técnica referenciada na literatura biológica [O'Neill, 1989]. A técnica de "O'Neill", baseada em buscas de consensos, representa o método mais referenciado na literatura biológica para a identificação de regiões promotoras.

As RNs se sobressaíram em relação à técnica para reconhecimento de promotores referenciada na literatura biológica, evidenciando a eficácia de técnicas de AM na solução deste problema. O desempenho dos algoritmos k-NN e ID3, porém, foram inferiores, o que pode ser conseqüência da dificuldade dessas técnicas em lidar com dados com muitos atributos, como os da aplicação considerada.

[5]Esse algoritmo realiza a classificação de novos padrões de acordo com a classe de seus k vizinhos mais próximos [Mitchell, 1997].

Em [Reese e Eeckman, 1995], uma combinação de RNs foi aplicada no reconhecimento de promotores vertebrados. A identificação de promotores eucariotos pode ser considerada mais custosa e complexa, uma vez que estes possuem diversos e variados sítios de contato com proteínas que agem no início da transcrição gênica.

No trabalho realizado, foram utilizadas RNs individuais para a identificação de duas regiões comumente presentes nos promotores eucariotos, a *TATA-box* e uma cadeia denominada Iniciadora. As RNs foram treinadas com um procedimento de poda de conexões [Reed, 1993].

Na combinação das RNs geradas, foi utilizada uma rede do tipo *Time Delay Neural Network* (TDNN) [Haykin, 1999]. Nos experimentos conduzidos, com uma janela de 51 nucleotídeos, os resultados das TDNNs foram comparados aos das RNs individuais. Estas se mostraram pouco acuradas se utilizadas individualmente. A combinação pela TDNNs gerou ganhos significativos em acurácia, além da redução da taxa de falsos positivos nas predições.

Problema 4.3 *Identificação de sítios de splicing.*
Dado: Conjunto de seqüências de DNA (de organismos eucariotos) de tamanho fixo com fronteiras do tipo intron/exon, exon/intron, e sem nenhum desses sítios.
Faça: Gerar um classificador capaz de determinar se uma janela de tamanho fixo de uma seqüência de DNA possui uma fronteira intron-exon, exon-intron, ou nenhuma delas.

Em organismos eucariotos, o reconhecimento completo dos genes envolve também a identificação de suas regiões que codificam proteínas, os exons, e dos introns, porções de DNA que intermediam os exons e não produzem proteínas[6]. O reconhecimento das fronteiras entre esses elementos possui grande importância, uma vez que é necessário demarcar precisamente os segmentos de DNA que são efetivamente traduzidos em proteínas daqueles que não são.

Uma terminologia utilizada neste domínio referencia as bordas entre exons e introns como regiões "doadoras" e as bordas intron/exon como regiões "receptoras" [Alberts et al., 1997]. Na identificação de doadores e receptores, comumente são utilizadas evidências reportadas em estudos da Biologia. Ambas as regiões caracterizam-se por possuir um par de nucleotídeos que se conservam no intron da junção, sendo GT para doadores e AG para receptores. Isto não significa, porém, que estes mesmos pares não estejam presentes nos exons, o que dificulta sua utilização para a classificação dessas regiões.

[Lapedes et al., 1989] utilizaram ADs, RNs e k-NN (com medida de distância ponderada de acordo com cada posição da janela) na classificação dessas regiões. Foram utilizadas janelas de 11, 21 e 41 nucleotídeos. As instâncias de treinamento foram alinhadas de forma que a região de *splicing* estivesse no centro da janela. Padrões sem a região de *splicing* também foram alinhados de forma a ter os nucleotídeos AG ou GT em seu centro, prevenindo o classificador de usar regras do tipo "se AG no centro então receptor" na diferenciação das junções.

Foram gerados classificadores diferentes para o reconhecimento de regiões doadoras e receptoras, respectivamente. Observou-se que as RNs produziram os melhores resultados, com 91% de acurácia para as regiões receptoras e 95% para as doadoras. 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 de um ponto de vista biológico. Uma

[6]Na definição apresentada, não está sendo considerada a ocorrência de *splicing* alternativo.

ilustração simplificada de uma AD para a aplicação de determinar regiões doadoras pode ser visualizada na Figura 15.

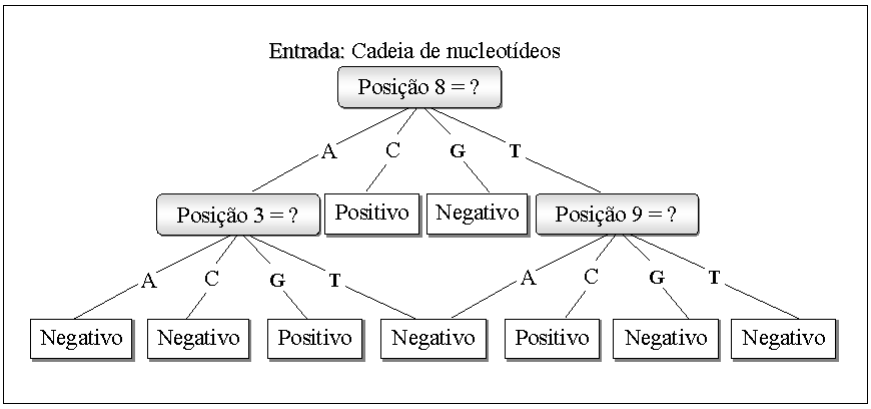

Figura 15: Ilustração de uma AD para a identificação de regiões doadoras em seqüências de DNA [Craven e Shavlik, 1994]

Em [Rampone, 1998], uma abordagem híbrida envolvendo o uso de regras e de uma RN foi utilizada no reconhecimento de sítios de *splicing* de primatas. O algoritmo obtido, denominado BRAIN (*Batch Relevance-based Artificial INtelligence*), infere fórmulas Booleanas dos exemplos, na forma de regras disjuntivas. Estas são então refinadas por uma RN e combinadas com um procedimento discriminante estatístico. No trabalho realizado, [Rampone, 1998] comparou seus resultados aos previamente reportados pelo projeto StatLog [Michie et al., 1994], que envolveu a avaliação de diversas técnicas de AM sobre um mesmo conjunto de dados. Entre os algoritmos de AM comparados, encontram-se uma RN do tipo RBF (do inglês *Radial Basis Function*), um classificador Bayesiano, uma RN do tipo MLP, o algoritmo C4.5, indutor de ADs, e o algoritmo k-NN. Verifica-se, de forma geral, uma maior acurácia dos modelos baseados em RNs.

Outros trabalhos envolvendo o reconhecimento de junções de *splicing* por meio de técnicas de AM foram realizados em [Lorena et al., 2002b, Lorena et al., 2002a, Lorena e de Carvalho, 2003]. Em [Lorena et al., 2002b], SVMs e ADs foram aplicadas no reconhecimento dessas junções. Os melhores resultados foram obtidos pelas SVMs, com significância estatística de 95%. A maior acurácia das SVMs deve-se, em grande parte, à sua robustez diante de dados com muitos atributos. Investigou-se também o efeito da aplicação de uma fase de pré-processamento visando eliminar ruídos dos dados considerados. A aplicação desta fase levou a simplificações nos modelos induzidos. No caso das SVMs, em alguns casos houve também melhora de desempenho. Para as ADs, a principal contribuição se deu em diminuições no tamanho das árvores induzidas, o que indica ganhos em termos de compreensibilidade dos modelos gerados.

4.2. Busca por Conteúdo

Na busca por conteúdo os genes são identificados por meio do reconhecimento de padrões gerais que ocorrem em regiões codificadoras. Os objetivos, nesse caso, são diferenciar regiões codificadoras de cadeias não codificadoras e identificar a fase aberta de leitura das seqüências codificantes.

A **fase de leitura** de um gene corresponde a como seus nucleotídeos são agrupados em códons. Seja, por exemplo, a seqüência da Figura 16. Há três possíveis combinações em triplas, ou seja, três fases de leitura. No caso de moléculas de DNA, tem-se seis fases de leitura, três para cada uma de suas fitas. A **fase aberta de leitura** ou **ORF** (do inglês, *Open Reading Frame*) corresponde a combinação de códons que codificam uma proteína em potencial, não possuindo códons de parada [Alberts et al., 1997].

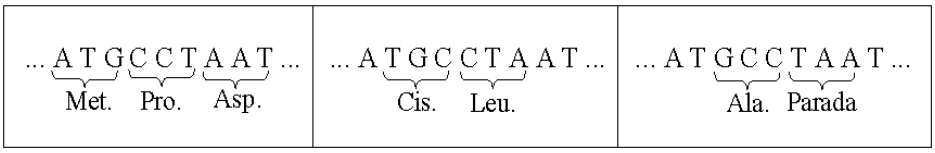

Figura 16: Fases de leitura de uma seqüência de nucleotídeos

No reconhecimento de regiões codificadoras, várias propriedades típicas dessas cadeias podem ser exploradas. Uma propriedade trivial é o fato de que as regiões codificadoras devem apresentar uma organização em forma de códons. Outras características que exercem influência na composição de regiões codificadoras, diferenciando-as das não codificadoras, são [Craven e Shavlik, 1994]:

- alguns aminoácidos estão mais presentes na composição de proteínas que outros;
- códons que mapeam para um mesmo aminoácido não são utilizados igualmente por todos organismos. Existe, em geral, uma preferência de códons para cada organismo/espécie;
- a forma das proteínas é parcialmente determinada por interações eletrostáticas entre aminoácidos vizinhos. Por este motivo, alguns aminoácidos têm maior "afinidade" em serem vizinhos.

