Universidade Federal do Rio Grande do Norte

Centro de Tecnologia

Programa de Pós-Graduação em Engenharia Elétrica

Predição de Promotores de *Bacillus subtilis* usando técnicas de Aprendizado de Máquina

Meika Iwata Monteiro

Natal, dezembro de 2005

Universidade Federal do Rio Grande do Norte
Centro de Tecnologia
Programa de Pós-Graduação em Engenharia Elétrica

PREDIÇÃO DE PROMOTORES DE *BACILLUS SUBTILIS* USANDO TÉCNICAS DE APRENDIZADO DE MÁQUINA

Meika Iwata Monteiro

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica do Centro de Tecnologia da Universidade Federal do Rio Grande do Norte, como parte dos requisitos necessários para obtenção do grau de Mestre em Ciências.

Orientador: Prof. Dr. Luiz Marcos Garcia Gonçalves
Co-orientador: Prof. Dr. Marcílio Carlos Pereira de Souto

Natal, dezembro de 2005

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Aprovada em dezembro de 2005 pela comissão examinadora, formada pelos seguintes membros:

Prof. Dr. Luiz Marcos Garcia Gonçalves (Orientador)
Departamento de Engenharia de Computação e Automação - UFRN

Prof. Dr. Marcílio Carlos Pereira de Souto (Co-Orientador)
Departamento de Informática e Matemática Aplicada - UFRN

Prof. Dr. Adrião Duarte Dória Neto (Examinador Interno)
Departamento de Engenharia de Computação e Automação - UFRN

Jauvane Cavalcante de Oliveira (Examinador Externo)
Laboratório Nacional de Computação Científica- LNCC -RJ

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

Date: **dezembro de 2005**

Author: **Meika Iwata Monteiro**

Title: **Predição de Promotores de *Bacillus subtilis* usando Técnicas de Aprendizado de Máquina**

Department: **Programa de Pós-Graduação em Engenharia Elétrica**

Degree: **M.Sc.** Convocation: **January** Year: **2003**

Vencer não é nada, se não se teve muito trabalho; fracassar não é nada se se fez o melhor possível. (Nadia Boulanger)

Resumo

Um dos grandes desafios da Bioinformática é manipular e analisar os dados acumulados nas bases de dados mundiais. A expressão dos genes em procariotos é iniciada quando a enzima RNA polimerase une-se com uma região próxima ao gene, chamada de promotor, onde é localizado os principais elementos regulatórios do processo de transcrição. Apesar do crescente avanço das técnicas experimentais (*in vitro*) em biologia molecular, caracterizar e identificar um número significante de promotores ainda é uma tarefa difícil. Os métodos computacionais existentes enfrentam a falta de um número adequado de promotores conhecidos para identificar padrões conservados entre as espécies. Logo, um método para predizê-los em qualquer organismo procariótico ainda é um desafio. Neste trabalho, apresentamos uma comparação empírica de técnicas individuais de aprendizado de máquina, tais como: Classificador Bayesiano Ingênuo, Árvores de Decisão, Máquinas de Vetores de Suporte, Redes Neurais do tipo *VotedPerceptron*, PART e k-Vizinhos Mais Próximos e sistemas multiclassificadores (*Bagging e Adaboosting*) e Modelo Oculto de Markov à tarefa de predição de promotores procariotos em Bacilos *subtilis*. Utilizamos a validação cruzada para avaliar todos os métodos de AM. Para esses testes, foram construídas base de dados com seqüências de promotores e não-promotores do *Bacillus subtilis* e uma base de dados híbrida. Os métodos de AM obtiveram bons resultados com o SVM e o *Naïve Bayes*. Não conseguimos entretanto, obter resultados relevantes para a base de dados híbrida.

Palavras chaves: Aprendizado de Máquina, Reconhecimento de Promotores, Bioinformática.

Abstract

One of the most important goals of bioinformatics is the ability to identify genes in uncharacterized DNA sequences on world wide database. Gene expression on prokaryotes initiates when the RNA-polymerase enzyme interacts with DNA regions called promoters. In these regions are located the main regulatory elements of the transcription process. Despite the improvement of *in vitro* techniques for molecular biology analysis, characterizing and identifying a great number of promoters on a genome is a complex task. Nevertheless, the main drawback is the absence of a large set of promoters to identify conserved patterns among the species. Hence, a *in silico* method to predict them on any species is a challenge. Improved promoter prediction methods can be one step towards developing more reliable *ab initio* gene prediction methods. In this work, we present an empirical comparison of Machine Learning (ML) techniques such as Naïve Bayes, Decision Trees, Support Vector Machines and Neural Networks, Voted Perceptron, PART, k-NN and and ensemble approaches (Bagging and Boosting) to the task of predicting *Bacillus subtilis*. In order to do so, we first built two data set of promoter and nonpromoter sequences for *B. subtilis* and a hybrid one. In order to evaluate of ML methods a cross-validation procedure is applied. Good results were obtained with methods of ML like SVM and *Naïve Bayes* using *B. subtilis*. However, we have not reached good results on hybrid database.

Keywords: Learning Machine, promoters identification, Bioinformatics

Agradecimentos

Gostaria de deixar meus agradecimentos a essas pessoas que sem elas este trabalho não existiria:

À Prof. Lucymara,

Prof. Sílvia,

Prof. Kátia,

Maria Beatriz,

Fabíola,

Patrícia e Deyse.

Estas pessoas representam a equipe de biologia que deram o empurrão inicial do trabalho. Além de terem paciência para esclarecerem todas as dúvidas que foram surgindo ao longo do caminho.

Ao Prof. Marcílio,

Welbson,

Márjory,

Shirlly,

Camila e Valéria

Esses são o pessoal do Dimap que ajudaram a concretizar o trabalho ajudando e analisando de todas as formas de serem vistas.

Ao Prof. Luiz Marcos,

Prof. Pablo,

Prof. Adelardo,

Valnaide,

Douglas,

César,

Allan,

Anibal,

São as pessoas do DCA que me acolheram inicialmente para a definição deste trabalho.

Aos meus pais Abelardo e Akemi, à Marcelo e meus amigos Jessione, Jamerson, Isaac, Marconi, Juan Pablo, Fábio, Clauber, Anfranserai, George pelo apoio, carinho e paciência.

Agradeço também a todas outras que não foram citadas, mas que de alguma forma ajudaram para que este trabalho transformasse em realidade.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

1.1 Motivação

Nos últimos anos, os bancos de dados de genoma vêm crescendo exponencialmente devido à grande quantidade de dados que estão sendo produzidos (Baldi and Brunak 2001). Isso vem acontecendo, entre outras coisas, pela descoberta de novas técnicas de análise de seqüências genômicas. Manipular e analisar os dados acumulados nessas bases tornou-se um dos grandes desafios da Bioinformática. Por exemplo, o genoma humano possui aproximadamente três bilhões de pares de bases (nucleotídeos), no entanto apenas 2% deles são genes (HGMIS 2003): Como identificar que regiões são genes? Quais são as funções desses genes? Quais são as relações entre eles? Essas questões são relevantes para todos os organismos que vêm sendo sequenciados: homem, camundongo, bactérias, entre outros (Baldi and Brunak 2001).

Do ponto de vista da Bioinformática (análise *in silico*[1]), existem várias maneiras de se identificar um gene . Uma dessas maneiras é pela busca de sinais específicos nas seqüências de DNA (Pedersen et al. 1996; Baldi and Brunak 2001; Towell, Shavlik, and Noordewier 1990; Souto et al. 2003). Por exemplo, no processo de mapeamento de um fragmento de DNA (gene) para a formação de proteínas em uma célula,

[1] Análise computacional de um processo biológico.

o primeiro passo é a transcrição desse fragmento em uma molécula de Ácido Ribonucléico (RNA), chamado RNA mensageiro (Johnson, Raff, and Walter 2004). Esse processo se inicia com a ligação de uma molécula, a RNA polimerase, a regiões específicas do DNA chamadas de promotores (Johnson, Raff, and Walter 2004).

Ou seja, conhecendo-se a posição de um promotor sabe-se aproximadamente o início do transcrito (gene), portanto delimita-se um dos finais da seqüência do gene (Baldi and Brunak 2001; Craven and Shavlik 1994; Souto et al. 2003). O reconhecimento de promotores, por si só, também é um problema relevante. Por exemplo, esses métodos também podem ser vistos como uma das principais etapas na construção de redes regulatórias (Tavazoie et al. 1999): um gene "A" pode afetar a expressão de um gene "B" por meio da ligação da proteína produzida por "A" ao promotor do gene "B". Esse processo pode acontecer entre vários genes - representado através da redes regulatórias.

Embora haja uma grande quantidade de pesquisas na área de predição de promotores[2], o problema ainda não foi solucionado de maneira eficiente (Pedersen et al. 1996; Baldi and Brunak 2001). Os algoritmos existentes tendem a exibir ou uma especificidade baixa, gerando muitos falsos positivos, ou uma sensibilidade baixa. Isto ocorre, entre outras coisas, devido às interações complexas entre proteínas e a região promotora, que torna difícil a identificação de seqüências "características" (motivos ou regularidades) nos elementos promotores. Uma outra complicação para a tarefa de reconhecimento de promotores é que mesmo os "motivos" encontrados em seqüências promotoras também podem aparecer, de maneira aleatória, em outras partes do genoma.

Apesar dessas limitações, um grande número de programas de predição de promotores tem sido desenvolvido para organismos eucariotos (Pedersen et al. 1999;

[2]Na literatura de Bioinformática, é comum a utilização do termo "predição" com o sentido de "classificação" ou "reconhecimento" em Aprendizado de Máquina. Neste trabalho, usaremos ambos os termos no sentido de reconhecimento

Fickett and Hatzigeorgiou 1997; Rombauts et al. 2003; Werner 2003). Entretanto, até agora, são poucos os sistemas que podem ser usados como ferramenta para a predição de promotores em organismos procariotos - o programa *Neural Network Promoter Prediction* (NNPP) (Reese 2000) é uma das poucas exceções. Alguns outros métodos de predição de promotores de procariotos são baseados em busca de padrões em matrizes com pesos (Mulligan et al. 1984; Huerta and Collado-Vides 2003). Um método mais recente, baseado em propriedades sobre a estabilidade do DNA, foi proposto em (Kanhere and Bansal 2005).

Motivado pelo que foi discutido anteriormente, nesta dissertação apresentaremos um estudo sobre o uso de técnicas de Aprendizado de Máquina para o problema de reconhecimento de promotores de organismos procariotos. Serão analisados especificamente os promotores do *Bacillus subtilis* (*B. subtilis*). Escolhemos este organismo procarioto por ser amplamente utilizado como modelo em estudos genéticos, mas ao mesmo tempo com poucos trabalhos sobre a predição de seus promotores *in silico* (Oppon 2000; Kanhere and Bansal 2005). Além disso, há, na literatura, muitas análises experimentais comprovando várias de suas seqüências de promotores (Helmann 1995), o que ajudou na seleção e construção da nossa base de dados.

1.2 Objetivo e Abordagem

O objetivo principal desta dissertação é a aplicação e análise do uso de técnicas de Aprendizado de Máquina (AM) para o problema de reconhecimento de promotores do organismo *B. subtilis*. No contexto de AM, a identificação de seqüências promotoras pode ser colocada da seguinte forma (Souto et al. 2003):

- **Problema**: Identificação de seqüências promotoras.
- **Dados de entrada**: Conjunto de seqüências de DNA com um tamanho fixo contendo regiões promotoras conhecidas e seqüências que não possuem este

sinal.

- **O que fazer**: Gerar um classificador capaz de predizer se a janela de tamanho fixo tem ou não uma região promotora.

Inicialmente, para atingir esse objetivo, e como uma das contribuições dessa dissertação, construímos uma base de dados de seqüências promotoras específica para o *B. subtilis*. Uma das dificuldades na construção dessa base de dados é fato de que as seqüências disponíveis nos bancos de dados mundiais do *GenBank* (NCBI, EMBL, entre outros) (NCBI 2004; EMBL 2004) não possuírem anotações confiáveis. Por exemplo, a maioria dos promotores de organismos procariotos estão anotados como informações hipotéticas. Isto significa dizer que há pouca confiabilidade nesses dados, já que não há uma evidência experimental comprovada. Na construção da nossa base de dados, descartamos todas as seqüências promotoras que não tinham sido experimentalmente comprovadas.

Uma vez construída a base de dados, técnicas de AM foram aplicadas: Árvore de Decisão, PART, Classificador Bayesiano Ingênuo (*Naive Bayes Classifier*), *k*-Vizinhos Mais Próximos, Máquinas de Vetores Suporte, Redes Neurais do tipo *VotedPerceptron* e MLP (*Multi-Layer Perceptron*), e Modelos Ocultos de Markov (*Hidden Markov Models*). Tais métodos foram escolhidos por representarem diferentes paradigmas de aprendizado (baseado em regras, estatístico, baseado em instâncias, entre outros) e serem bastante citados na literatura, inclusive na abordagem específica deste problema (Hawley and McClure 1983; Nakata, Kanehisa, and Maizel 1988; O'Neill and Chiafari 1989; O'Neill 1991; O'Neil 1992; Delemer and Zhou 1991; Towell, Shavlik, and Noordewier 1990; Reese, Harris, and Eeckman 1996; Reese 2001; Pedersen et al. 1996; Lisser and Margalit 1993; Gordon et al. 2003; Oppon 2000; Kanhere and Bansal 2005). Com o objetivo de se conseguir uma melhora no desempenho dos classificadores individuais, sistemas de multiclassificação como o

Bagging e o *Boosting* também foram utilizados (Dietterich 2000; Kuncheva 2004; Lima 2005).

Por fim, para avaliar os resultados obtidos com os diferentes classificadores gerados, neste trabalhos empregamos o procedimento de validação cruzada (*cross-validation* - Mitchell (1997)). Também, a fim de se detectar diferenças estatisticamente significativas entre os resultados, utilizamos o teste de hipótese (Mitchell 1997; Dietterich 1998).

1.3 Estrutura da Dissertação

O restante deste trabalho está dividido em cinco capítulos, organizados da seguinte forma:

- **Capítulo 2**. Explicamos conceitos básicos sobre a célula, mantendo o foco na definição de promotor e sua estrutura. Também são apresentamos os principais trabalhos relacionados com o nosso.

- **Capítulo 3**. Descrevemos, resumidamente, as técnicas de aprendizado de máquina usadas nos nossos experimentos.

- **Capítulo 4**. Descrevemos a metodologia de avaliação, como também a maneira como conduzimos os experimentos. Nesse capítulo, mostramos todas as etapas que usamos na construção da base de dados - uma das principais contribuições desta dissertação.

- **Capítulo 5**. Apresentamos e analisamos os resultados obtidos com os experimentos.

- **Capítulo 6**. Apresentamos algumas conclusões e perspectivas futuras derivadas a partir deste trabalho.

Capítulo 2

Promotores

Como mencionado anteriormente, o foco desta dissertação é sobre o reconhecimento de promotores de organismos procariotos, mais especificamente da espécies *B. subtilis*. Nesse contexto, no presente capítulo são descritos os principais conceitos biológicos necessários ao entendimento desta dissertação. Além disso, a fim de colocar os nossos resultados em perspectiva, é feita uma revisão bibliográfica sobre os trabalhos relacionados. A maioria desses conceitos biológicos foram retirados de (Johnson, Raff, and Walter 2004).

2.1 Conceitos Básicos

Para entender sobre promotores, deve-se compreender alguns conceitos básicos das estruturas químicas contidas nas células. As células são unidades constituintes de toda matéria viva existente (Johnson, Raff, and Walter 2004). Elas armazenam informações hereditárias compostas de diversos fatores químicos. Essas informações estão presentes na forma de moléculas de Ácido Desoxirribonucleico (do inglês *DesoxirriboNucleic Acid* - DNA) observados na Figura 2.1. Ele é formado pelos mesmos quatro tipos de nucleotídeos - duas purinas, adenina (A) e guanina (G) e duas pirimidinas, timina (T) e citosina (C).

Cada nucleotídeo é composto de duas partes: um açúcar (desoxirribose), com

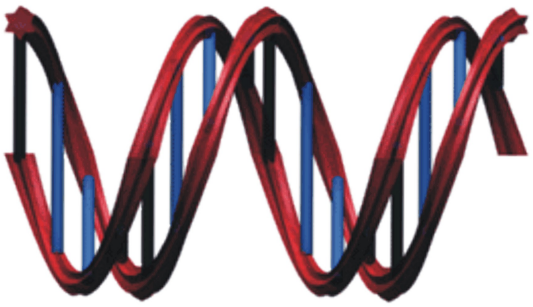

Figura 2.1: Estrutura da molécula do DNA

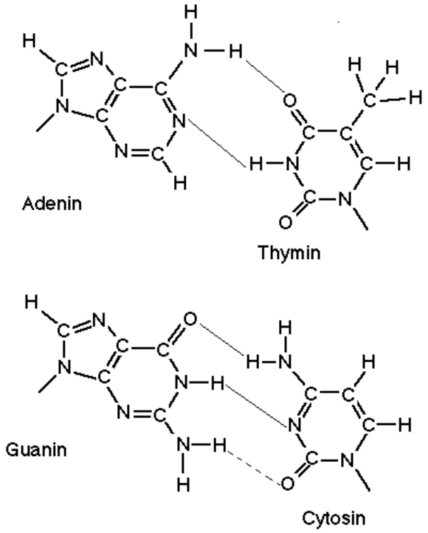

Figura 2.2: Representação das ligações químicas entre os nucleotídeos

um fosfato ligado a ele, e uma base. As bases nitrogenadas presentes na fita de DNA ligam-se com outras bases, de acordo com os padrões de suas estruturas: Adenina liga-se com Timina; Citosina liga-se com Guanina e vice-versa formando uma dupla-hélice Figura 2.2.

Para manter sua função de armazenamento de informação antes de divisão celular, o DNA faz cópias de si mesmo. Esse processo de replicação do DNA é catalisada por enzimas denominadas RNA polimerase. Tais enzimas sintetizam a nova fita de DNA, utilizando como molde (*template*) a fita complementar da molécula existente.

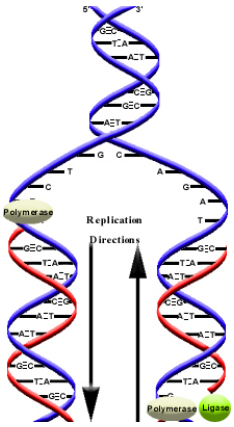

Figura 2.3: Processo de Replicação do DNA

Para que ocorra a síntese, a molécula de DNA deve ser desnaturada, ou seja, as fitas devem ser separadas pelo rompimento das ligações que mantêm a dupla hélice.

A informação do DNA também deve ser expressa para guiar a síntese de outras moléculas da célula. Isto ocorre na produção de proteínas e Ácido Ribonucléico (do inglês *RiboNucleic Acid* - RNA) visualizado na Figura 2.3 . O RNA é uma molécula de ácido nucléico similar do DNA, exceto pela substituição da desoxirribose por ribose e de Timina (T) por Uracil (U). As moléculas de RNA são sintetizadas por um processo conhecido como transcrição, similar à replicação (Figura[1] 2.4) .

Nesse processo, uma das duas fitas do DNA atua como modelo para que o pareamento de bases complementares possa acontecer. Após a transcrição do DNA, a fita de RNA mensageiro (RNAm) é liberada da molécula de DNA, voltando o DNA à conformação original. Depois, em um processo mais complexo denominado tradução, muitas dessas moléculas de RNA direcionam a síntese de polímeros pertencentes a classes químicas radicalmente diferentes - as proteínas (Figura[2] 2.5) . Todo este complexo processo é montado por uma gigantesca máquina multimolecular, o ribossomo, formado por duas cadeias principais de RNA, chamada RNA ribossomal (RNAr).

[1]Figura retirada de (Conceitos básicos de biologia molecular)
[2]Figura retirada de (Conceitos básicos de biologia molecular)

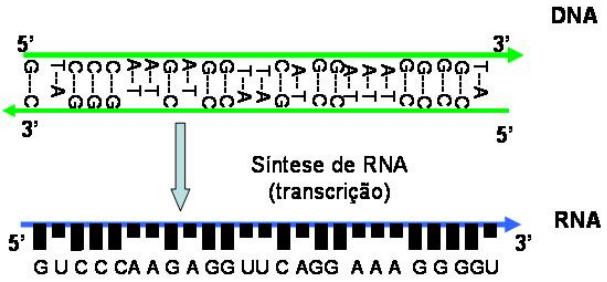

Figura 2.4: Processo de transcrição do DNA

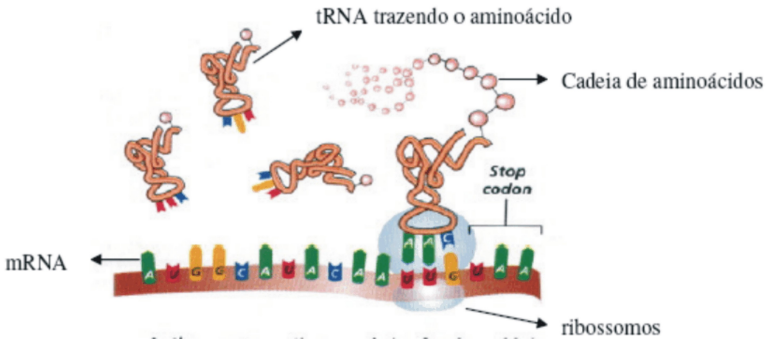

Figura 2.5: Processo de tradução do DNA

As moléculas de proteínas carregam informações em forma de seqüências lineares de símbolos (os aminoácidos). As proteínas dirigem a maioria dos processos químicos nas células. Elas são encarregadas de funções estruturais, geradoras de movimentos, sensoriais e, acima de tudo, elas são responsáveis por colocar em ação a informação genética da célula.

Normalmente, as moléculas de DNA são muito grandes e contêm as especificações para milhares de proteínas. Os segmentos de seqüências inteiras de DNA são portanto, transcritos em moléculas de RNAm separadas, codificando uma proteína diferente.

Nesse contexto, um gene é definido como um segmento de DNA correspondente a uma única proteína (ou a uma única molécula de RNA catalítico ou estrutural, para aqueles genes que produzem RNA, mas não proteínas). Os genes são ativados de acordo com a necessidade de cada célula. Todas as informações genéticas contidas numa seqüência completa de DNA formam o genoma de uma célula.

2.2 Organismo Procarioto

As células compõem os organismos, que podem ser procarióticos, estruturas mais simplificadas e sempre unicelulares (como as bactérias) ou eucarióticos, que incluem plantas multicelulares, animais e fungos, assim como organismos unicelulares, como leveduras e algas verdes. A principal diferença entre os ambos organismos está na ausência do envoltório nuclear nas células dos organismos procarióticos (Zaha 1996) visto na Figura[3] 2.6.

A maioria das células procarióticas são simples, pequenas e normalmente habitam locais independentes, sem necessitar viver em colônias, como os organismos multicelulares. Os procariotos possuem uma conformação esférica ou em forma de bastões.

[3]retirada de (Células procarióticas e eucarióticas)

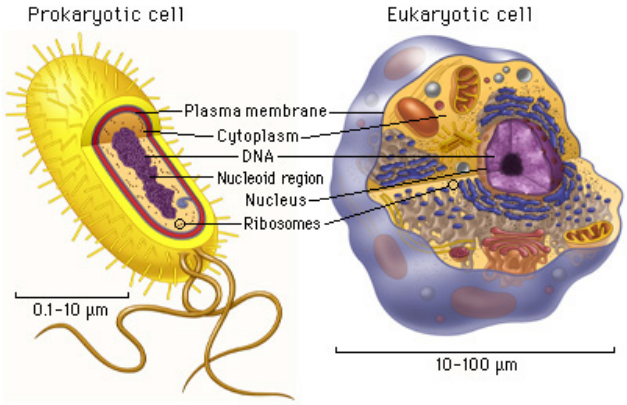

Figura 2.6: Células de organismos procariótico e eucariótico

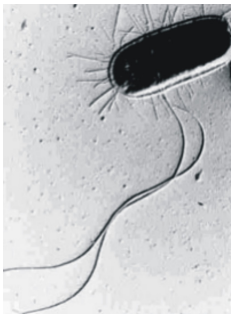

Figura 2.7: *Escherichia Coli*

Podem ser Gram-positivos ou Gram-negativos[4]. Freqüentemente apresentam uma capa protetora denominada parede celular, seguida de uma membrana plasmática envolvendo num único local o material citoplasmático que contém: o DNA, o RNA, as proteínas e outras moléculas. Elas vivem em uma enorme variedade de nichos e possuem uma capacidade bioquímica extraordinariamente variada muito além das células eucarióticas.

No contexto do presente trabalho, o foco será em duas espécies de bactérias: *Escherichia coli* e *Bacillus subtilis* (Tabela 2.1).

A biologia molecular tem focalizado sua atenção, há muito tempo, na espécie *E. coli* (Figura[5] 2.7). Esta espécie vive no intestino de humanos e de outros vertebrados,

[4]Classificação dos microorganismos, com base em sua resposta ao Teste de Gram, em decorrência das características de sua parede celular (Johnson, Raff, and Walter 2004)

[5]Figura retirada de (Nature 2001)

Bactéria	Formato	Tipo	Respiração
E. coli	Bastão	Gram negativo	Anaeróbico facultativo
B. subtilis	Bastonete	Gram positivo	Aeróbico

Tabela 2.1: Características das bactérias *E. coli* e *B. subtilis*

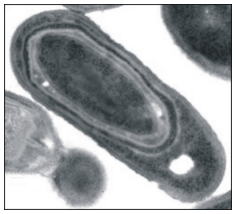

Figura 2.8: *Bacillus Subtilis*

e é amplamente encontrada em resíduos e dejetos recentes, sendo uma das razões que a torna uma indicadora de contaminação fecal (Maza et al. 1994).

Além de viver nesses ambientes, ela pode crescer facilmente em um simples meio com nutrientes, como em uma placa de cultura. A evolução tem otimizado a *E. coli* a sobreviver em condições químicas variáveis e reproduzir-se rapidamente. Atualmente temos mais conhecimento, em termos moleculares, da *E. coli* do que de qualquer outro organismo. Na *E. coli* são analisados a síntese de proteínas ou os mecanismos genéticos, que foram conservados ao longo da evolução e são, essencialmente, os mesmos mecanismos que existem em nossas próprias células, assim como na *E. coli*.

Por sua vez, a bactéria *B. subtilis* (Figura[6] 2.8) representa uma das bactérias esporuladas comumente encontradas no ar. Ela apresenta endosporo[7] localizado centralmente e cresce em temperaturas moderadas, além de produzir antibióticos (Maza et al. 1994). Sua seqüência genômica foi a primeira a ser completamente sequenciada (Kunst 1997).

[6]Figura retirada de (Bacillus Subtilis)

[7]O endosporo é uma célula, formada no interior da célula vegetativa, altamente resistente ao calor, dessecação e outros agentes físicos e químicos, capaz de permanecer em estado latente por longos períodos e degerminar dando início a nova célula vegetativa (Johnson, Raff, and Walter 2004).

Essa seqüência oferece informações relevantes da capacidade da bactéria na utilização de uma variedade de recursos de carbono e secreção de grandes quantidades de enzimas importantes nas indústrias. Ela também contém, no mínimo, 10 fragmentos de prófagos ou restos de prófagos[8], mostrando um papel significante para infecção de bacteriófago na transferência de genes durante a evolução bacterial (Kunst 1997).

2.3 Promotores em Procariotos

Promotor é um fragmento de seqüência de DNA localizado normalmente na posição 5' , anterior ao gene (*upstream*), contendo aproximadamente 200 pares de bases. Essa posição pode ser vista na Figura 2.9 - retirada de (Neves and Lemke 2005). (Posição 5' é a "polaridade" de polímeros para uma seqüência repetitiva de açúcar-fosfato-açúcar-fosfato no sentido de que possuem uma direcionalidade definida: o fosfato da posição 5' de um nucleotídeo liga-se à posição 3' do próximo nucleotídeo.) Apesar dessa definição, nem todas as seqüências *upstream* dos genes funcionam como promotores. Promotores são seções específicas de seqüências de DNA reconhecidos por proteínas específicas, diferenciando de outras partes do DNA que são transcritas e traduzidas.

A posição do promotor é identificada no início da transcrição, a partir da qual o DNA serve como molde para a síntese de RNA mensageiro. Esse sítio é conhecido como sítio de início da transcrição ou TSS (do inglês *Transcription Start Site*), numerado como +1. A posição dos nucleotídeos anteriores a ele é localizada por inteiros negativos (região *upstream*), e os posteriores, por inteiros positivos (região *downstream*) como pode ser observado na Figura 2.10.

A transcrição nos procariotos é iniciada quando uma enzima de RNA polimerase

[8] genomas de bacteriófagos integrados no DNA da célula de uma bactéria. Os prófagos podem ser duplicados para várias gerações de células até alguns estímulos induzirem sua ativação e virulência.

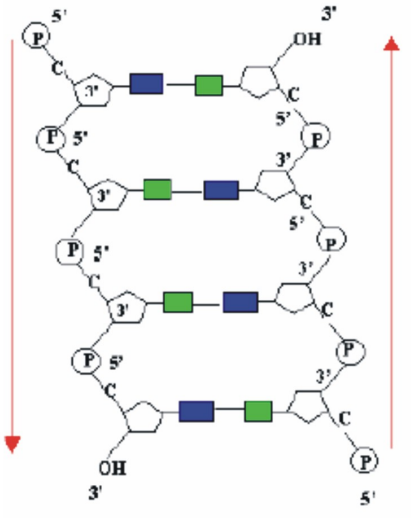

Figura 2.9: Esquema de arquitetura do DNA mostrando as extremidades 5' e 3'

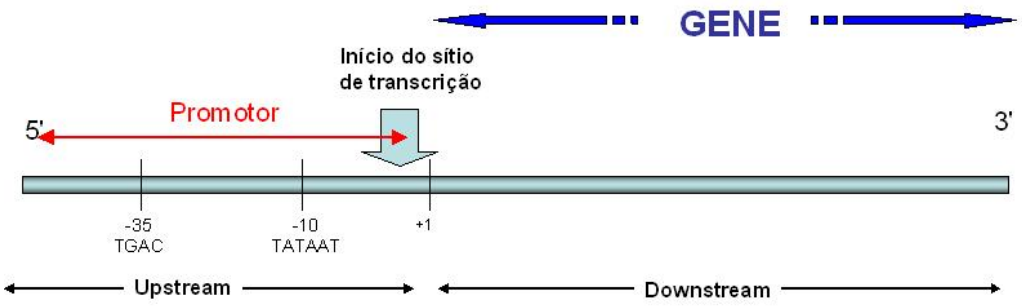

Figura 2.10: Região promotora numa seqüência de DNA

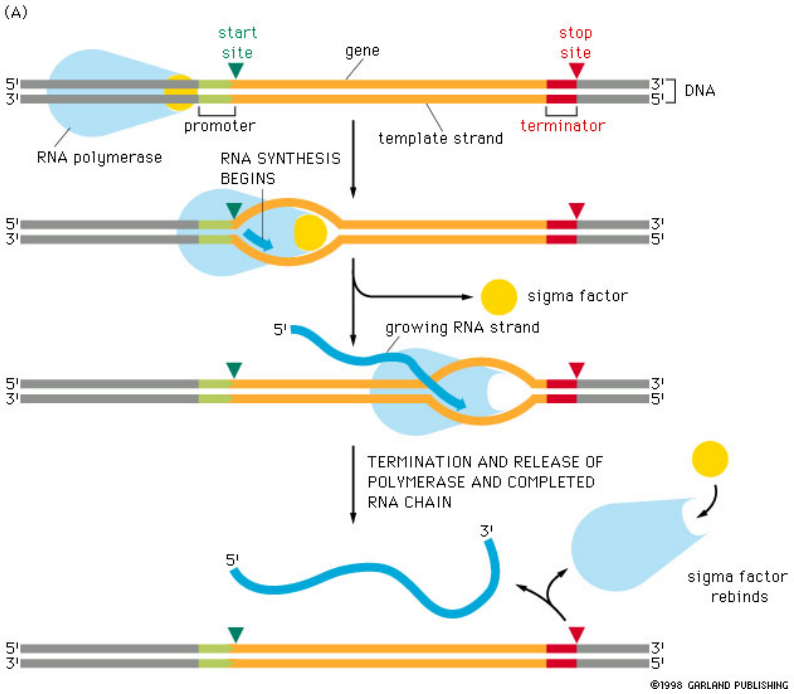

Figura 2.11: Transcrição do RNAm Início: promotor na extremidade 5'. O RNAm é sintetisado na direção 5' para 3'.A seqüência é a mesma que a fita *sense* do DNA. Término: sinais específicos na estrutura do próprio RNA ricas em uracila.

forma um complexo de fatores sigma σ (subunidade da RNA polimerase procariótica necessária à ligação ao promotor, dando início à transcrição). O "complexo fechado" (dupla hélice) converte-se em "complexo aberto" (DNA desenrolado) desfazendo uma pequena região do DNA. O passo final acontece na junção da região dos primeiros nucleotídeos numa ligação fosfodiéster (RNA mensageiro) seguida da liberação dos fatores sigma/transcricionais, os quais acredita-se serem um dos responsáveis pelo direcionamento da RNA polimerase para o promotor. Inicia-se, então, o alongamento do transcrito até encontrar a seqüência de parada que indica o término da transcrição (Helmann 1995). Esse processo pode ser observado na Figura[9] 2.11.