Técnicas estatísticas e discriminantes aplicam, em geral, um conjunto de medidas visando capturar essas e outras características das regiões codificadoras [Fickett, 1996]. No uso de técnicas de AM, formula-se o problema como uma tarefa de classificação, de forma semelhante à apresentada no caso da busca por sinais. Dada uma janela de tamanho fixo em uma seqüência de DNA, treina-se um classificador de forma a diferenciar regiões codificadoras (e identificar sua ORF) de não codificadoras. A seguir, aplicações envolvendo o uso de AM em abordagens de busca por conteúdo são descritas.

Problema 4.4 *Identificação de regiões codificadoras.*
Dado: Conjunto de seqüências de DNA de tamanho fixo com regiões codificadoras e não codificadoras.
Faça: Gerar um classificador capaz de reconhecer se uma janela de tamanho fixo de uma seqüência de DNA é ou não codificadora. Caso seja, identificar também sua ORF.

O problema dado pode ser ainda diferenciado entre organismos procariotos e eucariotos. Em procariotos, deve-se basicamente distinguir os genes das regiões não codificadoras que os intermediam. No caso de eucariotos, a presença de introns faz com que se torne necessário distinguir também exons de introns.

Em [Farber et al., 1992], redes Perceptron com ativação sigmoidal foram aplicadas na distinção entre exons e introns. Os resultados dessa rede foram comparados a um classificador Bayesiano baseado em preferências de códons desenvolvido por [Staden e McLachlan, 1982]. É reportada uma maior acurácia das RNs, sendo esse resultado atribuído ao fato do classificador Bayesiano assumir uma independência estatística entre códons vizinhos, ao contrário das RNs, que não possuem esta limitação.

Os autores também investigaram a utilização de diferentes codificações para as entradas da rede. A primeira consistiu em utilizar os nucleotídeos da seqüência em uma forma canônica. Na segunda utilizou-se a freqüência dos 64 possíveis códons na janela submetida. Cada entrada representava então o número de vezes que o códon representado ocorria na janela. A terceira codificação envolveu uma contagem dos dicódons presentes na janela (pares adjacentes de códons), o que elevou o número de entradas para 4096. Foram utilizadas janelas de 5 a 90 códons nos experimentos. Janelas maiores levaram em geral a melhores predições.

A utilização da representação por dicódons melhorou a generalização obtida pela RN. É interessante observar que o desempenho obtido pela RN com o uso da representação de apenas um códon foi inferior mesmo adicionando à rede uma camada intermediária com diferentes números de nodos. Este fato confirma que a habilidade de um sistema de aprendizado é dependente da representação dos atributos do problema [Craven e Shavlik, 1994]. Resultados e discussões semelhantes são apresentados em [Craven e Shavlik, 1993b].

Em [Uberbacher e Mural, 1991] também é apresentado um sistema para distinção entre exons e introns utilizando uma representação alternativa para os atributos. As seqüências, em janelas de 99 nucleotídeos, foram submetidas a sete algoritmos (denominados sensores), que avaliavam diferentes características da cadeia de DNA. Algumas das propriedades medidas foram a freqüência com que cada nucleotídeo ocupa cada posição nos códons e preferências verificadas em tuplas de seis nucleotídeos [Roberts et al., 1995]. Esses sensores fornecem medidas do "poder de codificação" da subseqüência sendo analisada. Uma RN foi então empregada na combinação dessas informações, atribuindo pesos a cada uma delas.

No teste desse sistemas com 19 genes humanos, foram localizados 90% de exons longos (com mais de 100 nucleotídeos). A RN gerada constitui parte de um servidor para identificação de genes em seqüências de DNA denominado GRAIL (*Gene Recognition and Assembly Internet Link*) [Uberbacher et al., 1993].

Após identificadas as regiões codificadoras nos sistemas anteriormente descritos, pode-se então verificar qual sua ORF.

No caso de organismos procariotos, uma abordagem envolvendo o uso de RNs foi proposta por [Craven e Shavlik, 1993a], para bactérias *E. coli*. O principal objetivo neste caso foi avaliar a previsão de ORFs. É argumentado que, para bactérias *E. coli*, este é um problema mais crucial, uma vez que grande parte de seu genoma é codificante.

Os resultados reportados foram comparados aos métodos Bayesianos de [Gribskov et al., 1984] e [Staden, 1990], baseados em preferências de códons. A RN foi treinada de forma a predizer a posição do códon que o nucleotídeo no centro da seqüência ocupava. Ela possuía, portanto, seis saídas, representando as posições 1, 2 e 3 na fita submetida e 4, 5 e 6 para o caso da fita complementar.

Os autores também investigaram diferentes formas de codificação para as entradas da RN. As representações utilizadas foram a de nucleotídeos na forma canônica, a de contagem de freqüência de códons na janela e o uso de medidas similares às de [Uberbacher e Mural, 1991], adaptadas para organismos procariotos. Também foi utilizada uma combinação das probabilidades providas pelo método de [Staden, 1990] com as medidas adaptadas de [Uberbacher e Mural, 1991]. Foram empregadas janelas 61 nuceotídeos.

Os resultados obtidos foram avaliados em termos da porcentagem de janelas para as quais gerou-se uma predição de ORF correta. Foi verificado um maior poder preditivo das abordagens envolvendo manipulações nos atributos de entrada. Esse fator confirma a indicação de que a representação das entradas da RN tem papel crucial no desempenho do modelo gerado.

4.3. Combinação de Métodos

Em geral os sistemas de identificação de genes existentes não se baseiam em buscas de sinais ou de conteúdo exclusivamente. Abordagens mais promissoras envolvem a combinação dessas duas estratégias de busca (por exemplo, GRAIL II [Xu et al., 1996],

GeneID [Guigó et al., 1992], GeneParser2 [Snyder e Stormo, 1995a]). Alguns sistemas também utilizam buscas por similiridade para confirmar suas previsões (por exemplo, GeneID+ [Burset e Guigó, 1996], GeneParser3 [Snyder e Stormo, 1995b]). Nesse caso, as estruturas gênicas identificadas são traduzidas em possíveis cadeias de aminoácidos, as quais são comparadas com seqüências em bases protéicas e pontuadas de acordo com sua similaridade com alguma cadeia conhecida. Mesmo se o gene identificado for novo, a similaridade entre os aminoácidos por ele gerados e aqueles presentes em outras proteínas permite detectar algum gene homólogo, auxiliando na inferência da função do gene desconhecido.

Em vários destes sistemas, técnicas de AM são empregadas em uma ou mais etapas da predição gênica. A predição da estrutura gênica é, porém, mais complexa e envolve a combinação de vários passos e técnicas. Seja, por exemplo, o sistema GRAIL II [Xu et al., 1996]. Sua operação pode ser dividida em quatro passos:

- **Passo 1:** *Geração de exons candidatos*. Encontra-se todos os possíveis exons identificando sítios doadores e receptores. A determinação desses sítios é realizada inicialmente por meio de métricas baseadas em consensos. Uma RN é então utilizada, atribuindo uma pontuação indicando se a junção identificada é um sítio verdadeiro ou falso. Um *pool* de exons candidatos é gerado de forma a satisfazer as restrições de possuir uma fase de leitura e ser "intermediado" por um par de junções receptoras e doadoras com pontuação acima de um limiar.
- **Passo 2:** *Eliminação de candidatos improváveis*. Neste passo, uma série de medidas e regras heurísticas derivadas do conhecimento biológico são aplicadas aos exons candidatos. Essas regras definem condições que um provável exon deve satisfazer. Sua aplicação leva à eliminação de grande parte dos exons candidatos (aproximadamente 90%).
- **Passo 3:** *Avaliação dos exons*. Os exons candidatos remanescentes são então avaliados por uma RN. Esta recebe como entradas treze medidas de avaliação do potencial de codificação e atribui a cada exon uma pontuação.
- **Passo 4:** *Geração do modelo do gene*. Nesta fase, um modelo de gene é obtido. Um algoritmo de programação dinâmica é aplicado na montagem do gene à partir dos exons candidatos, baseado em suas pontuações. Também são checadas se algumas restrições, como o fato de não poder haver sobreposição de exons e de que exons internos não poderem ter códons de parada, são satisfeitas.

Outros sistemas diferem basicamente nas técnicas empregadas e passos realizados para a obtenção da estrutura gênica.

Em um cuidadoso e extenso estudo, [Burset e Guigó, 1996] compararam diversos sistemas para predição da estrutura de genes eucariotos. Algumas deficiências comuns foram identificadas. A primeira diz respeito ao fato de que não há uma metodologia padrão na obtenção das acurácias fornecidas pelos autores de cada sistema. Nos experimentos realizados por [Burset e Guigó, 1996], as acurácias dos programas se mostraram menores que as originalmente reportadas. Outra deficiência é a de que a acurácia desses programas está intimamente ligada aos conjuntos de treinamento empregados em sua geração. Quando confrontados com seqüências com pouca similaridade com as utilizadas em seu treinamento, o desempenho torna-se pior. A acurácia dos sistemas avaliados também foi afetada substancialmente pela presença de ruídos nos dados. Esses resultados indicam pouca robustez diante de erros de sequenciamento.

Em seu estudo, [Burset e Guigó, 1996] também apontaram que o emprego de buscas por similaridade em bases protéicas mostra-se uma estratégia muito promissora. Também é sugerido que a combinação da saída de vários programas pode trazer be-

nefícios. A idéia é a de que, quando todos programas predizem um mesmo exon, este (quase certamente) pode ser considerado correto.

5. Análise de Dados de Expressão Gênica

Como visto na Seção 2.2, o advento da tecnologia de *microarray* de DNA, entre outras técnicas, vem propiciando aos biólogos a possibilidade de medir o nível de expressão de milhares de genes em um único experimento. Experimentos iniciais [Eisen et al., 1998] sugerem que genes de funções similares produzem padrões similares em experimentos de hibridização em *microarray*.