Algumas características foram relatadas na literatura como indicadores da presença de promotores. Os promotores de procariotos possuem um tipo de RNA

[9]retirada de (Lectures BMS 209/BMS)

polimerase mais simples do que as dos promotores de eucariotos, que possuem três tipos de RNA polimerase. Os elementos transcripcionais foram experimentalmente identificados em seqüências promotoras de procariotos nas posições *upstream* -10 e -35 do gene. Por exemplo, foram encontrados padrões em *Bacillus subtilis* por Pribnow (1975) com evidências experimentalmente confirmadas, mostrando que em muitas amostras a posição -10 terá nucleotídeos TATAAT (denominado hexâmero -10) (TATA-box[10]), e a posição -35 terá TTGACA (denominado hexâmero -35) (Pedersen et al. 1996; Pedersen and Engelbrecht 1995). Além disso, no caso específico de eucariotos, a grande presença de Guanina e Citosina, denominadas ilhas de CpG (sendo p representando a ligação fosfodiéster do DNA) (Antequera and Bird 1993; Craig and Bickmore 1994) também demonstra uma característica dos promotores. No entando, por ser o direcionamento desse trabalho voltado aos procariotos, os eucariotos não serão detalhados. Mais informações sobre eles podem ser encontradas em (Hannenhalli and Levy 2001; Pedersen et al. 1999).

Promotores em organismos procariotos são menos complexos que os eucariotos, por possuírem tamanho e números reduzidos, além de elementos reconhecidos pelos fatores σ (Chamberlin 1974; Hawley and McClure 1982). Esses fatores têm sido reconhecidos por consórcios de enzimas que transcrevem diferentes classes de promotores. Eles não somente localizam o início da transcrição mas também conseguem saber em que proporção isso ocorre.

Foi estabelecido que a intensidade do promotor, definida como o grau de transcritos por genes que são produzidos, é primariamente determinada por dois fatores: a afinidade de ligar-se com a RNA polimerase e a proporção entre complexos de promotores fechados, DNA dupla fita e complexos de promotores abertos, DNA aberto (Chamberlin 1974; Hawley and McClure 1982).

[10]Região rica em nucleotídeos T e A

Portanto, pode-se usar a seqüência promotora para regular a velocidade de transcrição do DNA numa proteína. Isso implica a sua utilização para modificação genética de alimentos (Reichhardt 2000), ou ainda em ter uma proteína que atacaria uma determinada doença, fazendo-a regredir, podendo até destruí-la (Rorth, Szabo, and et. al 1998).

2.3.1 Promotores em *Escherichia coli*

Vários pesquisadores descobriram mais de 300 promotores de *Escherichia coli* (*E. coli*) e foi notado que aproximadamente 60 pares de bases de cada uma dessas seqüências de promotores estão associados a uma interação com a RNA polimerase (Oppon 2000).

Há quatro características existentes na maioria dos promotores de *E. coli* (Hawley and McClure 1983):

- Sítio de início de transcrição: em mais de 90% dos promotores, foram encontrados uma concentração significativa de purina no TSS (Hawley and McClure 1983).

- Hexâmero -10 (TATAAT): o centro do hexâmero está em aproximadamente 10pb do TSS (Hawley and McClure 1983), por isso que, por essa localização, ela é chamada de hexâmero -10. Sua região consenso é TATAAT, que pode ser exemplificada na forma $T_{80}A_{95}T_{45}A_{60}A_{50}T_{96}$, em que os números subescritos indicam a probabilidade de ocorrência de cada base em sua localização (Hawley and McClure 1983).

- Hexâmero -35 (TTGACA): um outro hexâmero conservado está na região *upstream* do TSS em torno da região -35 do gene. Sua região consenso abrange os nucleotídeos TTGACA e possuem um percentual de conservação de aproximadamente $T_{82}T_{84}G_{78}A_{65}C_{54}A_{45}$ (Hawley and McClure 1983).

- Distância das seqüências entre -10 e -35: em 90% dos promotores são encontrados de 16pb a 18pb entre os sítios -35 e -10 (regiões intergênicas) e em poucos casos encontram-se com 15 ou 21pb. Esta distância está associada a conformação geométrica da RNA polimerase (Olekhnovich and Kadner 1999).

Em resumo, um promotor de *E. coli* ideal possui 17pb entre os hexâmero -35 e -10, posicionado em torno de 7pb do sítio de início de transcrição. O sinal de reconhecimento da RNA polimerase está na região -35, enquanto que a região -10 permite que o complexo se converta da forma "fechada" para forma "aberta" (Hawley and McClure 1983).

Promotores de sete genes *rrh* de *E. coli* possuem outra característica, além das citadas anteriormente. Esses genes codificam RNA ribossômico e são vitais para o rápido crescimento das células (Ross et al. 1993). Normalmente, o RNA ribossômico não possui tanta força de ativação quando comparado a qualquer outro. Mas nesse caso, sua força está atribuída ao acúmulo de regiões ricas em AT de aproximadamente 20pb localizadas logo depois da região *upstream* -35. Essas regiões ricas em motivos de AT são denominadas elemento *upstream* ou elemento UP. Este elemento foi encontrado num sistema de transcrição *in vitro* contida somente na RNA polimerase e em seqüências promotoras de DNA.

O funcionamento do elemento UP está diretamente relacionado a manutenção da fase de hélice no sítio de início de transcrição. Ele permite que ocorra alterações na conformação do DNA entre as regiões intergênicas, mesmo havendo mutações nos espaços entre as ligações dos nucleotídeos. Isso faz com que as distâncias entre as regiões -10 e -35 permaneçam as mesmas.

A região transcrita inicial (localizada aproximadamente na região +1 a +30 *downstream* do gene) influencia na taxa em que a RNA polimerase atua no promotor ativando ou anulando-o. Por isso a atuação do promotor de *E. coli* não pode ser

predita somente analisando as bases entre os consensos -35 e -10 (Oppon 2000).

2.3.2 Promotores em *Bacillus subtilis*

Segundo Yamada et al. (1991), promotores de *Bacillus subtilis* e *E. coli* transcritos pelos fatores Eσ^A ou Eσ^{70} possuem muitas similaridades, tais como: as seqüências conservadas nos hexâmeros -35 e -10, a distância entre os dois hexâmeros e a posição do sítio de início de transcrição. Portanto, a maioria dos promotores de *B. subtilis*, em geral, funcionam bem em *E. coli*. Em contraste, alguns promotores de *E. coli* (por exemplo, o promotor do gene lacUV5) não conseguem ser transcritos pela RNA polimerase do *B. subtilis* (Yamada et al. 1991).

Uma outra sequencia conservada no *B. subtilis* é a seqüência $5' - RTRTG - 3'$ (R = purina). Ela foi primeiro encontrada em 9 promotores na região -16. Uma análise mais exaustiva de 142 promotores, todos com o sítio de início de transcrição (TSS) experimentalmente confirmados, confirmaram a conservação da região -16 do *B. subtilis* (Helmann 1995). O dinucleotídeo TG da região *upstream* -10 foi encontrada em 45% dos promotores de *B. subtilis*, sendo 52% de T e 58% de G (Helmann 1995). Os resíduos T(imina) e R (purina) foram também relacionados com os dinucleotídeos TG em alguns promotores (Helmann 1995; Oppon 2000).

Quando comparados a longas regiões do genoma, as regiões *upstream* do promotor possuem seqüências de A^n e T^n em que $n > 4$, (Helmann 1995). Esses dados indicam que a ligação da RNA polimerase afeta a seqüência do DNA até a região *upstream* -70 (Helmann 1995).

Os padrões de conservação de nucleotídeos observados em promotores de *E. coli* podem ser resumido por TTGaca (N_{17+-1}) TAtAAT (as bases em letras maiúsculas representam esta ocorrência em mais de 70% dos promotores.) (O'Neill 1989). Estudos bioquímicos mostram que o *B. subtilis* seria menos tolerante ao desvio em relação ao consenso de 12pb, ao invés do *E. coli*[11] (O'Neill 1989).

2.4 Análise de promotores *in silico*

O estudo de promotores *in silico* foi iniciado em organismos procariotos devido a apresentarem uma estrutura menos complexa do que os promotores eucariotos. Os pioneiros na análise computacional sistemática que utilizaram os promotores de *E. coli* foram Hawley and McClure (1983). Eles estudaram 168 regiões promotoras e estabeleceram seqüências consensos de promotores procariotos. Mais tarde, Harley and Reynolds (1987) continuaram o trabalho identificando mais 263 promotores de *E. coli*.

Nakata, Kanehisa, and Maizel (1988) utilizaram a compilação de promotores de *E. coli* de Harley and Reynolds (1987) para construir uma base de dados. Essa base de dados foi submetida a uma rede perceptron (Minsky and Papert 1969). O trabalho de Nakata, Kanehisa, and Maizel (1988) podem ser vistos como o início das aplicações de aprendizado de máquina para o problema de reconhecimento de promotores.

O'Neill and Chiafari (1989) usaram o método de região consenso em 47 promotores conhecidos de *E. coli*, dividindo-os em três classes de acordo com as distâncias entre os hexâmeros -35 e -10 (16, 17 ou 18 pares bases). 77% dos promotores foram corretamente identificados, mas o nível de falsos-positivos foi muito alto. Mais tarde, este trabalho foi continuado por O'Neill (1991) e O'Neil (1992) com as seguintes resultados de predição: 78 a 100% para 16pb, 97% para 17pb e 79% para 18pb.

Delemer and Zhou (1991) extenderam o trabalho de Nakata, Kanehisa, and Maizel (1988) por meio do uso de uma MLP (*MultiLayer Perceptron*), em vez do *perceptron*, treinada com *backpropagation*. Essa rede foi treinada com 80 promotores e com uma certa quantidade de seqüências aleatórias (exemplo de não-promotores). O classificador obtido foi testado com 30 seqüências de promotores e

[11] Promotores de *B. subtilis* coincidem em média com o consenso em 9,1 posições, comparados a 7,9 dos promotores de *E. coli*.

1500 de não-promotores, apresentando uma taxa de classificação correta de promotores (verdadeiro-positivo) de 100%.

Porém, esses valores podem ter sido afetados pela escolha dos dados porque, primeiramente, a base de dados era muito pequena, além disso, a forma que os exemplos negativos (não-promotores) foram gerados, influenciou no protocolo de busca do promotor que possuía alta sensibilidade pela média de $A(denina)/T(imina)$ quando já existe, normalmente, relativa presença de $A(denina)/T(imina)$ no promotor (Mulligan et al. 1984; O'Neill and Chiafari 1989).

Towell, Shavlik, and Noordewier (1990) aplicaram uma abordagem híbrida de RNs (Redes Neurais) e regras simbólicas na identificação de promotores de *E. coli*. A rede empregada denominada KBANN (do inglês *Knowledge Based Neural Network*), utiliza regras proporcionais 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 (Souto et al. 2003). As regiões consenso correspondem ao TATA-box e a seqüência TTGACA, que se encontram aproximadamente -10 e -35 como descrito na seção 2.3.

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 (Souto et al. 2003).

No treinamento do classificador, as instâncias com promotores foram alinhadas de forma que a região promotora ficasse sete nucleotídeos à direita da janela de entrada, a qual possuía 57 nucleotídeos. A codificação dos nucleotídeos para a RN se deu de forma canônica de quatro *bits*. Nos experimentos conduzidos pelos autores, os resultados obtidos pela rede KBANN foram superiores aos de uma rede MLP, de

uma Árvore de Decisão induzida pelo algoritmo ID3, do algoritmo k-vizinhos mais próximos e com a técnica apresentada por O'Neill and Chiafari (1989).

Outro sistema baseado em RNs, chamado de NNPP (do inglês *Neural Network Promoter Prediction*), foi proposto por Reese, Harris, and Eeckman (1996). Esse sistema se baseia no reconhecimento de dois sinais específicos contidos na região promotora: o TATA-box e o *Inr*[12], assim como a distância entre eles. O sistema utiliza três redes neurais *time-delay* (TDNN do inglês *Time Delay Neural Network*). Uma rede reconhece o TATA-box e a outra o *Inr*. A terceira rede neural combina as saídas das duas redes neurais anteriores, através das distâncias entre o TATA-box e o *Inr*. O sistema atinge uma sensibilidade de predição, com 550 nucleotídeos, de aproximadamente 0,75. Esses resultados são da base de dados da *Drosophila melanogaster* mas podem ser utilizados para organismos procariotos.

Diferentemente dos trabalhos anteriores, Pedersen et al. (1996) treinaram um modelo HMM (do inglês *Hidden Markov Model*) com um conjunto de 166 seqüências promotoras de *E. coli* com a base de dados de Lisser and Margalit (1993). A ênfase do trabalho foi a análise do fato dos promotores serem divididos em classes de acordo com o fator de transcrição σ que os reconhecem. Os resultados mostram que o HMM apresenta um excelente índice de classificação para promotores desconhecidos com respeito à classe σ, além de conseguir "aprender" a estrutura seqüencial presente em promotores procarióticos (Pedersen et al. 1996).

Os HMMs tornaram-se a técnica de Aprendizado de Máquina bastante utilizada no estudo de promotores (Oppon 2000). Essa preferência está firmada na hipótese de que regiões características de promotores, relevantes para que a RNA polimerase se direcione corretamente ao sítio +1, devem se apresentar conservadas entre os promotores de um genoma ou até mesmo entre os promotores de genomas de organismos

[12]chamado também de iniciador (*initiator*) o *Inr* é uma seqüência menos conservada e que possui, portanto, um sinal mais fraco que o TATA-box (Reese 2000)

próximos evolutivamente (Oppon 2000). Além disso, sua aplicação não precisa de um alinhamento prévio das seqüências, e se mostra melhor que os métodos estatísticos mais simples (Pedersen et al. 1996), pois esses falham em considerar as posições ao longo da cadeia como estatisticamente independentes (Souto et al. 2003).

Oppon (2000) utilizou bases de dados de *E. coli, B. subtilis* e *Mycobacteria*. Os resultados obtidos foram os seguintes: 90% de verdadeiros-positivos e 6% de falsos-positivos para *E. coli*; 90% de verdadeiros-positivos e em torno de 3% falsos-positivos para *B. subtilis*. Em contraste, na base de dados dos promotores de *Mycobacteria*, cujos dados eram insuficientes e inseguros, foi obtido uma taxa alta de falso-positivo (13%). Um trabalho proximamente relacionado ao de Oppon (2000) é o de Neves and Lemke (2005). Em seus resultados, eles obtiveram, para base de dados de *E. coli*, um percentual de 95% de acurácia no reconhecimento enquanto que a base de dados *B. subtilis* obteve um percentual de 78% de predição correta.

Recentemente, Gordon et al. (2003) propuseram um método de reconhecimento de promotores utilizando máquinas de vetores de suporte. Para esse fim, foram analisadas 669 regiões de promotores de *E. coli* e dois conjuntos de dados com 709 não-promotores em cada um deles. Todas as regiões de promotores selecionadas contêm o fator σ^{70} com seus sítio de início de transcrição presente. Em seus resultados, eles atingem uma média de erro de 16,5% com os promotores e 18,6% com os não-promotores (Gordon et al. 2003).

A estabilidade do fragmento do DNA é uma propriedade que depende de sua seqüência e depende, principalmente, da soma da interações entre os constituintes dinucleotídeos. Kanhere and Bansal (2005) analisam um método de predição de promotores baseado nas diferenças de estabilidade das regiões promotoras e não-promotoras. Foi atingindo melhores resultados na predição de bacilos quando comparados a outros métodos (NNPP, RN), mas ainda necessitam reduzir a quantidade de falsos-positivos (Kanhere and Bansal 2005).

2.5 Considerações Finais

Como visto na seção anterior, desde 1983, com o trabalho de Hawley and McClure (1983), o problema de predição de promotores de organismos procariotos vem sendo abordado. Apenas em 1989, com o trabalho de O'Neill and Chiafari (1989), informações sobre regiões consenso das seqüências promotoras foram utilizadas como auxiliares na resolução do problema. A partir daí, até os dias atuais a grande maioria dos sistemas desenvolvidos para a predição de promotores procariotos usam informações relativas, por exemplo, aos hexâmeros -10 e -35 (O'Neill 1991; O'Neil 1992; Delemer and Zhou 1991; Towell, Shavlik, and Noordewier 1990). Um dos principais problemas com todos os sistemas revisados é a alta taxa de falsos-positivos que eles apresentam (O'Neill and Chiafari 1989). Na verdade, diminuir a taxa de falsos-positivos e, ao mesmo tempo, aumentar a taxa de verdadeiros-positivos, é um dos principais desafios na área de predição (*in silico*) de promotores (Pedersen et al. 1996; Baldi and Brunak 2001).

Dos trabalhos revisados, o mais proximamente relacionado ao nosso é o de Towell, Shavlik, and Noordewier (1990). Usamos esse trabalho como diretriz na construção da nossa base de dados de *B. subtilis*. Também, como eles, fizemos uma comparação entre vários métodos de aprendizado de máquina para a base em questão. Os outros trabalhos revisados, em geral, comparam duas ou três técnicas (Towell, Shavlik, and Noordewier 1990; Oppon 2000; Gordon et al. 2003). Além disso, quase todos os trabalhos, citados anteriormente, não fornecem informações sobre como a base foi construída: escolha dos promotores (hipotéticos e/ou determinados experimentalmente), como os não-promotores foram definidos, a similaridade entre os promotores e não-promotores, entre outras. Este tipo de informação é fundamental para se entender os resultados obtidos com os métodos de aprendizado de máquina.

Capítulo 3

Aprendizado de Máquina

O objetivo deste capítulo é descrever resumidamente, os algoritmos de Aprendizado de Máquina que serão utilizados nessa dissertação: k-vizinhos mais próximos, *naive bayes*, árvores de decisão, árvores de decisão parcial, redes neurais do tipo *multi-layer perceptron, voted perceptron,* máquina de vetores suporte, modelos ocultos de Markov, *bagging* e *boosting*.

3.1 Introdução

Segundo Monard and Baranauskas (2003), Aprendizado de Máquina (AM) é uma área da Inteligência Artificial (IA) cujo objetivo é o desenvolvimento de técnicas computacionais sobre o aprendizado bem como a construção de sistemas capazes de adquirir conhecimento de forma automática. Um sistema de aprendizado é um programa de computador que toma decisões baseado em experiências acumuladas através da solução bem sucedida de problemas anteriores (Monard and Baranauskas 2003).

As técnicas de AM podem ser divididas em dois tipos: aprendizado supervisionado e aprendizado não-supervisionado. Uma técnica de AM pode ser considerada supervisionado quando um conjunto de exemplos, sendo cada exemplo formado por

um conjunto de atributos de entrada e saída, é recebido pelo indutor antes do processo de aprendizado. Esse fornecimento dos dados de entrada e saída garante a supervisão de cada passo dado pelo algoritmo. Todos os algoritmos utilizados nessa dissertação são exemplos de técnicas de aprendizado de máquina supervisionado (que serão revisados nas Seções 3.2 a 3.8).

Já o aprendizado não-supervisionado acontece quando, para cada exemplo, apenas os atributos de entrada estão disponíveis. Esse tipo de técnica é utilizada para encontrar um conjunto de padrões ou tendências (aglomerados) que ajudem no entendimento desses dados. São exemplos de algoritmos de aprendizado não-supervisionado: mapa auto-organizável (SOM) (Haykin 2001), k-médias (do inglês *k-means*) (Haykin 2001) e agrupamento hierárquico (Souto et al. 2003), entre outros.

Uma classe especial de métodos de AM, que também será investigado neste trabalho, são chamados de comitês ou multiclassificadores (ou ainda *ensembles*) (Breiman 1996; Dietterich 2000; Kuncheva 2004). A idéia por trás desses métodos é a de que a combinação das respostas de vários classificadores diferentes, gerados com uma mesma base de dados, produz respostas mais precisas do que os classificadores quando considerados individualmente (Seção 3.9).

3.2 k-Vizinhos mais Próximos

O k-Vizinhos mais Próximos (k-NN do inglês *k-Nearest Neighbor*) é um dos algoritmos de AM mais simples (Mitchell 1997). Este algoritmo assume que todas as instâncias (padrões ou exemplos) correspondem a pontos no espaço n-dimensional (R^n).

No processo de treinamento todas as instâncias do conjunto de treinamento são armazenadas, junto com suas respectivas classes, em uma tabela. A fim de classificar uma nova instância de teste $\mathbf{x}$, as k instâncias da tabela que sejam mais próximas a

x, em termos de distância euclidiana, são recuperadas (inclusive os rótulos de suas classes). A instância **x** será atribuída a classe mais representada entre os k vizinhos.

Embora este método seja muito simples, ele apresenta uma série de problemas, tais como (Kuncheva 2004):

- Computacionalmente não eficiente na fase de uso, caso o conjunto de treinamento seja muito grande.
- Intolerante tanto a atributos ruidosos quanto a atributos redundantes.
- Não há uma medida de similaridade "natural" para atributos nominais.

3.3 *Naive Bayes classifier*

Um outro algoritmo de AM simples, mas bastante efetivo é o classificador bayseano ingênuo (NB, do inglês *Naive Bayes classifier*). Em alguns contextos, como o de classificação de textos, o desempenho desse método tem se mostrado comparáveis aos de redes neurais e árvores de decisão (Mitchell 1997; Witten and Frank 2000).

Formalmente, o NB pode ser definido como a seguir. Seja **x** a instância que queremos classificar, e c_i sua possível classe. A fim de classificar **x**, temos que saber qual é a probabilidade de **x** pertencer à classe c_i. Isto pode ser colocado na forma de regra de Bayes :

$$p(c_i|\mathbf{x}) = \frac{p(\mathbf{x}|c_i) * p(c_i)}{p(\mathbf{x})} \tag{3.1}$$

A probabilidade de cada classe c_i pode ser estimada a partir dos dados do conjunto de treinamento. Para tornar o cálculo de $p(\mathbf{x}|c_i)$ menos computacionalmente intensivo, é assumido que os valores dos atributos são independentes uns dos outros dados a classe (suposição *naive*):

$$p(\mathbf{x}|c_i) = \prod_{j=1}^{d} p(\mathbf{x}_j|c_i) \tag{3.2}$$

Embora a suposição acima não seja verdadeira para a maioria dos casos, como já mencionado, o NB apresenta um bom desempenho para várias tarefas (Mitchell 1997).

3.4 Árvores de Decisão

Segundo Monard and Baranauskas (2003), uma Árvore de Decisão (AD) é uma estrutura de dados definida recursivamente como:

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

A Figura 3.1 é um exemplo ilustrativo de um AD para o problema de reconhecimento de promotores (cada instância é uma seqüência de 112 nucleotídeos). Na figura, cada elipse é um teste em um atributo para o conjunto de dados. Cada retângulo representa uma classe, ou seja, promotor (P) e não-promotor (NP). Para classificar uma nova instância, basta começar pela raiz, seguindo cada teste até que uma folha seja alcançada.

É fácil gerar um conjunto de regras diretamente de uma árvore de decisão: cada regra tem seu início na raiz da árvore e caminha-se até uma de suas folhas (Witten and Frank 2000; Monard and Baranauskas 2003). De fato, esta é uma característica importante, visto que especialistas humanos podem analisar em conjunto de regras aprendidas por uma AD e determinar se o modelo aprendido é plausível, dadas as

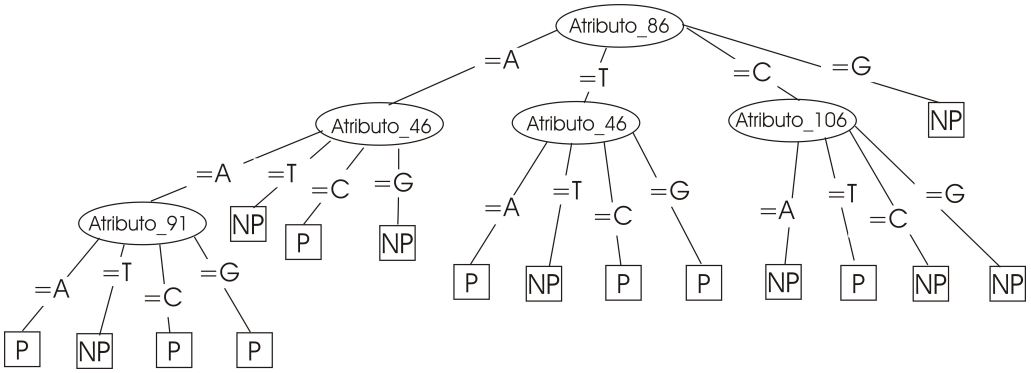

Figura 3.1: Predição de promotores utilizando Árvores de Decisão

restrições do mundo real (Witten and Frank 2000; Monard and Baranauskas 2003; Mitchell 1997).

Uma outra alternativa para a abordagem de geração de regras a partir de AD, é chamada de PART (do inglês, *Partial Decision Trees*) (Witten and Frank 2000). Para a construção de uma única regra, uma AD é construída para o conjunto de dados corrente. A folha com a maior cobertura (taxa de classificação correta) é transformada em uma regra (na verdade, o caminho até a folha). Esta árvore é descartada. As instâncias cobertas por essa regra são retiradas do conjunto de treinamento. Esse novo conjunto é, então, usado para criar uma nova AD é a respectiva regra.

3.5 Redes *Multi-layer Perceptron*

Nesta dissertação, usaremos um tipo de redes neurais conhecido como *Multi-layer Perceptron* (MLP) - Figura 3.2, treinadas com o algoritmo *backpropagation* (Haykin 2001). De fato, MLPs com *backpropagation* têm sido em dos modelos de redes neurais mais usadas em aplicação do mundo real (Haykin 2001), inclusive em problema de Biologia Molecular (Baldi and Brunak 2001; Souto et al. 2003).

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 po 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 3.2 ilustra uma típica RN com mais de uma camada, chamada RN multi-camadas.

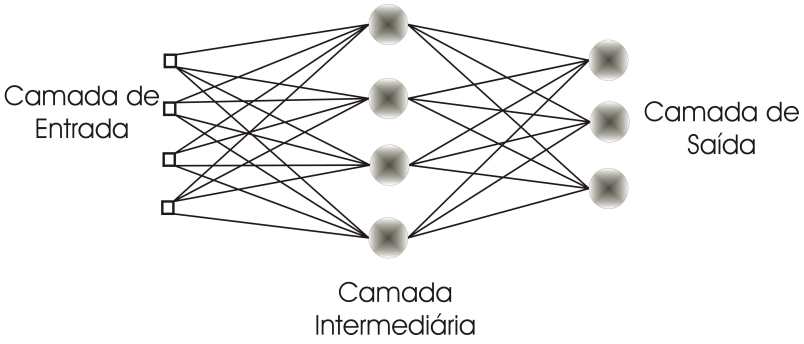

Figura 3.2: Rede Neural do tipo *Multi-Layer Perceptron*

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 (representado 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[1] (Minsky and 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, Hinton, and Williams 1986). O tipo de RN multi-camada usada com o backpropagation é geralmente denominada de rede Multi-Layer Perceptron (MLP) (Rumelhart, Hinton, and Williams 1986).

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.

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.

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

3.6 Máquinas de Vetores Suporte

As Máquinas de Vetores Suporte (SVM, do inglês *Support Vector Machine*) constituem uma técnica de AM que vem recebendo grande atenção no últimos anos (Hearst and et al. 1998). Entre as principais características que popularizaram seu uso em Bioinformática estão sua boa capacidade de generalização e robustez diante de dados e grande dimensão, como os presentes em grande parte de aplicações envolvendo o reconhecimento de promotores e análise de dados e expressão gênica (Souto et al. 2003).

As SVMs são baseadas no princípio da minimização do risco estrutural da teoria de Aprendizado Estatístico (Vapnik 1995). A idéia é encontrar um hiperplano que separe as instâncias positivas (exemplo: promotores) das negativas (exemplo: não-promotores) no conjunto de treinamento, ao mesmo tempo em que se maximiza a distância do hiperplano em relação a elas.

A Figura 3.3 ilustra o hiperplano com máxima margem de separação para um exemplo de problema de classificação em duas dimensões.

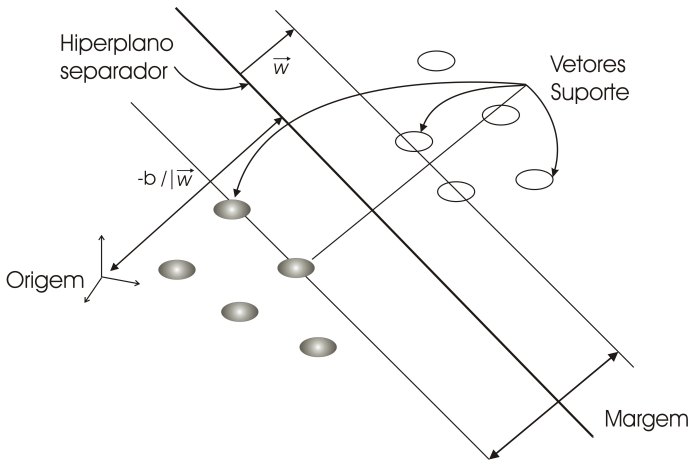

Figura 3.3: SVM linear visto num plano de duas dimensões

O vetor **w** representa a direção do hiperplano separador, o qual é determinado pelos vetores suporte. A fim de lidar com classes não-linearmente separáveis, uma

função *kernel* pode ser usada para mapear o conjunto de dados originais para uma dimensão mais alta, onde se espera que os dados se tornem linearmente separáveis. Uma nova instância de teste $\mathbf{z}$ é classificada de acordo com o sinal da seguinte função:

$$f(\mathbf{z}) = \sum \lambda_i y_i K(\mathbf{x}_i, \mathbf{z}) + b \tag{3.3}$$

em que $\mathbf{x}_i$ pertence ao conjunto de vetores suporte, e $y_i é a classe de \mathbf{x}_i$, λ_i é o peso de $\mathbf{x}_i$, K é a função *kernel*, e b é o bias. Nesta dissertação, consideramos SVM com *kernel* polinomial:

$$K(\mathbf{x}_i, \mathbf{z}) = (\mathbf{x}_i * \mathbf{z} + 1)^p \tag{3.4}$$

em que p é o grau do polinômio.