Na medida que dados desses experimentos forem se acumulando, será essencial o desenvolvimento de meios precisos tanto para a extração do significado biológico desses dados como também para a atribuição de funções aos genes. Embora o enfoque desta seção seja principalmente em dados obtidos com *microrray* de DNA, as técnicas de AM a serem descritas podem também ser aplicadas a dados de expressão gerados com outras tecnologias, como o SAGE [Velculescu et al., 1995], desde que eles sejam apresentados no formato adequado.

5.1. Aprendizado de Máquina e Dados de Expressão

Quando um experimento com *microarray* é realizado, um scanner registra valores da intensidade da fluorescência - o nível de fluorescência em cada ponto do *array*. No caso de *arrays* de expressão gênica, normalmente haverá diversos experimentos medindo o mesmo conjunto de genes em várias circunstâncias. Por exemplo, pode-se medir a expressão de uma célula em condições normais, quando ela é aquecida ou resfriada, ou quando uma droga é adicionada. Também se pode medir a expressão em vários intervalos tempo - por exemplo, 5, 10, e 15 minutos após um antibiótico ser adicionado.

Do ponto de vista de AM, os valores de expressão medidos podem ser organizados de várias maneiras, como ilustrado na Figura 17 [Molla et al., 2003]. As figuras 17(a) e 17(c) mostram que cada gene pode ser visto com um exemplo (padrão), em que os níveis de expressão medidos nas várias condições representam cada uma das características ou atributos do padrão. Uma outra maneira de representação é considerar cada experimento como um exemplo (padrão), em que as características são os valores da expressão para todos os genes no *microarray* - Figuras 17(b) e 17(d).

Quando técnicas de AM não supervisionadas são utilizadas para análise de dados expressão gênica, elas, em geral, procuram aprender um agrupamento dos genes que seja funcionalmente significativo. Como será visto na Seção 5.2, genes podem ser agrupados usando técnicas como mapas auto-organizáveis [Tamayo et al., 1999] e agrupamento hierárquico [Eisen et al., 1998].

RNs, SVMs e outras técnicas de aprendizado supervisionado adotam uma abordagem oposta [Brown et al., 2000, Furey et al., 2000, Khan et al., 2001, Xu et al., 2002, Shipp et al., 2002]. Por exemplo, enquanto as técnicas não supervisionados determinam como um conjunto de genes se organiza em grupos funcionais, técnicas supervisionadas determinam que características da expressão de um determinado gene o leva a fazer parte de uma dada classificação funcional. Aplicações de técnicas de AM supervisionadas a dados de expressão gênica serão vistos na Seção 5.3.

Uma questão fundamental, tanto para técnicas supervisionadas, quanto para não supervisionadas (embora mais nas primeiras) é que dados de *microarray* apresentam novos desafios aos algoritmos de AM, geralmente desenvolvidos para lidar com um grande

(a)

Características

Exemplos	Experimento 1	Experimento 2	...	Experimento N
Gene 1	1083	1464	...	1115
Gene 2	1585	398	...	511
...	...	...	...	...
Gene M	170	302	...	751

(b)

Características

Exemplos	Gene 1	Gene 2	...	Gene M
Experimento 1	1083	1464	...	1115
Experimento 2	1585	398	...	511
...	...	...	...	...
Experimento N	170	302	...	751

(c)

Características

Exemplos	Experimento 1	Experimento 2	...	Experimento N	Categoria
Gene 1	1083	1464	...	1115	Y
Gene 2	1585	398	...	511	X
...	...	...	...	...	...
Gene M	170	302	...	751	X

(d)

Características

Exemplos	Gene 1	Gene 2	...	Gene M	Categoria
Experimento 1	1083	1464	...	1115	B
Experimento 2	1585	398	...	511	A
...	...	...	...	...	...
Experimento N	170	302	...	751	B

Figura 17: Maneiras diferentes de representar dados de *microarray* para AM

número de amostras (exemplos) com relativamente poucos atributos ou características [Mitchell, 1997, Slonim, 2002].

Em contraste, um experimento típico de *microarray* mede milhares de genes (que podem ser vistos como características), mas inclui apenas dezenas ou centenas de amostras. A maioria dos algoritmos falha quando são aplicados a problemas dessas dimensões devido, entre outros fatores, ao super-ajustamento (*overfitting*) aos dados de treinamento [Mitchell, 1997].

Um outro problema comum com dados de *microarray* é o elevado grau de ruído presente. A variância das medidas do *array* pode ser substancial, muitas das características podem apresentar valores ausentes e, ocasionalmente, exemplos de treinamento podem estar incorretamente rotulados. Tudo isso faz da predição de classe uma tarefa particularmente desafiadora [Slonim, 2002].

5.2. Descoberta de Classes

Existe uma série de estudos que aplicam técnicas de AM não supervisionadas a dados de expressão gênica. Cada um desses estudos investigam diferentes técnicas de análise de dados, medidas de similaridades e base de dados. Entre as técnicas de análise de dados empregadas nesses estudos estão: redes SOM, agrupamento hierárquico, análise de componentes principais, k-médias e CLICK [Tamayo et al., 1999, Toronen et al., 1999, Tavazoie et al., 1999, Costa et al., 2003, Eisen et al., 1998, Wen et al., 1998, Michaels et al., 1998, Raychaudhuri et al., 2000, Sharan e Shamir, 2002, Golub et al., 1999, Alizadeh et al., 2000].

Com relação aos dados utilizados, a grande maioria dos trabalhos publicados utiliza níveis de expressão da levedura (*Saccharomyces cerevisiae*) [Eisen et al., 1998, Cho et al., 1998, Tamayo et al., 1999, Raychaudhuri et al., 2000, Costa et al., 2003, Alizadeh et al., 2000]. Uma da razões para essa preferência é que esse organismo possui todos os seus genes conhecidos e com uma boa parte de suas funções descobertas. Existem também estudos com dados de ratos [Wen et al., 1998] e de humanos [Tamayo et al., 1999]. A seguir apresenta-se de maneira geral o problema abordado.

Problema 5.1 *[Molla et al., 2003] Agrupamento de genes baseado nas suas expressões gênicas.*
Dado: Conjunto de genes de um organismo representado como na Figura 17(a). Cada gene é um exemplo. As características de um exemplo são os valores numéricos dos níveis expressão sobre várias circunstâncias experimentais (estresse ambiental, estágio de desenvolvimento etc.).
Faça: Agrupe os genes baseado na similaridade de seus valores de expressão.

Um dos estudos pioneiros nessa área foi o de [Eisen et al., 1998]. Os autores agruparam padrões de expressão gênica de 2467 genes da levedura (de um total de 6200), que tinham anotação funcional. Os dados foram adquiridos por meio de *microarrays* de cDNA, consistindo no nível de expressão da levedura submetida a quatro situações distintas (divisão celular e respostas a diferentes tipos de estresse ambiental). Isto resultou em quatro séries temporais com um total de 79 instantes de tempo.

A expressão de cada gene foi calculada a partir da razão (*log ratio*) entre o sinal medido em um determinado instante de tempo (Cy5 - corante vermelho) e o sinal de controle (Cy3 - corante verde), que é normalmente o sinal do instante zero. Ao fim dessas transformações, tem-se sinais positivos indicando níveis de expressão maiores do que o normal, e sinais negativos (níveis de expressão inferiores ao normal). Nesse caso, utilizando o formato da Figura 17(a), cada gene é um padrão e os níveis de expressão medidos durante cada um dos experimentos são as características.

Com o objetivo de medir a similaridade entre os padrões, foi utilizado o coeficiente de correlação de Pearson [Jain e Dubes, 1988]. Essa medida dá ênfase na forma assumida pela série temporal, ao invés da magnitude de seu sinal (medidas como a distância euclidiana). A correlação de Pearson produz valores próximos a 1 para padrões com correlação positiva, e valores próximos a -1 para padrões com correlação negativa. Baseado nesse índice de distância, um algoritmo hierárquico que implementa a técnica UPGMA é aplicado, gerando um dendograma como resultado.

Ao final do processo, é aplicado um ordenamento não ótimo à árvore gerada, fazendo com que expressões similares fiquem acima de outras (a estrutura da árvore em si não indica qual a ordem das sub-árvores abaixo de um nó). Esse ordenamento é calculado a partir de uma medida de pesos dos genes, utilizando, por exemplo, o seu nível de expressão médio. Uma representação gráfica dessa árvore é fornecida como resultado do algoritmo, a qual contém ao seu lado os níveis de expressão através do tempo: expressões induzidas são representadas em vermelho, as suprimidas em verde e as não alteradas em preto, o que permite uma interpretação visual dos resultados.

A Figura 18 ilustra um exemplo desse tipo de representação, observando que o dendograma tem cores distintas para diferentes aglomerados encontrados. De fato, o dendograma da Figura 18 é similar à representação da tabela apresentada na Figura 17(a), em que valores inteiros são representados por intensidades em escala de cinza. Porém, ao contrário da Figura 17(a), os genes na Figura 18 estão ordenados por similaridade (genes mais similares com respeito a seus vetores de valores de expressão são agrupados juntos)

- para uma representação usando dados reais, ver [Eisen et al., 1998].