3.7 *Voted Perceptron*

Um outro método proximamente relacionado às SVMs é o *Voted Perceptron* (VP) (Freund and Schapire 1998; Witten and Frank 2000). Este método é baseado no algoritmo clássico do *perceptron* proposto por Rosenblatt (1958).

No algoritmo de treinamento proposto por Rosenblatt (1958), o vetor de pesos $\mathbf{w}$ da rede é inicializado com zero. A classse de uma nova instância $\mathbf{x}$ é prevista como $\hat{y} = sinal(\mathbf{wx})$ em que a função sinal retorna $+1$ se $\mathbf{wx} \geq 0$ e -1, caso contrário. Se o valor previsto difere do rótulo da classe y, o vetor de pesos é atualizado como $\mathbf{w} = \mathbf{w} + y\mathbf{x}$. Se a classe é prevista corretamente, o vetor de pesos permanece inalterado. Esse processo é repetido para todas as instâncias do conjunto de treinamento, até que seja encontrado um vetor de pesos que classifique corretamente todas as instâncias.

No caso do *Voted Perceptron*, são mantidos todos os vetores de pesos (perceptrons intermediários) diferentes, gerados durante o treinamento (ou seja, cada vez que acontece um erro para uma instância de treinamento).

Para cada um desses vetores (*perceptrons* intermediários), conta-se quantas iterações ele "sobreviveu" até que um novo erro ocorra. Este contador é referenciado

como o "peso"do *perceptron* intermediário.

Na fase de uso, a classificação final é calculada baseada na predição individual de cada *perceptron* intermediário que é ponderado pelo respectivo peso.

Treinamento

Entrada:

conjunto de treinamento $\{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_m, y_m)\}$

número de iterações T

Saída:

lista de vetores de pesos $\{(\mathbf{w}_1, c_1), ..., (\mathbf{w}_k, c_k)\}$

Inicialize k:=0,$\mathbf{w}_1 := 0, c_1 := 0$.

Repita T vezes:

Para i=1,...,m:

* Faça a predição: ŷ:= $sinal(\mathbf{w}_k \ . \ \mathbf{x}_i)$

* Se ŷ=y então $c_k := c_k + 1$.

senão $\mathbf{w}_{k+1} := \mathbf{w}_k + y_i\mathbf{x}_i$;

$c_{k+1} := 1$;

k:=k+1.(3.5)

Predição

Dados :

uma lista de vetores de pesos: $\{(\mathbf{w}_1, c_1), ..., (\mathbf{w}_k, c_k)\}$

instância: $\mathbf{x}$

A predição da nova instância $\hat{y}$ será a seguinte :

$$s = \sum_{i=1}^{k} c_i \ sinal(\mathbf{w}_i \ . \ \mathbf{x}); \hat{y} = sinal(s) \tag{3.6}$$

3.8 Modelo Oculto de Markov

O método de Modelo Oculto de Markov (HMM do inglês *Hidden Markov Model*) é um processo estocástico definido por:

- Um conjunto de S estados.
- Um alfabeto A de m símbolos.
- Uma matriz de probabilidade de transição $T = (t_{ij})$.
- Uma matriz de probabilidade de emissão $E = (e_i x)$.

O sistema evolui de estado a estado aleatoriamente, enquanto emite símbolos do alfabeto. Quando o sistema está num estado i, ele tem a probabilidade t_{ij} de mudar para o estado j, e a probabilidade $e_i x$ de emitir o símbolo X (Pedersen et al. 1996).

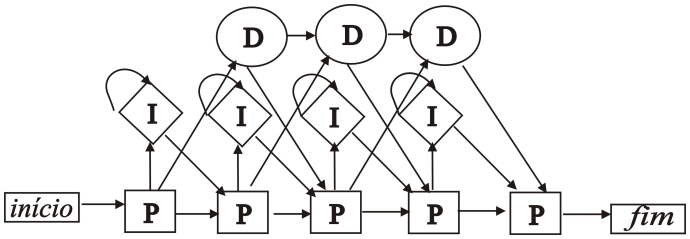

Figura 3.4: Estrutura do HMM para seqüências biológicas. Arquitetura apresentada com os cinco estados: início, principal (P), inserção (I), deleção (D) e fim

Uma das arquiteturas criada para aplicações em Biologia Molecular foi apresentada por Krogh, Mian, and Haussler (1994) com a arquitetura padrão da esquerda para direita. O alfabeto A tem $m = 4$ símbolos, um de cada nucleotídeo (A, T, C e G). O método HMM possui dois estados especiais, início e fim, que indicam o início e fim de uma cadeia respectivamente. O sistema sempre se encontra inicialmente no estado início e passa de estado em estado até o estado fim. A cada transição é emitido um símbolo de acordo com as probabilidades apresentadas na

matriz de probabilidade de emissão (Neves and Lemke 2005). Os estados principal e de inserção emitem um nucleotídeo, enquanto o estado de deleção não emite símbolos.

Portanto, há três classes de estados no conjunto S, além dos estados de início e fim: o estado principal (m), de deleção (d) e de inserção (i) com S = {início, $\mathrm{m}_1, ...m_N, i_1, ...i_{N+1}, d_1, ..., d_N, fim\}$ em que N é o tamanho do modelo, normalmente igual a média dos comprimentos das seqüências.

Existem vários algoritmos baseados em programação dinâmica que são utilizados para dar suporte à teoria HMM. Um deles é o algoritmo *forward-backward* determina a probabilidade de uma seqüência observada ter sido gerada por um modelo HMM. Esse cálculo é utilizado para a tarefa de reconhecimento de seqüências (Neves and Lemke 2005).

3.9 Sistemas de Multiclassificação

A principal idéia em usar técnicas de construção de multi-classificadores é a de que a combinação de classificadores base (individuais) pode levar a uma melhora na precisão e estabilidade dos resultados dos algoritmos de classificação (Witten and Frank 2000). Essa idéia conduz às seguintes questões: como gerar classificadores diferentes? Como combinar esses diferentes classificadores? A respeito da primeira pergunta, diversas abordagens foram propostas, como por exemplo, introduzir instabilidades artificiais em algoritmos de classificação. Por exemplo, um determinado algoritmo de aprendizado pode ser executado diversas vezes, cada vez com um subconjunto diferente de exemplos. Essa técnica é mais bem aplicada para classificadores instáveis, tais como Árvores de Decisão e Redes Neurais (Witten and Frank 2000). Com relação à segunda questão, a maneira mais simples de se fazer isso, no caso de classificação é por meio de uma votação (ponderada ou não).

Os sistemas de multiclassificação (também denominado comitê, combinador, multi-estratégia de aprendizado e integrador) podem ser definidos como um conjunto de classificadores cujas decisões individuais são combinadas de alguma maneira de modo a classificar novas instâncias (Dietterich 2000). Diversos estudos empíricos vêm mostrando que o desempenho de multiclassificadores é superior àquele obtido pelos classificadores base usados em sua formação (Dietterich 1998; Dietterich 2000).

A idéia intuitiva da multiclassificação é que nenhum método ou sistema pode ser uniformemente superior a qualquer outro, e que a integração de diversos métodos individuais melhora o desempenho do classificador final (Dietterich 2000). Portanto, um sistema de multiclassificação deve ter um melhor desempenho global do que os classificadores base usados em sua formação.

A eficácia de métodos de multiclassificação está altamente relacionada à independência do erro cometido pelos classificadores base. Seu desempenho, portanto, depende fortemente da exatidão e da diversidade dos classificadores usados para formá-los (Opitz and Mach 1999). Ou seja, combinar a saída de diversos classificadores é útil somente se há diferenças entre eles (por exemplo, em um caso extremo, combinar diversos classificadores idênticos não produz ganho algum).

3.9.1 Bagging

Bagging (*Boostrap AGGregatING*) foi proposto por Breiman (1996), baseado na amostragem *bootstrap* (Efron and Tibshirani 1993). Neste método, são gerados vários subconjuntos diferentes de treinamento a partir de uma amostragem aleatória do conjunto original de dados, com reposição. Então, são induzidos diferentes classificadores a partir de cada um destes subconjuntos de treinamento (Canuto 2001). Os subconjuntos de treinamento têm o mesmo número de instâncias do conjunto original. Porém, algumas instâncias do conjunto original podem aparecer mais de

uma vez, enquanto que outras instâncias podem não ser selecionadas. Esta distinção aleatória entre os vários subconjuntos de treinamento confere diversidade aos modelos de classificação que são obtidos a partir de cada um desses subconjuntos (Kuncheva 2004).

A amostragem com reposição representa uma técnica eficaz para a produção de conjuntos de treinamento representativos do conjunto de treinamento original e, ao mesmo tempo, distintos entre si (Lima 2005). No entanto, a existência de subconjuntos de dados distintos não implica que todos os classificadores induzidos irão generalizar de forma distinta. Para que isso ocorra, os classificadores devem ser instáveis, no sentido de produzirem comportamentos distintos sempre que submetidos a conjuntos de treinamento distintos, como os métodos AD e RNA.

O método *Bagging* busca, portanto, classificadores que sejam instáveis, capazes de generalizar de forma distinta, não necessariamente de forma ótima. Isso porque, mesmo que os componentes não apresentem uma boa capacidade de generalização, a combinação deles tende a generalizar bem. Na verdade, a capacidade de generalização dos classificadores base não pode ser muito ruim, pois a melhora obtida com a combinação deles pode não ser suficiente para garantir um bom desempenho do sistema classificador. Por outro lado, também não precisa ser muito boa, pois neste caso a combinação dos diferentes classificadores base pode não produzir ganho no desempenho de classificação (Lima 2005).

3.9.2 Boosting

O método *Boosting* (Schapire 1990) abrange, na verdade, uma família de diferentes técnicas (como por exemplo, *Arcing*, *Adaboosting*, *MultiBoosting* e *AveBoost*). Em geral, no *Boosting* a escolha é baseada no desempenho dos primeiros classificadores. É necessário que os vários classificadores base do *Boosting* sejam treinados seqüencialmente, visando definir as instâncias que irão constituir os próximos conjuntos de

treinamento (Lima 2005).

Uma das formas de *Boosting* muito difundida e aplicada é o *AdaBoosting* (Freund and Schapire 1996). Neste caso, a probabilidade de escolha de uma instância para compor um conjunto de treinamento depende da contribuição dela para o erro de treinamento dos classificadores já treinados. Isto é, caso uma instância não tenha sido corretamente classificada pelos classificadores já gerados, a probabilidade de escolha desta instância aumenta em relação às demais, quando do treinamento de novos classificadores (Lima 2005). Conseqüentemente, esta instância terá uma chance maior de ser escolhida para compor o conjunto de treinamento do próximo classificador a ser gerado. Assim, pode-se observar que, ao contrário do que ocorre no *Bagging*, apenas o primeiro classificador base da seqüência do método é treinado a partir de uma amostragem aleatória do conjunto de dados original.

O *AdaBoosting* também pode seguir um outro modelo para a construção de multiclassificadores, no qual todas as instâncias, e seus pesos associados, fazem parte do conjunto de treinamento dos classificadores base. Esta abordagem tem a vantagem de que todas as instâncias, com pouca ou muita representatividade (de acordo com o peso correspondente), são incorporadas aos conjuntos de treinamento.

Capítulo 4

Métodos e Experimentos

Neste capítulo, descrevemos a metodologia de avaliação, como também a maneira como conduzimos os experimentos. Além disso, mostramos todas as etapas que usamos na construção da base de dados - uma das principais contribuições desta dissertação.

4.1 Base de dados

A fim de se construir nossas bases de dados, no início do desenvolvimento desta dissertação pesquisamos espécies de dois gêneros de bactérias: os Bacilos e as Pseudomonas. Em ambos os contextos, após uma análise exaustiva de amostras de promotores disponíveis nos banco de dados mundiais de genes, verificamos que poucas delas eram experimentalmente comprovadas. Particularmente, no caso de Pseudomonas, não encontramos um número suficiente de seqüências promotores (aproximadamente 45 promotores) que viabilizasse a construção de uma base de dados para algoritmos de Aprendizado de Máquina.

No caso dos Bacilos, dependendo da espécie considerada, encontramos uma quantidade variada de seqüências promotoras. Por exemplo, no caso da espécie *B. subtilis*, identificamos 112 seqüências promotoras experimentalmente comprovadas, o que já

viabiliza a construção de uma base de dados[1]. Já a segunda espécie com a maior quantidade de promotores experimentalmente comprovados foi o *B. thuringiensis* com apenas 14 promotores.

Nesse ponto, é importante chamar a atenção ao fato de que em geral os programas desenvolvidos para a tarefa de predição de promotores são projetados para determinadas espécies (por exemplo, para a bactéria *E. coli*) e não para todo um gênero. Uma das razões para isto é a tentativa de se controlar a sensibilidade e a especificidade do sistema. Motivado por isto, decidimos que a nossa base de dados principal seria formada apenas por promotores da espécie *B. subtilis*.

Resolvemos também testar uma outra base de dados, chamada base de dados híbrida, na qual juntamos as seqüências de promotores de várias espécies de bacilos que fossem proximamente relacionadas do ponto de vista evolutivo. A idéia é a de que espécies próximas evolutivamente teriam suas seqüências de promotores muito similares, ou seja, elas poderiam ser vistas como um grupo homogêneo. Por exemplo, a Figura[2] 4.1 ilustra a árvore filogenética[3] de várias espécies de bacilos.

Com base na Figura 4.1, podemos observar três grandes grupos: o primeiro grupo formado por seqüências promotoras do *B. subtilis* (total de 112 seqüências); o segundo grupo formado por seqüências promotoras de *Bacillus thuringiensis, licheniformis, amyloliquefaciens, cereus, megaterium e firmus* (no total de 30 seqüências); e o terceiro grupo, formado por seqüências de *Bacillus pumilus, brevis e stearothermophilus* (no total de 14 seqüências). Neste caso, optamos por construir uma base híbrida apenas com os promotores das espécies do segundo grupo. No entanto, como desempenho (taxa de classificação correta) dos métodos AM para essa base híbrida

[1] uma das base de dados mais conhecida na literatura é a da bactéria *E. coli* com 57 promotores(Towell, Shavlik, and Noordewier 1990)

[2] retirada de Joung and Côté (2002)

[3] Representação das relações de parentesco entre grupos de organismo, visando estabelecer uma classificação sistemática e um conhecimento da evolução dos grupos (Matioli 2001).

foi, em geral, inferior a 50%. Acreditamos que esse resultado ocorreu devido a pequena quantidade de promotores obtidos além de estarmos manipulando espécies diferentes e portanto, seus padrões são bastante divergentes entre si. Decidimos então, não mais discuti-la nesta dissertação.

Na próxima seção apresentaremos mais detalhes sobre a metodologia utilizada na construção da base de dados, incluindo também a escolha das instâncias representando os não-promotores.