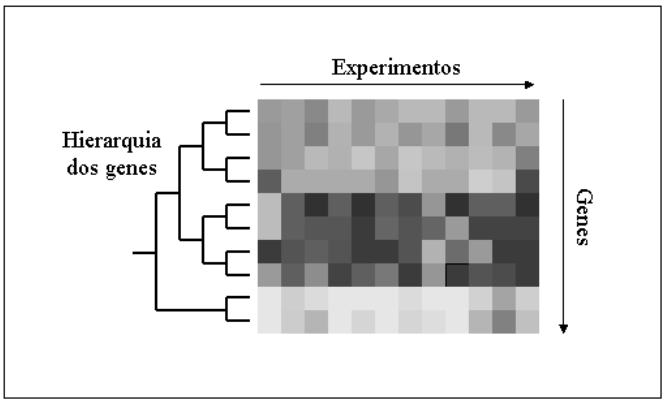

Figura 18: Representação gráfica dos resultados de um agrupamento hierárquico com dados de expressão gênica [Molla et al., 2003]

Um dos destaques desse estudo é a representação visual dos resultados que permite a identificação de aglomerados por uma simples inspeção visual. Por meio de observações não empíricas e não sistemáticas foi verificada a tendência de genes com seqüências similares ou com o mesmo papel em processos celulares se encontrarem em regiões próximas no dendograma. O algoritmo foi também testado com dados gerados de forma aleatória, para avaliar se os padrões na árvore eram gerados ao acaso ou através de algum viés do algoritmo, o que foi não observado nas três bases de dados aleatórios testadas.

Em outro trabalho relevante na área [Tamayo et al., 1999], redes SOM foram utilizadas para o agrupamento de genes. Essas redes não apresentam algumas limitações encontradas em técnicas de agrupamento hierárquico, como assumir a existência de uma descendência hierárquica nos indivíduos agrupados, algo não observado em genes de uma mesma espécie. Duas bases de dados foram utilizadas nesse trabalho, uma referente à formação de células sanguíneas de seres humanos e a outra relativa a células de levedura em dois ciclos celulares em 32 intervalos de tempo. Essa última já foi previamente estudada em [Cho et al., 1998].

Os dados presentes nas duas bases de dados foram coletados a partir de *Gene Chips* (*arrays* de oligonucleotídeos). Os valores das intensidades foram calculados a partir da média da intensidade de cada uma das vinte sondas de um gene. Foram excluídos genes cujo nível de expressão variava pouco com o tempo. Por fim, os valores de cada gene foram normalizados.

Redes SOM usam a distância euclidiana como medida de similaridade entre padrões, medida essa que não captura a similaridade da forma dos padrões temporais e sim de suas intensidades. Entretanto, como houve uma normalização dos dados, a distância euclidiana assume propriedades semelhantes ao coeficiente de correlação de Pearson [Dopazo et al., 2001].

Um programa de domínio público, chamado GENECLUSTER, foi desenvolvido para esse trabalho. Esse aplicativo apresenta os resultados dos experimentos por meio de uma interface gráfica, contendo a expressão média de cada um dos aglomerados, além da listagem dos genes de cada grupo. Para avaliação dos resultados alcançado pelo algoritmo, foi realizada uma comparação visual dos aglomerados obtidos com os dados de levedura com resultados já apresentados em um estudo realizado em [Cho et al., 1998], sendo encontradas grandes similaridades entre os aglomerados.

No caso dos dados do processo de formação de células sanguíneas foi observado, por inspeção visual (e não automática), uma grande afinidade funcional entre os genes dos

aglomerados encontrados. Porém, é importante salientar que esse processo de inspeção é bastante custoso do ponto de vista de tempo e sujeito à subjetividade.

Uma maneira de evitar esse problema é efetuar um agrupamento dos nodos da rede SOM, por meio do uso de um outro algoritmo de agrupamento (os pesos de cada nodo da rede SOM representariam o padrão de entrada para o outro algoritmo de agrupamento). Em [Vesanto e Alhoniemi, 2000, Costa et al., 2003], k-médias e agrupamento hierárquico são empregados para essa tarefa.

Em um trabalho semelhante, [Tavazoie et al., 1999] agruparam perfis de expressão dos 3000 genes mais variáveis da levedura, durante o ciclo celular (15 pontos - mesmos dados usados em [Cho et al., 1998]), em 30 aglomerados usando o k-médias. Eles encontraram, para metade dos aglomerados formados, que fortes padrões de seqüencia estão presentes nas seqüencias anteriores (*upstream*) aos genes.

A técnica k-médias também foi utilizada como parte de um método, proposto em [Brazma e Vilo, 2000], para identificação de supostos (*putative*) sinais regulatórios. Mais especificamente, os genes foram agrupados usando k-médias com a distância euclidiana. Ao invés de fixar o número k de aglomerados, ele foi variado de 2 a 100. Para cada k, o algoritmo foi repetido 10 vezes com inicializações diferentes de centros. No total, 900 partições separadas foram criadas, em que aglomerados de tamanhos entre 20 e 100 genes foram selecionados, totalizando 52100 aglomerados diferentes. Esses aglomerados foram usados como entrada para etapas subseqüentes do processo.

Técnicas de AM não supervisionadas também vêm sendo aplicadas à descoberta de novas classes doenças - Problema 5.2.

Problema 5.2 *[Molla et al., 2003] Descoberta de novas classes de doenças.*
Dado: Conjunto de experimentos de *microarray*, cada um realizado com células de pacientes diferentes (Figura 17(d)). Os pacientes têm um grupo de doenças bastante relacionadas entre si. Os níveis de expressão gênica de cada paciente de um experimento de *microarray* representam as características do padrão. A classificação da doença correspondente a cada paciente é chamada de categoria do paciente.
Faça: Usar algoritmos de agrupamento (sem considerar a categoria da doença) para encontrar células que não se ajustam bem em suas classes atuais. Assumir que essas células pertencem a novas classificações de doenças.

Duas questões básicas estão associadas ao problema de descoberta de classes (por exemplo, Problemas 5.1 e 5.2): além do desenvolvimento de algoritmos para efetuar o agrupamento dos tumores baseado nas expressões dos genes (ou o agrupamento dos próprios genes), existe ainda a necessidade de determinar se os aglomerados formados refletem uma verdadeira estrutura dos dados ou simplesmente um agrupamento aleatório.

Com o objetivo de analisar dados de tumores, [Golub et al., 1999] usaram redes SOM (mapas 2×1) para agrupar automaticamente 38 exemplos de dois tipos de leucemia bastante relacionadas, com base na expressão gênica de 6817 genes - leucemia mielóide aguda (AML - do inglês *Acute Myeloid Leukemia*) e leucemia aguda limfoblástica (ALL - do inglês *Acute Lymphoblastic Leukemia*). O conjunto de dados era formado por 11 amostras do tipo AML e 27 do tipo ALL. Os resultados obtidos mostram que a rede SOM foi eficiente, embora não tenha obtido uma acurácia de 100%, em descobrir automaticamente as duas categorias de leucemia (tumores) - Figura 19(a).

Nesse mesmo trabalho, os autores estenderam a descoberta de classes por meio de uma busca por sub-classes mais refinadas. Para isto, foi utilizada uma rede SOM para dividir os exemplos em quatro aglomerados (B1 a B4) - um mapa 4×1. Subseqüentemente, obtiveram dados de imuno-fenótipos das amostras, em que encontraram que os

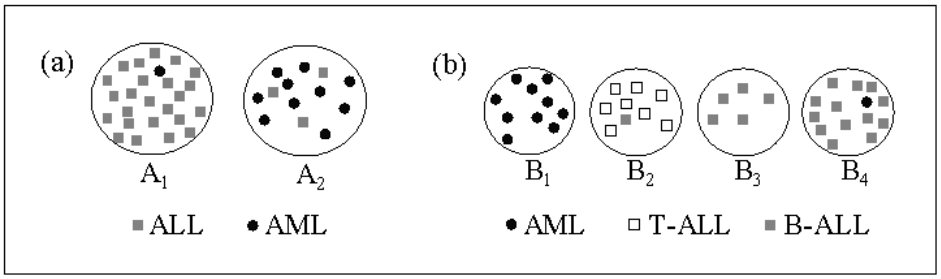

Figura 19: Descoberta de classes para ALL-AML [Golub et al., 1999]

quatro aglomerados correspondiam a AML, ALL - linhagem T, ALL - linhagem B e ALL - linhagem B, respectivamente - Figura 19(b).

Portanto, a abordagem de descoberta de classes usando redes SOM automaticamente descobriu as diferenças entre AML e ALL, como também entre as células ALL dos tipos B e T. Essas são as distinções mais importantes entre as leucemias agudas, ambas em termos da biologia quanto do tratamento clínico. Ou seja, a rede SOM conseguiu dividir os padrões em quatro aglomerados, encontrando uma outra categorização biológica importante.

Por fim, um outro exemplo de descoberta de classes utilizando técnicas de agrupamento pode ser encontrado em [Alizadeh et al., 2000]. Nesse trabalho, o linfoma difuso de grandes células B (DLBCL, do inglês *Diffuse Large B-cell Lymphoma*) foi estudado usando 96 amostras de linfócitos, 72 de células normais e 24 de células malignas, cada amostra contendo 4026 genes. Por meio da aplicação da técnica UPGMA a essas amostras, os autores mostraram que há uma diversidade na expressão gênica entre tumores de pacientes com DLBCL.

Foram identificadas duas formas moleculares distintas de DLBCL, que tinham padrões de expressão gênica indicativa de estágios diferentes da diferenciação da célula B. De fato, esses dois grupos estão correlacionados com a taxa de sobrevivência dos pacientes, portanto confirmando que os aglomerados gerados são biologicamente significativos.

5.3. Previsão de Classes

Até o momento foram discutidas apenas técnicas de AM não supervisionadas para a identificação de padrões de expressão gênica. Técnicas supervisionadas representam uma alternativa poderosa que pode ser aplicada se existe informação prévia sobre a classe dos genes. Por exemplo, técnicas de AM supervisionadas podem ser utilizadas para a predição de classes, conforme descrito nos problemas 5.3 e 5.4.

Problema 5.3 *[Molla et al., 2003] Predição de classes de doenças existentes.*
Dado: Os mesmos dados do Problema 5.2.
Faça: Aprender um modelo que possa classificar de maneira precisa uma nova célula na categoria da doença apropriada.

Problema 5.4 *[Molla et al., 2003] Predição da função biológica de um gene.*
Dado: Conjunto de genes representado como na Figura 17(c). Cada gene é um exemplo cujas características são os níveis numéricos de expressão medidos em várias circunstâncias experimentais. Essas condições experimentais incluem choque de temperatura, mudança no pH, ou a introdução de antibiótico; outra condição experimental inclui estágios diferentes de desenvolvimento de um organismo ou instantes de uma série temporal. A classe de cada gene pode ser simplesmente sua categorial funcional: por exemplo, ciclo TCA, Respiração e Histona.

Faça: Aprender a predizer a categoria funcional de genes adicionais (não vistos durante o treinamento) baseado em um vetor de níveis de expressão formado de acordo com o conjunto de condições experimentais especificadas.

Assim como para as técnicas de agrupamento, a escolha de um classificador supervisionado requer a seleção entre uma grande variedade de técnicas [Eisen et al., 1998, Brown et al., 2000, Furey et al., 2000, Khan et al., 2001, Xu et al., 2002, Shipp et al., 2002]. Nesse contexto, duas técnicas muito investigadas são as SVMs e as RNs [Brown et al., 2000, Furey et al., 2000, Khan et al., 2001, Xu et al., 2002, Shipp et al., 2002]. SVMs correspondem a uma família de técnicas estatísticas de AM particularmente apropriadas para as dimensões dos problemas de *microarray* [Brown et al., 2000, Furey et al., 2000].

Um exemplo da aplicação de SVMs para o Problema 5.4 pode ser encontrado em [Brown et al., 2000]. Os autores descrevem o uso de SVMs na classificação de genes utilizando expressões gênicas. Foram analisados dados de expressão de 2467 genes da levedura em 79 experimentos diferentes. As classes funcionais dos genes foram determinadas por meio de uma consulta ao MYGD (do inglês, *Munich Information Center for Protein Sequences Yeast Genoma Database*). Embora 2467 genes tivessem anotações, apenas seis classes funcionais foram escolhidas. Cinco delas foram consideradas pelos autores como representantes de perfis de mRNA reconhecíveis, e um grupo (classe) como dissociado de perfil reconhecível, o qual foi então utilizado como controle.

O experimento foi dividido em duas partes: primeiro, os autores treinaram SVMs, além de outras técnicas de AM, com os genes conhecidos. Foi utilizada a metodologia 5-*fold cross validation* para analisar os resultados. Em seguida, a SVM que mostrou o melhor resultado com os genes restantes não anotados foi utilizada para determinar se algum dos genes desconhecidos deveria estar em uma das classes conhecidas.

As técnicas empregadas incluem Janelas de Parzen (*Parzen windows*), discriminante linear de Fisher, ADs e SVMs com diferentes Kernels. Os experimentos com SVMs utilizaram dois tipos principais de funções Kernel: funções de base radial e funções polinomiais. De maneira geral, as várias técnicas tiveram desempenho similares, embora SVMs tenham apresentado a melhor acurácia. As técnicas tiveram um bom desempenho para classes com maior número de exemplos e tiveram dificuldades em identificar corretamente classes com poucos exemplos.

Com respeito ao Problema 5.3, [Khan et al., 2001] utilizaram RNs para analisar um conjunto de dados contendo medidas de expressão gênica, obtidos por meio de *microarray*, de quatro tipo de células cancerosas: neuroblastoma (NB), rhabdomyosarcoma (RMS), família de tumores Ewing (EWS), e linfomas Burkitt (BL), um tipo de linfoma non-Hodgkin. O objetivo do trabalho foi treinar uma RN capaz de classificar corretamente uma célula de um desse quatro tipos de tumores, utilizando apenas os níveis de expressão. Este foi o primeiro trabalho a realizar o diagnóstico de vários tipos de câncer simultaneamente.

Os *microarrays* mediram a expressão de 6567 genes. Esses dados foram em seguida filtrados para remover amostras abaixo de um valor mínimo de expressão, resultando em 2308 genes. Um total de 88 células (níveis de expressão dos genes) foram analisadas. Dessas, 63 foram usadas para treinamento e as restantes para teste. A técnica de análise de componentes principais foi utilizada para reduzir a dimensionalidade dos dados de 2308 genes para as dez componentes principais mais significativas. Para cada amostra, estes dez valores foram usados como entrada para a RN. A rede possuía quatro nodos de saída (um para cada classe), implementando funções lineares, e nenhuma camada escondida, ou seja, era uma simples rede Adaline [Haykin, 1999].

Ao invés de treinar uma única rede para a classificação, um total de 3758 redes foram treinadas, usando os dados do conjunto de treinamento. Um comitê formado pelas 3758 redes foi usado como classificador final. A saída do comitê para uma dada entrada era a média das saídas de todas as 3758 redes. Baseado nesse valor final e em uma regra para calcular a classe vencedora, uma dada entrada era classificada em das quatro classes ou rejeitada. Com esse método, todos os exemplos de treinamento e teste foram classificados corretamente.

6. Reconstrução Filogenética

A **reconstrução filogenética** consiste em estimar as relações de ancestralidade para um determinado número de táxons. Os **táxons** podem ser famílias, gêneros, espécies ou seqüências de macro-moléculas como DNA e aminoácidos. O resultado dessa análise é uma árvore, chamada **árvore filogenética ou filogenia**, que representa a história evolutiva dos táxons presentes nela. Uma Árvore Filogenética (AF) é composta por nós e ramos, em que os nós terminais correspondem aos táxons em estudo (no caso deste tutorial, seqüências de DNA ou aminoácidos), os nós internos representam seqüências ancestrais e os ramos representam relacionamentos topológicos entre os nós. Geralmente, o comprimento dos ramos representa o número de mudanças que ocorrem em relação ao último nó [Swofford et al., 1996].

A Figura 20 ilustra a representação de uma AF enraizada. Aves, crocodilos, cobras, lagartos, tartarugas, mamíferos e anfíbios são os táxons representados pelos nós terminais. Esses táxons são unidos aos seus ancestrais - representados pelos nós internos A, B, C, D, E e F - por meio de ramos (1-12) [Miyaki e Russo, 2001].

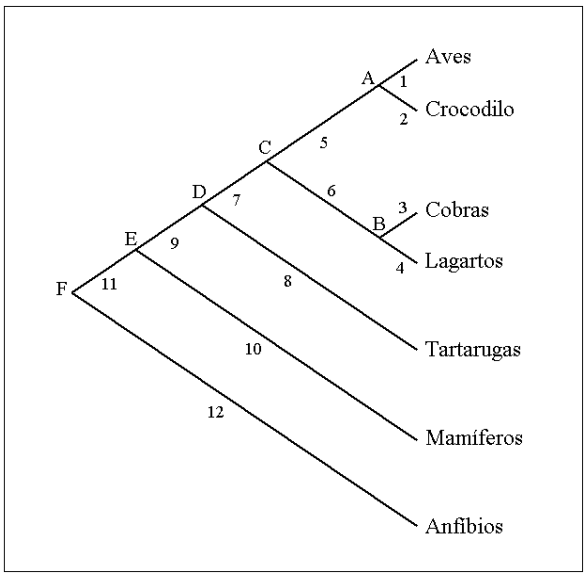

Figura 20: Representação de uma AF enraizada [Miyaki e Russo, 2001]

Por meio das AFs, pode-se, por exemplo, compreender os genomas atuais como o resultado dos processos evolutivos, verificar que o "funcionamento" das células atuais é o resultado de fenômenos que estão ocorrendo há bilhões de anos, comparar genomas, estudar a evolução de populações e processos migratórios ocorridos na espécie humana.

Um problema para os algoritmos de reconstrução de AFs é que o número de possíveis árvores cresce com o número de táxons. Para quatro táxons, existem três árvores possíveis. Para dez táxons, o número de árvores supera dois milhões. Diante da explosão combinatorial do espaço de busca, estratégias muito eficientes preci-

sam ser utilizadas para a obtenção da melhor árvore ou mesmo de uma árvore sub-ótima [Felsenstein, 1978, Swofford et al., 1996].

6.1. Métodos para Reconstrução de Árvores Filogenéticas

A reconstrução de AFs tem sido um dos principais desafios da Biologia Computacional [Camin e Sokal, 1965]. De fato, este é um problema NP-completo, o que significa que até o momento não existe nenhuma solução eficiente conhecida para o mesmo [Day, 1987]. Várias tentativas têm sido feitas para formalizar os critérios de escolha da árvore "verdadeira" (isto é, a árvore que represente de maneira mais fiel os relacionamentos evolutivos nos dados) de modo a tornar o problema mais tratável computacionalmente. Esse desafio tem levado ao surgimento de uma grande variedade de métodos para reconstrução de AFs, muitos deles usando algum tipo de heurística [Swofford et al., 1996].

Os vários métodos propostos para a construção de AFs podem ser classificados em dois grupos [Swofford et al., 1996, Miyaki e Russo, 2001]:

- Métodos não baseados em modelo, que usam um algoritmo para construir diretamente a árvore por meio de uma série de passos definidos. Nesse caso, o princípio do método está embutido no próprio algoritmo que resulta em uma árvore final.
- Métodos baseados em modelo, que primeiro definem um critério a ser maximizado (ou minimizado), para posteriormente usar um algoritmo para avaliar árvores potenciais, baseado no critério escolhido. Nesse caso o princípio do método é o critério utilizado para a escolha da melhor dentre um conjunto árvores.

Como o enfoque deste tutorial será no segundo grupo, por representar os métodos que tradicionalmente vêm sendo empregados em conjunto com AGs, o primeiro grupo de técnicas não será abordado.