4.2 Construção da base de dados do *B. subtilis*

Construímos uma base de dados utilizando os mesmos parâmetros usados na elaboração da base de dados[4] da *E. coli*, usada originalmente por Towell, Shavlik, and Noordewier (1990). Essa base de *E. coli* contém 53 exemplos de promotores e 53 exemplos de não-promotores obtidos de uma compilação produzida por Harley and Reynolds (1987). De acordo com Harley and Reynolds (1987), exemplos de não-promotores foram obtidos através de exemplos derivados de fragmentos de seqüências de um bacteriófago de *E. coli* T7. Cada exemplo, tanto positivo como negativo, é composto por 57 atributos.

No caso da base de dados do *B. subtilis*, consideramos inicialmente, apenas os dados experimentalmente comprovados, apresentados na compilação produzidos por Helmann (1995). Calculamos a média de comprimento dessas seqüências, que foi de aproximadamente 117 nucleotídeos. Fixamos um comprimento de 117 nucleotídeos para as nossas seqüências de promotores. Seqüências menores que esse número, e ditas hipotéticas pelo autor, foram descartadas.

Seqüências de promotores iguais ou maiores a 117, por outro lado, foram preservadas na base de dados. Neste último caso, as seqüências foram primeiramente

[4]Esta base de dados está disponível em (ftp://ftp.ics.uci.edu/pub/machine/molecular-biology/promotergene-sequences/)(último acesso: junho/2003).

Figura 4.1: Relacionamentos Filogenéticos da espécie Bacillus entre padrões de restrição pelo gene 16S do RNAr

cortadas em suas regiões *upstream*, de modo que seu comprimento final tivesse 117 nucleotídeos - esta estratégia foi usada para preservar a região do promotor do *B. subtilis* que é normalmente encontrada na região entre -100 (*downstream*) e $+15$ (*upstream*) do gene. No final do processo obtivemos 112 seqüências de promotores retiradas das 236 seqüências originalmente apresentadas por Helmann (1995).

Selecionamos 112 seqüências contínuas de 117 nucleotídeos de um genoma de um bacteriófago PZA do *B. subtilis* para criar os não-promotores para nossa base de dados (Paces et al. 1986), em que provavelmente, não existe seqüência promotora identificada. Estas seqüências foram escolhidas de tal forma que se obtivesse o maior grau de similaridade possível entre cada uma das seqüências dos promotores.

A média de similaridade entre os promotoras e não-promotoras resultou em 27%. No caso da base de dados de *E. coli* (Towell, Shavlik, and Noordewier 1990), essa média é de 24%. Resumindo, obtivemos uma base de dados de 112 promotores e 112 não-promotores, cada uma possuindo 117 nucleotídeos entre eles.

4.3 *Cross-validation*

A comparação de dois métodos de aprendizado supervisionado é realizada, tradicionalmente, analisando a significância estatística da diferença entre a media da taxa do erro de classificação, em conjuntos independentes de teste, dos métodos avaliados. Para esta avaliação, diversos conjuntos (distintos) de dados são necessários. Entretanto, a quantidade de dados disponíveis é normalmente limitada. Uma forma de superar este problema é dividir a base de dados em conjuntos de treinamento e de teste pelo uso do procedimento do *k-fold cross validation* (Mitchell 1997; Dietterich 1998).

Esse procedimento funciona da seguinte maneira: o conjunto de dados formado

por n instâncias é dividido aleatoriamente em k partições (*folds*) mutuamente exclusivas, cada uma contendo aproximadamente n/k instâncias. Então, o treinamento é realizado em k etapas, cada vez usando-se uma partição diferente como conjunto de teste e todas as outras $k - 1$ partições restantes como conjunto de treinamento (Mitchell 1997; Dietterich 1998; Witten and Frank 2000). O erro na validação cruzada é a média dos erros calculados em cada uma das k partições usadas como conjuntos de teste. Aplicando diferentes métodos de AM às mesmas partições, a significância estatística das diferenças entre os métodos pode ser medida (Mitchell 1997; Dietterich 1998; Witten and Frank 2000).

Tentamos, inicialmente, utilizar este método com 10 e 5 partições (10-*fold cross-validation*) mas não obtivemos resultados significativos. Nesta dissertação, para utilizarmos a maior quantidade de dados possível no treinamento, utilizamos o método *leave-one-out* (Mitchell 1997; Dietterich 1998; Witten and Frank 2000), que é uma forma especial de *k-fold cross validation*, em que k é o número de instâncias do conjunto de dados. Neste método, somente uma instância é inserida no conjunto de teste deixando todas as outras no conjunto de treinamento. Por esse motivo, cada instância testada será julgada contendo 0 ou 1, ou seja, terá 100% de erro ou 100% de acerto. Em nosso caso, k é 224, em que k é a quantidade total de seqüências promotoras e não-promotoras da base de dados.

O método *leave-one-out* foi escolhido por possibilitar a utilização de uma maior quantidade de dados (já que possuímos uma base de dados relativamente pequena) para o treinamento e assim conseguimos um melhor desempenho na criação do classificador de promotores. Outra vantagem deste método é seu procedimento determinístico ou seja, não há amostragem aleatórias envolvidas. Isso significa dizer que o mesmo resultado será obtido, repetindo, por exemplo, 10 vezes ou tudo de uma vez.

4.4 Teste de hipótese

Ao comparar dois classificadores com, aproximadamente a mesma taxa de erro, o desvio padrão pode ser visto como uma imagem da robustez do algoritmo: se os erros calculados sobre diferentes conjuntos de teste, utilizando diferentes conjuntos de treinamento são muito diferentes de um experimento para outro, então o classificador não é robusto a mudanças no conjunto de treinamento proveniente de uma mesma distribuição (Mitchell 1997; Dietterich 1998).

Se tanto a taxa média de erro como o desvio padrão de dois classificadores diferentes apresentarem valores próximos, é difícil afirmar se há diferença entre eles, ou decidir qual dos dois é o melhor. É uma prática comum na área de AM o uso do teste de hipótese (Mitchell 1997; Dietterich 1998). Uma hipótese estatística é uma afirmativa a respeito de um parâmetro de uma distribuição de probabilidade. Teste de hipótese é uma regra de decisão para aceitar ou rejeitar uma hipótese estatística com base nos elementos amostrais. A finalidade dos testes é verificar se as variações que são encontradas nas amostras são casuais ou verdadeiras. Geralmente é interpretado como a chance de detectar uma real diferença, entre as médias, ou seja, detectar a diferença se ela realmente existir.

Chama-se hipótese nula ($H0$) a hipótese estatística a ser testada e, por H1, a hipótese alternativa. A hipótese nula expressa uma igualdade, enquanto que a hipótese alternativa é dada por uma desigualdade ($\neq$, $<$, $>$). O teste de hipótese irá indicar a probabilidade de decisão correta baseada na hipótese alternativa.

Conforme o valor do parâmetro, a hipótese nula será aceita ou rejeitada a partir de procedimentos estatísticos. O valor de α, que também é conhecido como índice de significância, é a probabilidade de errar ao se rejeitar a hipótese nula, quando, na verdade, ela é verdadeira. Quando dois classificadores diferentes são comparados, deseja-se que tal probabilidade seja mínima. Os índices de significância clássicos são

0,05 e 0,01. Ou seja, trabalha-se com chances de errar de 5% ou de 1%. Para se realizar um teste de significância, é preciso conhecer a distribuição de probabilidade a ser aplicada. Definido o teste, calcula-se a sua estatística para os dados que se tem, e compara-se o mesmo ao valor crítico da estatística da distribuição, segundo a significância desejada. Se o valor calculado for menor ou igual ao valor crítico, aceita-se a hipótese nula; se for maior, rejeita-se. Se a diferença de médias for considerada como distribuição normal, pode-se recorrer à estatística z ou à estatística t para se fazer o teste. O teste z exige que se conheça o desvio padrão populacional, e isso não ocorre na maioria dos casos. Se não há esse conhecimento, recorre-se à distribuição t (na verdade, o teste t é o mais utilizado, pois raramente se conhece o desvio padrão populacional).

4.5 Experimentos

Nossos experimentos foram realizados utilizando a base de dados construída para algoritmos de aprendizado de máquina. Devido à metodologia *leave-one-out*, cada técnica foi executada 224 vezes. Os melhores parâmetros de cada um dos métodos individuais foram escolhidos de acordo com o seguinte procedimento: para um algoritmo com somente um parâmetro a ser configurado, um valor inicial para tal parâmetro é escolhido e o algoritmo executado. Portanto, experimentos com valor maior e menor que ele são também realizados. Se, com o valor inicialmente escolhido, o classificador obteve os melhores resultados (em termos da média de erro de classificação), então outros experimentos não precisarão mais ser executados. Caso contrário, o mesmo processo é repetido para o valor do parâmetro com o melhor resultado até então, e assim por diante. Esse procedimento naturalmente consome mais tempo com o aumento do número dos parâmetros a serem investigados.

Usando tal procedimento, os seguintes valores dos parâmetros de cada um dos

métodos empregados foram obtidos (os parâmetros não citados foram configurados para os seus próprios valores *default* do Weka (2004) e do Meta-Meme para o HMM elaborado por Grundy et al. (1997)):

- PART: todos os parâmetros foram configurados para os seus valores *default*;
- k-NN: k=8 e a distance Weighting = 1/distance;
- NB: todos os parâmetros foram configurados para os seus valores *default*;
- VP: todos os parâmetros foram configurados para os seus valores *default*;
- AD: todos os parâmetros foram configurados para os seus valores *default*;
- SVM: c=1 e expoente = 4;
- HMM: todos os parâmetros foram configurados para os seus valores *default*;
- *Bagging* para PART, k-NN, AD, NB, VP e SVM: todos os parâmetros dos classificadores de bases foram configurados de forma semelhante aos acima citados. O processo Bagging foi configurado com o número de iterações igual a 100 e todos os outros parâmetros permaneceram em seu *default*.
- *AdaBoosting* para PART, k-NN, AD, NB, VP e SVM: todos os parâmetros dos classificadores de bases foram configurados de forma semelhante aos acima citados. O processo *Adaboosting* foi configurado com o número de iterações igual a 100 e todos os outros parâmetros permaneceram em seu *default*.

Cada um dos métodos de AM apresentados, como já mencionado, foi treinado com a metodologia *leave-one-out*, considerando os melhores parâmetros encontrados. Então, para todos os experimentos, a média da porcentagem de classificação incorreta nos conjuntos de testes independentes foi calculada. Em seguida, essas médias

foram comparadas duas a duas, pelo teste de hipótese (Mitchell 1997; Dietterich 1998).

Capítulo 5

Resultados

Neste capítulo apresentamos e analisamos os resultados obtidos com a aplicação das técnicas de AM e da metodologia previamente definidas ao problema de reconhecimento de promotores de *B. subtilis*. Na Seção 5.1 analisamos o desempenho dos classificadores individuais: k-NN, *Naive bayes classifier* (NB), Árvores de Decisão (AD), Redes Neurais do tipo *Multilayer Perceptron* (RN), PART, *Voted Perceptron* (VP), Máquinas de Vetores Suporte (SVM), e Modelos Ocultos de Markov (HMM). Os resultados referentes ao *Bagging* e ao *Boosting* são apresentados e discutidos na Seção 5.2.

5.1 Classificadores Individuais

5.1.1 Resultados

A Tabela 5.1 apresenta, para cada algoritmo de aprendizado de máquina, a média e o desvio de padrão da porcentagem incorreta (taxa de erro) dos exemplos classificados em conjuntos de testes independentes. Para facilitar sua visualização, esses resultados também são ilustrados na Figura 5.1.

Além da média geral do erro de classificação, para uma discussão mais detalhada sobre o desempenho dos classificadores analisados, é importante considerar a média do erro por classe (Promotor e Não-Promotor). Isto pode ser feito por meio da

Algoritmo	Média	Desvio-Padrão
k-NN	34,82%	47,75%
NB	18,3%	38,76%
AD	30,8%	46,27%
RN	25,0%	43,4%
PART	31,25%	46,46%
VP	32,14%	46,81%
SVM	18,3%	38,76%
HMM	26,7%	44,38%

Tabela 5.1: Média de taxas de erros dos classificadores individuais

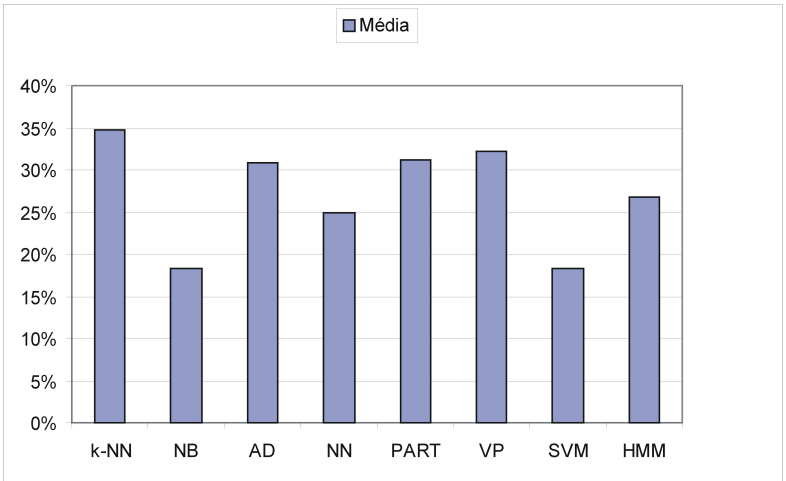

Figura 5.1: Média de taxas de erros dos classificadores individuais

análise de uma matriz de confusão. Por exemplo, a Tabela 5.2 ilustra um exemplo da estrutura de uma matriz de confusão: VP (Verdadeiro-Positivo) denota a média de classificação correta de exemplos de promotores (exemplos positivos); VN (Verdadeiro-Negativo) representa a média da classificação correta de exemplos de não-promotores (exemplos negativos); FP (Falso-Positivo) denota a média de classificação incorreta de exemplos negativos dentro da classe de exemplos positivos; e FN (Falso-Negativo) representa a média dos exemplos positivos classificados incorretamente na classe dos exemplos negativos.

Cada tabela a seguir ilustra a matriz de confusão de um dos classificadores individuais utilizados - k-NN, NB, AD, RN, PART, VP, SVM e HMM.

Verdadeiro/Predito	Promotor	Não-promotor
Promotor	VP	FN
Não-Promotor	FP	VN

Tabela 5.2: Matriz de confusão

Verdadeiro/Predito	Promotor	Não-promotor
Promotor	92,8%	7,2%
Não-Promotor	62,5%	37,5%

Tabela 5.3: Matriz de confusão - k-NN

Verdadeiro/Predito	Promotor	Não-promotor
Promotor	82%	18%
Não-Promotor	19%	81%

Tabela 5.4: Matriz de confusão - NB

Verdadeiro/Predito	Promotor	Não-promotor
Promotor	75%	25%
Não-Promotor	36,6%	63,4%

Tabela 5.5: Matriz de confusão - AD

Verdadeiro/Predito	Promotor	Não-promotor
Promotor	80%	20%
Não-Promotor	30%	70%

Tabela 5.6: Matriz de confusão - RN

Verdadeiro/Predito	Promotor	Não-promotor
Promotor	67%	33%
Não-Promotor	29,5%	70,5%

Tabela 5.7: Matriz de confusão - PART

Verdadeiro/Predito	Promotor	Não-promotor
Promotor	67%	33%
Não-Promotor	31,3%	68,7%

Tabela 5.8: Matriz de confusão - VP

Verdadeiro/Predito	Promotor	Não-promotor
Promotor	76%	24%
Não-Promotor	12,4%	87,5%

Tabela 5.9: Matriz de confusão - SVM

Verdadeiro/Predito	Promotor	Não-promotor
Promotor	82%	28%
Não-Promotor	33%	67%

Tabela 5.10: Matriz de confusão - HMM

5.1.2 Discussão

De acordo com a Tabela 5.1, NB e SVM obtiveram a menor taxa de erro de classificação global (18,30%). A hipótese nula foi rejeitadas em favor da SVM e do NB, em comparação tanto com o k-NN, quanto com a AD e o PART - com $\alpha = 0,05$, em que α é o coeficiente do nível de significância do teste de hipótese.