Os métodos baseados em modelo seguem dois passos. Primeiro, um critério para a otimização é definido, que é simplesmente uma nota (*score*) usada para avaliar a qualidade de uma dada árvore. Segundo, um algoritmo é usado para calcular a nota para várias árvores, enquanto faz a busca pela melhor árvore (aquela que maximiza o critério) [Swofford et al., 1996]. Embora esses métodos pareçam atrativos por terem como compromisso encontrar a árvore ótima, de acordo com o critério estabelecido, eles podem ser computacionalmente muito lentos. Isso pode ocorrer até mesmo para um número moderado de táxons, tornando o tempo total necessário para uma busca exaustiva proibitivo. Essa limitação tem levando ao desenvolvimento de várias técnicas computacionais que procuram chegar, de maneira confiável, tão próximo à árvore ótima quanto possível.

Dois dos métodos mais utilizados para reconstrução de AFs baseados em modelo são os métodos de Máxima Parcimônia (MP) e Máxima Verossimilhança (MV). Em seguida, serão discutidas propostas recentes do uso de AGs para a implementação do mecanismo de busca empregado pelo método MV, com o objetivo de torná-lo mais eficiente.

Embora o método de MP ainda esteja entre os favoritos para construção de AFs, o método de MV está se tornando muito popular [Pereira et al., 2001]. A diferença crítica entre esses dois métodos é que o MP minimiza a quantidade de mudanças de estado de cada caracter[7] requeridas para a explicação dos dados, enquanto o método de MV tenta estimar a quantidade real de mudança de acordo com o modelo evolucionário adotado.

Além disso, para a aplicação do método de MV, é necessário que um modelo concreto de mudanças evolutivas que leve à conversão de uma seqüência em outra seja especificado. O modelo pede ser completamente definido, ou conter uma série de parâmetros

[7] Característica ou atributo que identifica, descreve, define ou diferencia um táxon.

a serem estimados a partir dos dados. Em outras palavras, no método de MV, os modelos de mudanças evolutivas são avaliados quanto à sua probabilidade de explicar um conjunto de dados de forma que reflita a história evolutiva mais próxima da realidade, ou seja, a história mais verossímil. Nessa avaliação, os modelos recebem valores de verossimilhança e aquele que apresentar o melhor valor é utilizado para inferir a AF.

6.2. Método da Máxima Verossimilhança: técnicas tradicionais e AGs

O maior problema com uso da MV é o alto custo computacional em termos de tempo de processamento, principalmente para encontrar a melhor árvore. Para um número pequeno de táxons (por exemplo, 11), a melhor e mais simples maneira para encontrar a árvore com maior verossimilhança é avaliar todas as árvores possíveis, ou seja, fazer uma busca exaustiva. Para problemas com um maior número de táxons, podem ser usados técnicas mais eficientes, tais como *branch-and-bound* e abordagens heurísticas (*stepwise addition* and *branch swapping*). Porém, elas ainda apresentam deficiências como a lentidão do *branch-and-bound* e os problemas de mínimos locais das abordagens heuristícas [Swofford et al., 1996].

Problema 6.1 *Reconstrução de árvore filogenética com método de MV*
Dado: Conjunto de seqüências de DNA ou aminoácidos e um modelo evolutivo.
Faça: Encontre a AF que possua maior probabilidade de explicar os dados segundo o modelo evolutivo.

Nesse contexto, tentativas recentes em diminuir o esforço computacional do método de MV por meio do uso de AGs têm surgido na literatura [Matsuda, 1996, Lewis, 1998, Reijmers et al., 1999, Skourikhine, 2000, Prado et al., 2002, Brauer et al., 2002, Katoh et al., 2001].

Por exemplo, em um trabalho pioneiro, [Matsuda, 1996] aplicou AGs na solução desse problema. Nesse caso, foram utilizadas seqüências de aminoácidos na obtenção da AF. Em termos de implementação, cada indivíduo da população era representado por um grafo e correspondia diretamente a uma AF. A função de aptidão utilizada na avaliação dos indivíduos era baseada na medida do log de suas verossimilhanças. Para a seleção dos indivíduos a formar novas gerações utilizou-se o critério da roleta. Sobre as AFs selecionadas, aplicava-se então operadores de elitismo, *crossover* e mutação, sendo esses dois últimos adaptados para a aplicação em questão. O operador de *crossover* foi modificado de forma a considerar os comprimentos dos ramos das AFs na obtenção do novo indivíduo. Foi utilizado um critério para maximização dos valores de verossimilhança das AFs antes destas serem submetidas ao *crossover*. No caso da mutação, ramos escolhidos aleatoriamente de algumas AFs eram trocados.

Na avaliação desse algoritmo algoritmo usando seqüências de aminoácidos de uma proteína denomina EF-1α, comparou-se o resultado obtido com o emprego de AGs aos alcaçados por vários algoritmos tradicionais para geração de AFs, como o UPGMA e o *neighbor-joining*. Embora não tenha chegado à melhor árvore possível (máximo global), o desempenho dos AGs foi considerado comparável aos das outras técnicas.

Uma abordagem semelhante à anterior foi proposta em [Lewis, 1998]. Porém, nesse caso as AFs foram obtidas a partir de seqüências de DNA. Outras diferenças importantes dizem respeito à forma como os comprimentos do ramos das árvores foram calculados e à definição e aplicação dos operadores de *crossover* e mutação. Nesse trabalho, os comprimentos dos ramos das AFs foram variados juntamente com suas topologias, em cada geração do AG. Todos os indivíduos selecionados para compor uma nova geração, exceto o melhor, eram submetidos ocasionalmente a mutações e *crossover*. As

mutações, aplicadas primeiramente, consistiam de mudanças nos comprimentos dos ramos das árvores e/ou de alterações topológicas, sendo estas últimas realizadas de forma similar à apresentada em [Matsuda, 1996]. O operador de *crossover* era então aplicado. Cada par de AFs para o *crossover* era dado por uma AF da população que sofreu mutações e uma AF da população "original".

Nos experimentos conduzidos, três árvores foram geradas a partir de um conjunto composto por 55 táxons. A melhor delas foi comparada a uma AF obtida por uma técnica heurística implementada no pacote PAUP* [Swofford e Begle, 1993]. Verificou-se que as AFs geradas possuíam a mesma topologia, variando apenas o tamanho dos ramos obtidos. O AG apresentou a vantagem, porém, de ser consideravelmente mais rápido (18 vezes) na obtenção desta solução. O algoritmo obtido, também denominado GAML (do inglês, *Genetic Algorithm for Maximum Likelihood Phylogeny Inference*), foi posteriomente paralelizado [Brauer et al., 2002]. Com isto, tornou-se possível aplicá-lo a seqüências de maior tamanho.

7. Conclusão

Abordagens de AM têm um papel fundamental na Biologia Molecular, devido a abundância de dados altamente variados e à ausência de teorias a um nível molecular. Este tutorial procurou propiciar uma visão ampla das principais abordagens de AM para a resolução de problemas da Biologia Molecular. Especificamente, os principais tópicos cobertos foram reconhecimento de genes, análise de dados de expressão gênica e reconstrução de filogenia.

7.1. Reconhecimento de Genes: Perspectivas

O reconhecimento e previsão da estrutura de genes é uma tarefa que envolve, em geral, diversos passos e técnicas. Nesse contexto, algoritmos de AM podem ser aplicados em uma ou mais etapas preditivas como, por exemplo, na identificação de junções de processamento e de regiões codificadoras.

Deve-se destacar que a forma de representação dos atributos do problema (no caso, das seqüências) pode ser determinante no sucesso do classificador gerado, sendo o foco de diversos estudos na área [Craven e Shavlik, 1993b, Farber et al., 1992, Uberbacher e Mural, 1991].

Como principais desafios futuros na área de reconhecimento de genes, tem-se a acomodação de eventos biológicos como o processamento alternativo, particularmente comum em seres humanos [Mironov et al., 1999]. A utilização de informações sobre a estrutura cromossômica pode também representar um meio para auxiliar o processo de identificação de genes [Pedersen et al., 1999]. Por exemplo, regiões de difícil acesso a proteínas podem ser consideradas mais propícias a não terem genes.

7.2. Análise de Dados de Expressão Gênica: Novas Direções

A análise de expressões de genes obtidas a partir de técnicas como *microarray* oferece uma oportunidade de gerar dados funcionais em uma escala genômica, podendo propiciar uma grande quantidade de dados necessários para interpretações biológicas dos genes e suas funções. Como visto nas seções anteriores, análises computacionais desses dados por meios de técnicas de AM têm se mostrado promissoras para, por exemplo, classificar doenças.

No entanto, resultados baseados em algoritmos de agrupamento ou mesmo em algoritmos de classificação supervisionados são dependentes das técnicas particulares usadas, da maneira pela qual os dados foram normalizados (intra ou inter experimentos), da

técnica utilizada para seleção de atributos e das medidas de proximidade usadas (no caso do algoritmos de agrupamentos). Qualquer um desse fatores pode ter um efeito considerável nos resultados da análise [Quackenbush, 2001].

Portanto, em geral, não se pode afirmar que há uma única classificação correta, embora técnicas de agrupamento diferentes possam ser mais ou menos apropriadas para diferentes conjuntos de dados. De fato, a aplicação de mais de uma técnica para a análise de um conjunto de dados particular pode trazer diferentes relacionamentos entre esses dados [Quackenbush, 2001, Slonim, 2002], inclusive com o uso das promissoras técnicas de agrupamento baseadas em modelo [McLachlan et al., 2002, Ji et al., 2003].