Do ponto de vista de SVMs, esse resultado é mais uma evidência do sucesso dessa técnica no contexto de Bioinformática (Souto et al. 2003). Em contraste, como os atributos das instâncias na nossa base de dados não são condicionalmente independentes[1] dada a classe (violação da suposição do NB), não esperávamos que esse método tivesse um desempenho superior a todos os outros, com exceção das SVMs.

Entre os classificadores analisados, o k-NN apresentou a maior taxa de erro de classificação global (34,82%). Esse resultado, de certo modo, já era esperado visto que o k-NN não lida bem com instâncias com atributos categóricos (Witten and Frank 2000) - os atributos das instâncias no nosso conjunto de dados são todos do tipo categórico (A, G, C, e T).

Com respeito aos HMMs, uma justificativa para termos obtidos resultados inferiores (26,7%) aos apresentados na literatura (Neves and Lemke 2005) (22%), pode estar relacionado com o grau de similaridade entre seqüências promotoras e não-promotoras que usamos (27%). Quanto maior for este grau, mais difícil se torna o

[1]Por exemplo, a ocorrência dos hexâmeros nas posições -10 e -35 não são eventos independentes.

problema de discriminar entre essas seqüências. Neves and Lemke (2005) não apresentam essa informação. Além disso, a base de dados usada por eles possuía 220 seqüências promotoras (incluindo promotores hipotéticos), enquanto a nossa tem apenas 112 seqüências promotoras (todos experimentalmente comprovados).

Em uma análise mais detalhada dos resultados obtidos (Tabelas 5.3-5.10), podemos observar que o k-NN apresentou a maior sensibilidade (VP) - 92,2%. Em contrapartida, esse mesmo método mostrou a menor especificidade (VN) - 37,5%. Ou seja, na verdade, o k-NN não conseguiu distinguir os não-promotores dos promotores.

De todos os métodos, as SVMs apresentaram a maior especificidade (87,5%), seguida pelo NB (81%) e pelo PART (70,5%). De fato, nesse contexto, as SVMs e o NB mostram-se bastante superiores aos demais métodos - por exemplo, há uma diferença de aproximadamente 10% entre a taxa de VN do NB e do PART.

Ainda com base nas Tabelas 5.3-5.10, uma outra maneira de se comparar os resultados é por meio da análise da relação entre as taxas de VP e FP - Tabela 5.11. Nesse caso, NB e SVM também oferecem um melhor compromisso entre generalização e discriminação - por exemplo, a relação VP/FP do NB (segunda melhor) foi de 4,32, enquanto a de RN (terceira melhor) foi de 2,67. Isto também implica em um melhor controle dos falsos positivos. Essa questão é importante devido a existir, como já mencionado, uma alta probabilidade de se acharem seqüências que não são promotores similares a promotores (falsos positivos).

Também testamos nossa base de dados utilizando o NNPP que está disponível no site ($www.fruitfly.org/seq\_tools/promoter.html$). Todas as predições do NNPP foram executadas com um limiar (*cut-off*) de 0,80 (que é o padrão *default* do NNPP). No caso da nossa base de dados, o NNPP obteve uma predição correta de 107 dos 112 promotores. Porém, para as seqüências não promotoras, ele obteve uma taxa de falso positivo inaceitável de 74,1% (83 das 112 seqüências não promotoras foram preditas como promotoras). Ou seja, para esta base de dados, o método

Algoritmo	VP/FP
k-NN	1,48
NB	4,32
AD	2,05
RN	2,67
PART	2,27
VP	2,14
SVM	6,13
HMM	2,48

Tabela 5.11: Relação Verdadeiro-Positivo *versus* Falso-Positivo

NNPP apresentou uma relação VP/FP de 1,29, o que é bem menor que a taxa apresentada pelo SVM (6,13). De fato, este método obteve melhor performance quando comparado ao NNPP. Uma das razões para que isto tenha ocorrido está no fato de termos construído um classificador específico para a bactéria *B. subtilis*, enquanto que o NNPP é um classificador geral.

5.2 Multiclassificadores: resultados e discussão

As Tabelas 5.12 e 5.13 apresentam, respectivamente, a média e o desvio padrão da porcentagem de classificação incorreta para as técnicas *Bagging* e *Adaboosting*. Os multiclassificadores gerados com essas técnicas usaram como classificadores base o PART, VP, AD e SVM. O k-NN e o NB não foram escolhidos por serem técnicas estáveis em relação a pequenas mudanças no conjunto de treinamento, ou seja, não são apropriados para o *Bagging* e o *Boosting* (Witten and Frank 2000). O mesmo argumento é válido para os HMMs. No caso da RN, não a aplicamos por limitações computacionais[2].

Em relação aos resultados obtidos com a técnica *Bagging* (Tabela 5.12), não foi verificada nenhuma diferença estatisticamente significativa entre os classificadores gerados (α=0,05). Da mesma forma, os classificadores gerados com o *Adaboosting*

[2]o treinamento ultrapassou a capacidade de memória em uma máquina com 1G de memória RAM

Algoritmo	Média	Desvio-Padrão
Bagging PART	21,88%	41,43%
Bagging VP	20,09%	40,16%
Bagging AD	24,55%	43,14%
Bagging SVM	18,30%	44,17%

Tabela 5.12: Taxa de erro da técnica *Bagging*

Algoritmo	Média	Desvio-Padrão
Boosting PART	23,21%	42,31%
Boosting VP	22,77%	42,02%
Boosting AD	18,30%	40,56%
Boosting SVM	18,30%	38,76%

Tabela 5.13: Taxa de erro da técnica *Boosting*

(Tabela 5.13), não apresentaram evidência estatística para se afirmar que existe uma diferença entre eles.

Por outro lado, quando comparamos o desempenho dos classificadores individuais com os gerados pelo *Bagging* (Figura 5.2) e pelo *Adaboosting* (Figure 5.3), podemos observar que houve um ganho significativo com os multiclassificadores em termos de desempenho - com exceção do *Bagging* e AdaBoosting de SVMs que exibiram o mesmo desempenho que as SVMs individuais. Por exemplo, de acordo com as Tabelas 5.1, 5.12 e 5.13, podemos observar que o resultado original obtido com a AD, como método individual, apresentou uma taxa de erro de 30,80%. Então, com o uso do *Bagging*, esta taxa de erro caiu para 24,55%. Aplicando-se o *Adaboosting*, foi obtida uma taxa de erro ainda mais baixa (18,30%) do que aquela conseguida com o *Bagging*.

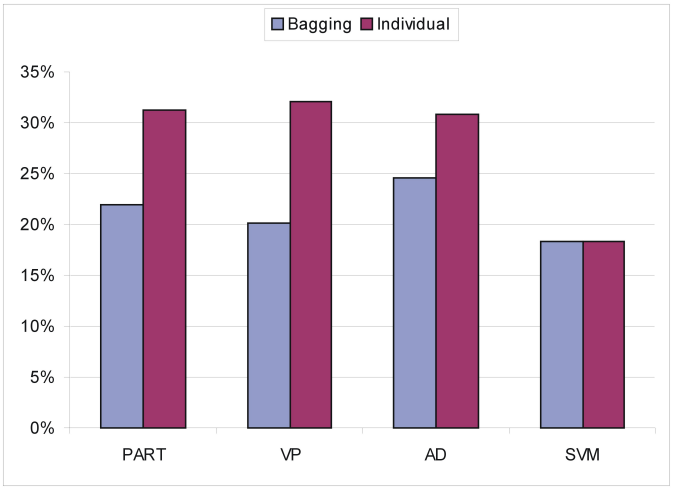

Figura 5.2: Resultados obtidos com o *Bagging* e seus respectivos classificadores base

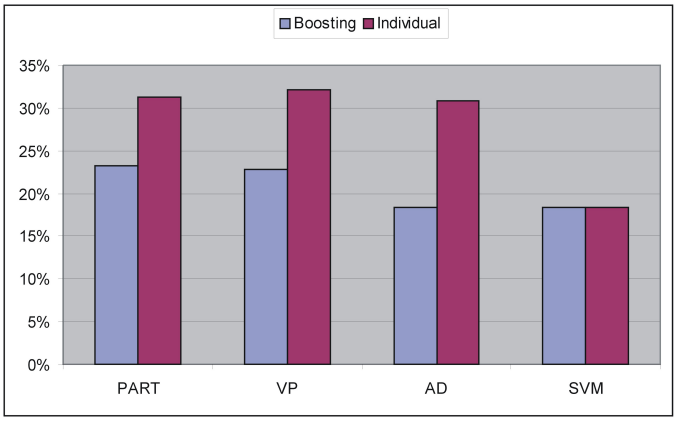

Figura 5.3: Resultados obtidos com o *Boosting* e seus respectivos classificadores base

Capítulo 6

Conclusão e Perspectivas

Neste trabalho, fizemos uma comparação empírica entre várias técnicas de aprendizado de máquina aplicadas à tarefa de predição de promotores de *B. subtilis*. Porém, como uma das principais contribuições desta dissertação, antes de realizarmos os experimentos, construímos uma base de dados de promotores e não-promotores para esse organismo (Monteiro et al. 2005a). Somente espécies comprovadas experimentalmente foram incluídas na base de dados.

Em termos de experimentos, realizamos uma análise comparativa entre oito tipos de classificadores individuais, induzidos por algoritmos que representam diferentes paradigmas de AM (Monteiro et al. 2005a): k-NN, Classificador Bayesiano Ingênuo (NB, do inglês *Naive Bayes Classifier*), Árvores de Decisão (AD), Máquinas de Vetores de Suporte (SVMs), Redes Neurais do tipo VotedPerceptron (VP), Redes Neurais do tipo *Multilayer Perceptron* (RNs), PART e modelos ocultos de Markov (HMM, do inglês *Hidden Markov Models*). Entre eles, SVM e NB apresentaram um desempenho superior aos demais métodos (taxa de erro de 18,3%).

Do ponto de vista de SVMs, como já mencionado, esse resultado é mais uma evidência do sucesso dessa técnica no contexto de Bioinformática (Souto et al. 2003). De fato, esse método obteve melhor desempenho, inclusive quando comparado ao NNPP, um dos métodos bastante conhecido na literatura (Reese 2000) aplicado a

dados de genoma de outras espécies. Uma das razões para que isto tenha ocorrido reside no fato de termos construído um classificador específico para a bactéria *B.subtilis*, enquanto que o NNPP é um classificador geral. No caso dos multiclassificadores foram obtidos resultados iguais ou inferiores aos dos classificadores base, ou seja, o uso dessa técnica não apresentou um ganho em termos de acurácia. Em contraste, como os atributos das instâncias na nossa base de dados não condicionalmente independentes dada a classe, não esperávamos que esse método tivesse um desempenho superior a todos os outros, com exceção das SVMs.

A fim de melhorar o desempenho dos classificadores individuais, também aplicamos sistemas de multiclassicação: *Bagging* e *Adaboosting* (Monteiro et al. 2005b). Nesse contexto, podemos observar que, em geral, foram obtidos resultados superiores ou iguais aos dos classificadores individuais. Os multiclassificadores (tanto o *Bagging* quanto o *Boosting*) que usaram árvores de decisão como classificador base apresentaram o maior ganho. Isso se deve ao fato de árvores de decisão serem classificadores instáveis, favorecendo o *Bagging*, e simples, favorecendo o *Boosting*. Em contraste, ao serem empregadas SVMs como classificador base, os multiclassificadores gerados não apresentaram ganho no desempenho.

Ainda, através de testes parciais, analisamos a base de dados híbrida onde foram testados os mesmos oito classificadores base usados na base de dados *B. subtilis*. Nessa base híbrida, não obtivemos resultados relevantes quando analisados estatisticamente. Mais testes com mudanças dos parâmetros deverão ser realizados para uma melhor verificação.

Foram realizados durante essa dissertação, outros trabalhos além desse, que paralelamente obtivemos resultados e publicações relevantes para a comunidade científica (Tavares et al. 2003; Rocha et al. 2004).

A título de trabalhos futuros, deveremos realizar ainda os seguintes experimentos:

- Analisar e refinar as bases de dados das seqüências promotoras de cada organismo em específico, fazendo testes tanto a nível individual como em grupos.
- Avaliar e implementar outros métodos de AM na predição de promotores que foram citados na revisão de literatura.
- Fazer outros tipos de testes de validação para os métodos utilizados.

Referências Bibliográficas

Antequera, F., and A. Bird. 1993. "Number of CpG islands and genes in human and mouse." *Proc Natl Acad Sci* 90:11995–11999.

Bacillus Subtilis. Livets pigment: om de viktiga hemmolekylerna. `http://www.biol.lu.se/cellorgbiol/membprot/pictures/sm_subtilis_EM.jpg`(acesso em novembro/2005).

Baldi, P., and S Brunak. 2001. *Bioninformatics: The Machine Learning Approach.* 2. MIT press.

Breiman, Leo. 1996. "Bagging Predictors." *Machine Learning* 24 (2): 123–140.

Canuto, A. M. de P. 2001. "Combining Neural Networks and Fuzzy Logic for Aplications in Character Recognition." Ph.D. diss., University of Kent.

Chamberlin, M. J. 1974. "The selectivity of transcription." *Annual Rev. Biochemistry* 43:721–725.

Células procarióticas e eucarióticas. `http://www.phschool.com/science/biology_place/biocoach/cells/common.html`(acesso em novembro/2005).

Conceitos básicos de biologia molecular. `http://www.inf.unisinos.br/~mombach/ConceitosBasicosBM.pdf`(acesso em novembro/2005).

Craig, J. M., and W. A. Bickmore. 1994. "The distribution of CpG islands in mamalian chromosomes." *Nature Genetics* 7:376–382.

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

Delemer, B., and G. W. Zhou. 1991. "Neural network optimization for E.coli promoter prediction." *Nucleic Acids Research*, pp. 1593–1599.

Dietterich, T. G. 1998. "Approximate statistical test for comparing supervised classification learning algorithms." *Neural Computation* 10 (7): 1895–1923.

Dietterich, Thomas G. 2000. "An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization." *Machine Learning* 40 (2): 139–157.

Efron, B., and R. Tibshirani. 1993. *An Introduction to the Bootstrap*. Chapman and Hall.

EMBL. 2004. European Bioinformatics Institute. `http://www.ebi.ac.uk/embl/` (acesso: nov/2005).

Fickett, J.W., and A.G. Hatzigeorgiou. 1997. "Eukaryotic promoter recognition." *Genome Research*, pp. 861–878.

Freund, Yoav, and Robert E. Schapire. 1996. "Experiments with a New Boosting Algorithm." *International Conference on Machine Learning*. 148–156.

———. 1998. "Large Margin Classification Using the Perceptron Algorithm." *Computational Learing Theory*. 209–217.

Gordon, Leo, Alexey Ya. Chervonenkis, Alex J. Gammerman, Ilham A. Shahmuradov, and Victor V. Solovyev. 2003. "Sequence alignment kernel for recognition of promoter regions." *bioinformatics* 19 (15): 1964–1971.

Grundy, William N., Timothy L. Bailey, Charles P. Elkan, and Michael E. Baker. 1997. "Meta-MEME: Motif-based Hidden Markov Models of Protein Families." *Computer Applications in the Biosciences* 13 (4): 397–406.

Hannenhalli, S., and S. Levy. 2001. "Promoter prediction in the human genome." *Bioinformatics* 17:90–96.

Harley, C., and R. Reynolds. 1987. "Analysis of E.coli Promoter Sequences." *Nucleic Acids Research* 15:2343–2361.

Hawley, D. K., and W. R. McClure. 1982. "Mechanism of activation of transcription initiation from the lambda PRM promoter." *J. Mol. Biol.* 157 (3): 493–525.