Em termos de desafios, a maioria das técnicas de agrupamento não inclui qualquer informação a *priori* em seus algoritmos. Uma questão central a ser investigada é se o agrupamento de padrões de expressão gênica pode ser feito biologicamente mais preciso incorporando nas técnicas de agrupamento de dados não apenas os níveis de expressão gênica, mas também a posição dos genes nos cromossomos, regiões de promotores, proteínas geradas, entre outros [Li et al., 2003, Slonim, 2002, Altman e Raychaudhuri, 2001].

7.3. Construção de Filogenia: Uso de Algoritmos Evolutivos

Como visto na Seção 6, um dos passos críticos na implementação do método da MV é a escolha de um algoritmo de busca que possa tratar de maneira eficiente a procura pela melhor árvore. Atualmente, há um conjunto de algoritmos que tradicionalmente são usados para implementar esse passo de busca (*branch-and-bound*, *stepwise addition* e *branch swapping*) [Swofford et al., 1996]. Recentemente, os AGs vêm sendo usados com uma alternativa a esses métodos tradicionais [Matsuda, 1996, Lewis, 1998, Reijmers et al., 1999, Skourikhine, 2000, Prado et al., 2002, Brauer et al., 2002, Katoh et al., 2001].

Os primeiros resultados obtidos com a aplicação de AGs na implementação do mecanismo de busca do método de MV são motivadores. Na maioria dos casos estudados, o desempenho dos AGs tem se mostrado comparável ou melhor do que aqueles obtidos por algoritmos tradicionais. No entanto, como no caso dos algoritmos tradicionais, quando o número de táxons cresce, o tempo de computação torna-se muito alto. Por exemplo, o processamento para algumas dezenas de táxons podem levar horas.

Com relação a essa questão, há uma perspectiva promissora. Alguns Algoritmos Evolutivos (incluindo AGs) com codificação de grafos especiais [Palmer e Kershenbaum, 1995, Knowles e Corne, 2000, Gen et al., 2001] foram propostas para problemas genéricos que envolvem construção de árvores geradoras. Tais abordagens têm obtido soluções adequadas para árvores com centenas de nós com tempos de computação da ordem de minutos ou mesmo segundos. Nesse sentido, adaptações dessas técnicas para os métodos de reconstrução filogenéticas podem produzir avanços significativos, permitindo a construção de árvores com um número de táxons relativamente elevado. A determinação de relações evolutivas para conjuntos de dados maiores poderia abrir portas para novas descobertas.

Agradecimentos

Este trabalho foi parcialmente financiado pela FAPESP e CNPq.

Referências

Alberts, B. et al. (1997). *Biologia Molecular da Célula*. Editora Artes Médicas, terceira edição.

Alizadeh, A. A. et al. (2000). Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511.

Altman, R. B. e Raychaudhuri, S. (2001). Whole-genome expression analysis: challenges beyond clustering. *Curr. Opin. Struct. Biol.*, 6(11):340–347.

Baldi, P. e Brunak, S. (2001). *Bioinformatics: the Machine Learning approach*. MIT Press, segunda edição.

Bäck, T., Hammel, U., e Schwefel, H.-P. (1997). Evolutionary computation: Comments on the history and current state. *IEEE Trans. on Evolutionary Computation*, 1(1):3–17.

Brauer, M. J. et al. (2002). Genetic algorithms and parallel processing in maximum-likehood phylogeny inference. *Mol. Biol. Evol.*, 19(10):1717–1726.

Brazma, A. e Vilo, J. (2000). Gene expression data analysis. *FEBS Letters*, 480(1):17–24.

Brenner, S. et al. (2000). Gene expression analysis by Massive Parallel Signature Sequencing (MPSS) on microbead array. *Nat. Biotechnol.*, 18(6):630–640.

Brown, M. P. et al. (2000). Knowledge-based analysis of microarray gene expression data by using support vector machines. In *Proc. of National Academy of Sciences USA*, volume 97, pp. 262–267.

Burset, M. e Guigó, R. (1996). Evaluation of gene structure prediction programs. *Genomics*, 34:353–367.

Camin, J. H. e Sokal, R. R. (1965). A method for deducing branching sequences in phylogeny. *Evolution*, 19:311–326.

Cancino, W., Liang, Z., e de Carvalho, A. C. P. F. (2003). Aplicação de algoritmos evolutivos multi-objetivo para o alinhamento de proteínas. In *Anais do IV Encontro Nacional de Inteligência Artificial (ENIA)*. A ser publicado.

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

Cho, R. J. et al. (1998). A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecu]lar Cell*, 2:65–73.

Costa, I. G., de Carvalho, F. A. T., e de Souto, M. C. P. (2003). Comparative study on proximity indices for cluster analysis of gene expression time series. *Journal of Inteligent and Fuzzy Systems*. A ser publicado.

Craven, M. W. e Shavlik, J. W. (1993a). Learning to predict reading frames in E. coli DNA sequences. In *Proc. of the 16th Hawaii International Conference on System Sciences*, pp. 773–782. IEEE Computer Society Press.

Craven, M. W. e Shavlik, J. W. (1993b). Learning to represent codons: a challenge problem for constructive induction. In *Proc. of the International Joint Conference on Artificial Intelligence*, pp. 1319–1324.

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

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

Day, W. H. E. (1987). Computational complexity of inferring phylogenies from dissimilarity matrices. *Bull. Math. Biol.*, 49:461–467.

Dopazo, J. et al. (2001). Methods and approaches in the analysis of gene expression data. *Journal Immunol. Methods*, 250(1/2):93–12.

Duggan, D. J. et al. (1999). Expression profiling using cDNA microarrays. *Nature Genetics*, 21:10–14.

Eisen, M. B. et al. (1998). Cluster analysis and display of genome-wide expression pattern. In *Proc. of National Academy of Sciences USA*, volume 95, pp. 14863–14868.

Faceli, K., de Carvalho, A. C. P. F., e da Silva-Jr, W. A. (2003). Selection for tumor cell classification. In *Anais do IV Encontro Nacional de Inteligência Artificial (ENIA)*. Aceito para publicação.

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

Felsenstein, J. (1978). The number of evolucionary trees. *Syst. Zool.*, 27:27–33.

Fickett, J. W. (1996). The gene identification problem: an overview for developers. *Computer and Chemistry*, 20(1):103–118.

Freeman, W. M., Walker, S. J., e Vrana, K. E. (1999). Quantitative RT-PCR: pitfalls and pontential. *Biotechniques*, 26:112–122.

Furey, T. S. et al. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):906–914.

Futschik, M. et al. (1999). Comparative studies of Neural Network models for mRNA analysis. In *Proc. of the International Conference on Intelligent Systems for Molecular Biology*. Heidelberg, Germany.

Gen, M., Cheng, R., e Oren, S. S. (2001). Network design techniques using adapted genetic algorithms. *Advances in Engineering Software*, 32:731–744.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley.

Golub, T. et al. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 5439(286):531–537.

Gribskov, M., Devereux, J., e Burges, R. (1984). The codon preference plot: graphic analysis of protein coding sequences and prediction of gene expression. *Nucleic Acids Research*, 12(1):539–549.

Guigó, R. et al. (1992). Prediction of gene structure. *Journal of Molecular Biology*, 253:51–60.

Harrington, C. A., Rosenow, C., e Retief, J. (2000). Monitoring gene expression using DNA microarrays. *Curr. Opin. Microbol.*, 3:285–291.

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*. Prentice Hall.

Hearst, M. A. et al. (1998). Trends and controversies - support vector machines. *IEEE Intelligent Systems*, 13(4):18–28.

Holland, J. H. (1992 (reprint)). *Adaptation in Natural and Artificial Systems : An Introductory Analysis With Applications to Biology, Control, and Artificial Intelligence (Complex A)*. Bradford Books.

Jain, A. K. e Dubes, R. C. (1988). *Algorithms for clustering data*. Prentice Hall.

Jain, A. K., Murty, M. N., e Flynn, P. (1999). Data clustering: a review. *ACM Computing Surveys*, 3(31):264–323.

Ji, X., Li-Ling, J., e Sun, Z. (2003). Mining gene expression data using a novel approach based on hidden markov models. *FEBS Letters*, 542(1/3):125–131.

Katoh, K., Kuma, K., e Miyata, T. (2001). Genetic algorithm-based maximum-likelihood analysis for molecular phylogeny. version 3.1. *J. Mol. Evol.*, 53:477–484.

Khan, J. et al. (2001). Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine*, 7:673–679.

Knowles, J. e Corne, D. (2000). A new evolutionary approach to the degree-constrained minimum spanning tree problem. *IEEE Trans. on Evolutionary Computation*, 4:125–134.

Kohonen, T. (1997). *Self-Organizing Maps*. Springer-Verlag.

Lapedes, A. et al. (1989). Application of neural networks and other machine learning algorithms to dna sequence analysis. In Bell, G. e Marr, T., editors, *Computers and DNA, SFI Studies in the Sciences of Complexity*, volume 7, pp. 157–182.

Lewis, P. (1998). A genetic algorithm for maximum-likehood phylogeny inference using nucleotide sequence data. *Mol. Biol. Evol.*, 15:277–283.

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

Li, T. et al. (2003). Classification by semi-supervised learning from heterogeneous data. In *Proc. of 18th ACM SAC - Bioinformatics Track*, pp. 78–82.

Lorena, A. C. et al. (2002a). The influence of noisy patterns in the performance of learning methods in the splice junction recognition problem. In *Proc. of the 7th Brazilian Symposium on Neural Networks (SBRN)*, pp. 31–36. IEEE Computer Society Press.

Lorena, A. C. et al. (2002b). Splice junction recognition using machine learning techniques. In *Proc. of the Brazilian Workshop on Bioinformatics (WOB)*, pp. 32–39.

Lorena, A. C. e de Carvalho, A. C. P. L. F. (2003). Human splice site identification with multiclass support vector machines and bagging. In *Lecture Notes on Artificial Intelligence, Joint 13th ICANN and 10th ICONIP*, Istanbul. Springer Verlag. A ser publicado.

Mangiameli, P., Chen, S. K., e West, D. (1996). A comparison of SOM neural network and hierarchical clustering methods. *European Journal of Operational Research*, 93:402–417.

Matsuda, H. (1996). Protein phylogenetic inference using maximum likelihood with a genetic algorithm. In *Proc. Pacific Symposium on Biocomputing*, pp. 512–523.

McLachlan, G. J., Bean, R. W., e Peel, D. (2002). A mixture model-based approach to the clustering of microarray expression data. *Bioinformatics*, 18(3):413–422.

Michaels, G. S. et al. (1998). Cluster analysis and data visualization of large-scale gene expression data. In *Pacific Symposium on Biocomputing*, volume 3, pp. 42–53.

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

Minsky, M. e Papert, S. (1969). *Perceptrons: An introduction to Computational Geometry*. MIT Press.

Mironov, A. A., Fickett, J. W., e Gelfand, M. S. (1999). Frequent alternative splicing of human genes. *Genome Research*, 9(12):1288–1293.

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

Miyaki, C. Y. e Russo, C. A. M. (2001). *Biologia Molecular e Evolução*, capítulo Reconstrução Filogenética. Introdução e o método da máxima parcimônia, pp. 97–107. Editora Holos.

Müller, K. R. et al. (2001). An introduction to kernel-based learning algorithms. *IEEE Trans. on Neural Networks*, 12(2):181–201.

Molla, M. et al. (2003). Using machine learning to design and interpret gene-expression microarrays. *AI Magazine (Special Issue on Bioinformatics)*. A ser publicado.

Monard, M. C. e Baranauskas, J. A. (2003a). *Sistemas Inteligentes: Fundamentos e Aplicações*, capítulo Conceitos sobre Aprendizado de Máquina, pp. 89–114. Editora Manole.

Monard, M. C. e Baranauskas, J. A. (2003b). *Sistemas Inteligentes: Fundamentos e Aplicações*, capítulo Indução de Regras e Árvores de Decisão, pp. 115–139. Editora Manole.

O'Neill, M. (1989). Escherichia coli promoters: II. A spacing class-dependent promoter search protocol. *Journal of Biological Chemistry*, 264:5531–5534.

Palmer, C. e Kershenbaum, A. (1995). An approach to a problem in network design using genetic algorithms. *Networks*, 26:101–107.

Pavlidis, P. et al. (2001). Promoter region-based classification of genes. In *Proc. of the Pacific Symposium on Biocomputing*, pp. 151–163.

Pedersen, A. et al. (1999). The biology of eukaryotic promoter prediction: a review. *Computers and Chemistry*, 23:191–207.

Pedersen, A. G. e Nielsen, H. (1997). Neural network prediction of translation initiation sites in eukaryotes: Perspectives for EST and genome analysis. In *Proc. of ISMB'97*, pp. 226–233.

Pereira, S. L., Miyaki, C. Y., e Russo, C. A. M. (2001). *Biologia Molecular e Evolução*, capítulo Reconstrução Filogenética: Métodos Probabilísticos, pp. 117–129. Editora Holos.

Pham, D. T. e Karaboga, D. (2000). *Intelligent Optimisation Techniques*. Springer-Verlag.

Prado, O., Zuben, F. J. V., e Reis, S. F. (2002). Evolving phylogenetic trees: An alternative. In *Proc. of Brazilian Workshop on Bioinformatics (WOB)*, pp. 56–63.

Quackenbush, J. (2001). Computational analysis of cDNA microarray data. *Nature Reviews*, 6(2):418–428.

Rampone, S. (1998). Recognition of splice-junctions on DNA sequences by BRAIN learning algorithm. *Bioinformatics*, 14(8):676–684.

Raychaudhuri, S., Stuart, J. M., e Altman, R. B. (2000). Principal components analysis to summarize microarray experiments: Application to sporulation time series. In *Proc. of Pacific Symposium on Biocomputing*, pp. 455–466.

Reed, R. (1993). Pruning algorithms - a survey. *IEEE Trans. on Neural Networks*, 4(5):740–747.

Reese, M. G. e Eeckman, F. H. (1995). Novel neural network algorithms for improved eukaryotic promoter site recognition. In *7th International Genome Sequencing and Analysis Conference*, South Carolina.

Reijmers, T. H. et al. (1999). Using genetic algorithm for the construction of phylogenetic trees: Application to g-protein coupled receptor sequences. *Biosystems*, 49:31–43.

Roberts, L. et al. (1995). Training neural networks to identify coding regions in genomic DNA. In *Proc. of the IEE Conference on Artificial Neural Networks*, pp. 399–403, London. Institution of Electrical Engineers.

Rosenblatt, F. (1958). The perceptrons: A probabalistic model for information and organization in the brain. *Psychological Review*, 56:386–408.

Rumelhart, D. E., Hinton, G. E., e Williams, R. J. (1986). Learning internal representations by error propagation. In Rumelhart, D. E., McClelland, J. L., e o PDP Research Group, editors, *Parallel Distributed Processing*, volume 1, pp. 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.

Setúbal, J. C. e Meidanis, J. (1997). *Introduction to Computational Molecular Biology*. PWS Publishing Company.

Sharan, R. e Shamir, R. (2002). CLICK: Clustering algorithm with applications to gene expression analysis. In *Proc. of Intelligent Systems for Molecular Biology*, pp. 307–316.

Shavlik, J. W. (1991). Finding genes by case-based reasoning in the presence of noisy case boundaries. In *Proc. of the DARPA Cased-Based Reasoning Workshop*, pp. 327–338.

Shipp, M. A. et al. (2002). Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature Medicine*, 8(1):68–74.

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, `http://www.icmc.usp.br/~inbio/material/Apostila_Inbio_Biomol.pdf`.

Skourikhine, A. (2000). Phylogenetic tree reconstruction using Self-Adaptive genetic algorithm. In *Proc. IEEE International Symposium on Bioinformatics and Biomedical Engineering*, pp. 129–134.

Slonim, D. (2002). From patterns to pathways: gene expression data analysis comes of age. *Nature Genetics*, 32:502–508.

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

Sneath, P. H. A. e Sokal, R. R. (1973). *Numerical Taxonomy*. W. H. Freeman.

Snyder, E. E. e Stormo, G. D. (1995a). Identification of protein coding regions in genomic DNA. *Journal of Molecular Biology*, 248:1–18.

Snyder, E. E. e Stormo, G. D. (1995b). Identifying genes in genomic DNA sequences. In *Nucleic Acid and Protein Sequence Analysis: a Pratical Approach*, Oxford. IRL Press.

Staden, R. (1984). Computer methods to locate signals in nucleic acid sequences. *Nucleic Acids Research*, 12(1):505–519.

Staden, R. (1990). Fiding protein coding regions in genomic sequences. *Methods in Enzymology*, 183:163–180.

Staden, R. e McLachlan, A. D. (1982). Codon preference and its use in identifying protein coding regions in long DNA sequences. *Nucleic Acids Research*, 10(1):141–156.

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

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

Swofford, D. L. e Begle, D. P. (1993). PAUP: phylogentic analysis using parsimony. Smithsonian Institution, Laboratory of Molecular Systematics, Washington, DC.

Swofford, D. L. et al. (1996). *Molecular Systematics*, capítulo Phylogenetic inference, pp. 407–514. Sinauer Associates, segunda edição.

Tamayo, P. et al. (1999). Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. In *Proc. Natl. Acad. Sci. USA*, 96:2907–2912.

Tavazoie, S. et al. (1999). Systematic determination of genetic network architecture. *Nature Genetics*, 22:281–285.

Toronen, P. et al. (1999). Analysis of gene expression data using self-organizing maps. *FEBS Letters*, 451:142–146.

Towell, G., Shavlik, J., e Noordewier, M. (1990). Refinment of approximate domain theories by knowledge-based neural networks. In *Proc. of the National Conference on Artificial Intelligence*, pp. 861–866. AAAI Press.

Uberbacher, E. C. et al. (1993). Gene recognition and assembly in the GRAIL system: Progress and challenges. In *Proc. of the International Conference on Bioinformatics, Supercomputing and Complex Genome Analysis*, pp. 465–476. World Scientific, Singapura.

Uberbacher, E. C. e Mural, R. J. (1991). Locating protein coding regions in human DNA sequences by a multiple sensor - neural network approach. In *Proc. of the National Academy of Sciences*, volume 88, pp. 11261–11265.

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

Velculescu, V. E. et al. (1995). Serial analysis of gene expression. *Science*, 270:484–487.

Vesanto, J. e Alhoniemi, E. (2000). Clustering of the self-organizing map. *IEEE Trans. on Neural Networks*, 11(3):586–600.

Wen, X. et al. (1998). Large-scale temporal gene expression mapping of central nervous system development. In *Proc. Natl. Acad. Sci. USA*, volume 95, pp. 334–339.

Xu, Y. et al. (1996). GRAIL: A multi-agent neural network system for gene identification. In *Proc. of the IEEE*, volume 84, pp. 1544–1552.

Xu, Y. et al. (2002). Artificial neural networks and gene filtering distinguish between global gene expression profiles of barrett's esophagus and esophageal cancer. *Cancer Res.*, 62:3493–3497.

Zhang, M. O. e Marr, T. G. (1993). A weight array method for splicing signal analysis. *Computer Application in Biosciences*, 9(5):499–109.

Zien, A. et al. (2000). Engineering suppot vector machine kernels that recognize translation initiation sites in DNA. *Bioinformatics*, 16:906–914.