———. 1983. "Compilation and analysis of Escherichia coli promoter DNA sequences." *Nucleic Acids Res.* 11:2237–2255.

Haykin, S. 2001. *Redes Neurais - Princípios e Práticas*. 2. Bookman.

Hearst, M. A., and et al. 1998. "Trends and controversies - support vector machines." *IEEE Intelligent Systems* 13 (4): 1828.

Helmann, J.D. 1995. "Compilation and analysis of Bacillus subtilis a-dependent promoter sequences evidence for extended contact between RNA polymerase and upstream promoter DNA." *Nucleic Acids Res.* 23:2351–2360.

HGMIS, Human Genome Management Information System. 2003, March. "Genomics and its Impact on Science and Society: The Human Genome Project and Beyond."

Huerta, A., and J. Collado-Vides. 2003. "Sigma70 promoters in Escherichia coli: specific transcription in dense regions of overlapping promoter-like signals." *Journal of Mol. Biol.* 333:261–278.

Johnson, Alberts, Lewis Raff, and Roberts Walter. 2004. *Biologia Molecular da Célula*. Editora Artmed.

Joung, K. B., and J. C. Côté. 2002. "Evaluation of ribossomal RNA gene restriction patterns for the classification of Bacillus species and related genera." *Journal of Applied Microbiology* 92 (july): 97–108.

Kanhere, A., and M. Bansal. 2005. "A novel method for prokaryotic promoter prediction based on DNA stability." *BMC Bioinformatics* 6:1–10.

Krogh, A., I.S. Mian, and D. Haussler. 1994. "A Hidden Markov Model that finds genes in E. coli DNA." *Nucleic Acids Research* 22 (22): 4768–4778.

Kuncheva, L. 2004. *Combining pattern classifiers:methods and algorithms*. John Wiley Sons Inc.

Kunst, F. 1997. "The complete genome sequence of the Gram-positive bacterium Bacillus subtilis." *Nature* 390:249–256.

Lectures BMS 209/BMS. `http://www.csu.edu.au/faculty/health/biomed/subjects/molbol/basic.htm`(acesso em novembro/2005).

Lima, Clodoaldo Aparecido de Moraes. 2005. "Comitê de Máquinas: Uma Abordagem Unificada Empregando Máquinas de Vetores-Suporte." Ph.D. diss., Universidade Estadual de Campinhas - Faculdade de Engenharia Elétrica e de Computação.

Lisser, S., and H. Margalit. 1993. "Compilation of E. coli mRNA promoter sequences." *Nucleic Acids Research* 21 (7): 1507–1516.

Matioli, Sérgio Russo. 2001. *Biologia Molecular e Evolução*. Oholos, Editora.

Maza, L. M., M. T. Pezzlo, P.H.A. Sneath, J.T. Staley, and S.T. Williams. 1994. *Bergey´s manual of determinative bacteriology*. 9. Baltimore: Lippincott Williams and Wilkins.

Minsky, M. L., and S. A. Papert. 1969. *Perceptrons*. Cambridge: MIT Press.

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

Monard, M. C., and J. A. Baranauskas. 2003. Chapter Conceitos sobre Aprendizado de Máquina of *Sistemas Inteligentes: Fundamentos e Aplicações*, 89114. Editora Manole.

Monteiro, Meika Iwata, M. C. P. Souto, L. M. G. Gonçalves, and Lucymara F. Agnez-Lima. 2005a. "Machine Learning Techniques for Predicting Bacillus Subtilis Promoters." Edited by J.C. Setubal and S. Verjovski-Almeida, *Proc. of the Brazilian Symposium on Bioinformatics*. Proc. of the Brazilian Symposium on Bioinformatics: Lecture Notes on Computer Science, 77–84.

Monteiro, Meika Iwata, M. C. P. Souto, L. M. G. Gonçalves, and V. Bittencourt. 2005b. "Classificadores e Multi-classificadores para a Predição de Promotores em Bacillus subtilis." Edited by Adrião Duarte Dória Neto, Jorge Dantas de

Melo, A. L. Maitelli, and Allan de Medeiros Martins, *Anais do VII Congresso Brasileiro de Redes Neurais*.

Mulligan, M.E., D.K. Hawley, R. Entriken, and W.R. McClure. 1984. "Escherichia coli promoter sequences predict in vitroRNA polymerase selectivity." *Nucleic Acids Res.* 12:789–800.

Nakata, K., M. Kanehisa, and J. V. Jr. Maizel. 1988. "Discriminant analysis of promoter regions in Escherichia coli sequences." *Comput Appl Bioscience* 4 (3): 367–71.

Nature. 2001. Genome sequence of enterohaemorrhagic Escherichia coli O157:H7. `www.nature.com/genomics/images/e_coli_200.jpg`(acesso em novembro/2005).

NCBI. 2004. National Center of Biotechnology Information. `http://www.ncbi.nlm.nih.gov/` (acesso: nov/2005).

Neves, Adriana, and Ney Lemke. 2005. "An Improved Hidden Markov Model Methodology." *Springer-Verlag Berlin Heidelberg*, pp. 85–94.

Olekhnovich, Igor N., and Robert J. Kadner. 1999. "RNA Polymerase alpha and sigma 70 Subunits Participate in Transcription of the Escherichia coli uhpT Promoter." *Journal of Bacteriology* 181 (23): 7266–7273 (Dezembro).

O'Neil, M. C. 1992. "Escherichia coli promoters: neural networks develop distinct descriptions in earning to search for promoters of different spacing classes." *Nucleic Acids Research*, pp. 3471–3477.

O'Neill, M.C. 1989. "Consensus methods for finding and ranking DNA binding sites. Application to Escherichia coli." *J. Mol. Biol.* 207:301–310.

———. 1991. "Training back-propagation neural networks to define and detect DNA binding sites." *Nucleic Acids Res.* 19:313–318.

O'Neill, M.C., and F. Chiafari. 1989. "Escherichia coli promoters. II. A spacing class-dependent promoter search protocol." *J. Biol. Chem.* 264:5531–5534.

Opitz, D., and R. Mach. 1999. "Popular Ensemble Methods: An Empirical Study." *Journal of Artificial Intelligence Research* 11:169–198.

Oppon, Ekow CruickShank. 2000. "Synergistic use of promoter prediction algorithms:A choice for small trainning dataset?" Ph.D. diss., University of western cape.

Paces, V., C. Vlcek, P. Urbanek, and Z. Hostomsky. 1986. "Nucleotide sequence of the right early region of Bacillus subtilis phage PZA completes the 19366-bp sequence of PZA genome.Comparison with the homologous sequence of phage phi 29." *Gene 44* 1:115–120.

Pedersen, A. G., P. Baldi, S. Brunak, and Y. Chauvin. 1996. "Characterization of Prokaryotic and Eukaryotic Promoters Using Hidden Markov Models." *Source Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology table of contents.* AAAI Press, 182–191.

Pedersen, A. G., and J. Engelbrecht. 1995. "Investigations of Escherichia coli promoter sequences with artificial neural network: New signal discovered upstream of the transcriptional startpoint." *Third International Conference on Intelligent Systems for Molecular Biology.*

Pedersen, Anders Gorm, Pierre Baldin, Yves Chauvin, and Soren Brunak. 1999. "The Biology of Eukaryotic Promoter Prediction - A Review."

Pribnow, D. 1975. Nucleotide sequence of an RNA polymerase binding site at an early T7 promoter. Proc. Natl. Acad. Sci USA.

Reese, M. G. 2001. "Application of a time-delay neural network to promoter annotation in the Drosophila melanogaster genome." *Computers and Chemistry* 26 (1): 51–56 (Dez).

Reese, M. G., N. L. Harris, and F. H. Eeckman, eds. 1996, Jan. *Large scale sequencing specific neural networks for promoter and splice site recognition.* World Scientific Publishing Co.

Reese, Martin G. 2000. "Computational prediction of gene structure and regulation in the genome of Drosophila melanogaster." Ph.D. diss., der Fakultät II - Biologie der Universität Hohenheim.

Reichhardt, T. 2000. "Will souped up salmon sink or swim?" *Nature* 406:10–12.

Rocha, Kliger, César Medeiros, Meika Iwata Monteiro, L. M. G. Gonçalves, and P. MARINHO. 2004. "Design of Specie-Specific Primers for Virus Diagnosis in Plants with PCR." *IEEE Fourth Symposium on Bioinformatics and Bioengineering - BIBE2004*.

Rombauts, S., K. Florquin, M. Lescot, K. Marchal, P. Rouze, and Y. van de Peer. 2003. "Computational approaches to identify promoters and cis-regulatory elements in plant genomes." *Plant Physiol.* 132:1162–1176.

Rorth, P., K. Szabo, and et. al. 1998. "Systematic gain-of-function genetics in Drosophila." *Development* 125:1049–1057.

Rosenblatt, F. 1958. "The perceptrons: A probabalistic model for information and organization in the brain." *Psychological Review* 58:386408.

Ross, W., K.K. Gosink, J. Salomon, K. Igarashi, C. Zou, A. Ishihama, K. Severinov, and R.L. Gourse. 1993. "A third recognition element in bacterial promoters DNA binding by the alpha subunit of RNA polymerase." *Science* 262 (5138): 1407–13.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams. 1986. *Learning internal representations by error propagation*. Edited by PDP Research Group. Parallel Distributed Processing - MIT Press.

Schapire, R. 1990. "The strength of weak learnability." *Machine Learning* 5 (2): 197–227.

Souto, M.C.P., A. C. Lorena, A. C. B Delbem, and A.C.P.L.F. Carvalho. 2003. *Técnicas de Aprendizado de Máquina para problemas de Biologia Molecular*. Instituto de Ciências Matemáticas and de Computação.

Tavares, Douglas Machado, Aquiles Medeiros Burlamaqui, Anfranserai Morais Dias, Meika Iwata Monteiro, L. M. G. Gonçalves, P. J. Alsina, A. A. D Medeiros, and G. L. Filho. 2003. "Hyperpresence - An Application Environment for Control of Multi-User Agents in Mixed Reality Space." *36th Annual Simulation Symposium*.

Tavazoie, S., J. D. Hughes, M. J. Campbell, R.J. Cho, and G. M. Church. 1999. "Systematic determination of genetic network architecture." *Nature Genetics* 22:281–285.

Towell, G., J. Shavlik, and M. Noordewier. 1990. "Refinment of approximate domain theories by knowledge-based neural networks." Edited by AAI Press, *Proc. of the National Conference on Artificial Intelligence*. 861–866.

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

Weka. 2004. Weka 3: Data Mining Software in Java. `http://www.cs.waikato.ac.nz/ml/weka/` (acesso: abril/2005).

Werner, T. 2003. "The state of the art of mammalian promoter recognition." *Brief. Bioinform.* 4:22–30.

Witten, I., and E. Frank. 2000. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann publishers.

Yamada, M., M. Kubo, T. Miyake, R. Sakaguchi, Y. Higo, and T. Imanaka. 1991. "Promoter sequence analysis in Bacillus and Escherichia: construction of strong promoters in E.coli." *Gene* 1 (99): 109–114 (Mar).

Zaha, Arnaldo. 1996. *Biologia Molecular Básica*. Porto Alegre.

.1 Construção da base de dados híbrido

Da mesma forma que a base de dados de *B.subtilis*, a construção da base de dados híbrido foi comparada com a base de dados da *E.coli*. Para isso, analisamos 51 seqüências, vistas na tabela 1, encontradas no do Centro Nacional de Informação em Biotecnologia - NCBI (*National Center for Biotechnology Information* - http://www-.ncbi.nlm.nih.gov/).

Espécie de bacilo	Promotores encontrados	Promotores utilizados
Liqueniformis	11	07
Amyloliquefaciens	04	04
Cereus	02	01
Megaterium	16	05
Thurigiensis	14	11
Firmus	02	02
TOTAL	51	30

Tabela 1: Promotores híbridos utilizados na base de dados

Apenas 30 promotores foram aproveitados dos 51 encontrados. Da mesma forma que a outra base de dados, foram descartadas todas as seqüências hipotéticas, repetidas e pequenas. Infelizmente, algumas seqüências encontradas nos banco de dados possuem anotações com vários erros. Foram encontrados erros de anotação da chave primária (ig - número de identificação do gene), erros de escritas de definições, etc. Isso dificultou nossa análise e tivemos que reavaliar todas as seqüências, para comprovar, por exemplo, se também haveria erro de anotação do próprio promotor. Para isso foi utilizado o software de alinhamento global denominado BLAST (sigla em inglês *Basic Local Alignment Search Tool*) disponível no site do NCBI. Este software faz o alinhamento de todas as seqüências existentes no banco e organiza-as de acordo com a maior taxa de semelhança, denominada *(E-value)*.

.1.1 Escolha dos promotores híbridos

Tentamos inicialmente aproveitar o máximo de promotores possíveis, pelo fato de existirem poucos exemplos dentro de tantos outros que foram descartados. Fizemos

uma base de dados inicial delimitando-a pelo menor tamanho possível. Não conseguimos bons resultados por possuírem seqüências as poucas informações necessárias a um treinamento aceitável.

Refizemos os cortes dos nucleotídeos para tamanhos maiores. O tamanho final da janela fixa foi obtido a partir da média do tamanho de todas as seqüências de promotores. Seqüências menores que 111 também foram descartadas. O corte (descarte de nucleotídeos) de seqüências maiores que 111 foi realizado no final da seqüência (*upstream*). Esta estratégia de corte foi escolhida para preservar a região promotora dos Bacilos que se encontram, aproximadamente, entre -80 (*downstream*) e +15 (*upstream*) do gene de cada promotor.

.1.2 Escolha dos não-promotores híbridos

Selecionar um genoma completo de um organismo que não tenha características de diversos bacilos ao mesmo tempo é uma tarefa complicada. Uma das opções dadas pelos biólogos foi a de selecionarmos, no genoma dessas espécies, fragmentos que não teriam nenhuma relação com um promotor, como por exemplo, um segmento no meio do gene. Mas isso estaria pondo em risco a confiabilidade dos dados.

Devido à falta de literatura confiável, ou seja, publicações em revistas conhecidas, optamos portanto, por utilizarmos também do mesmo genoma de um bacteriófago de Bacilo subtilis PZA, que foi utilizado na construção dos não promotores dos *B.subtilis*. O *B.subtilis* possui mais de 80% de similaridade entre todos os Bacilos presentes na base de dados híbridos.

Da mesma forma que a base de dados citada anteriormente, a similaridade foi calculada através do alinhamento de todas as seqüências promotoras com o genoma inteiro do bacteriófago, pelo programa online ClustalW.

.2 Resultados para base de dados híbrido

Como já dito no capítulo anterior, a base de dados híbrido foi construída utilizando seis espécies de bacilos com alta similaridade evolutiva. Analisá-los, primeiramente,

com os classificadores de base.

.2.1 Classificadores Base

Os melhores resultados obtidos utilizando os classificadores base podem ser observados na tabela 2 pela sua média e seu desvio-padrão. Todos os parâmetros foram selecionados igualmente à base de dados mostrados anteriormente.

Algoritmo	Média	Desvio-Padrão
k-NN	38,33%	48,66%
NB	35%	47,74%
AD	35%	47,74%
SVM	34,17%	46,81%

Tabela 2: Média de taxas de erros dos classificadores base para a base de dados híbrido

Ao analisar todos os métodos através do teste de hipótese, foi descoberto que não há evidência de diferença estatisticamente significativa entre os resultados de nenhum desses métodos.

Livros Grátis

(http://www.livrosgratis.com.br)

Milhares de Livros para Download:

Baixar livros de Literatura
Baixar livros de Literatura de Cordel
Baixar livros de Literatura Infantil
Baixar livros de Matemática
Baixar livros de Medicina
Baixar livros de Medicina Veterinária
Baixar livros de Meio Ambiente
Baixar livros de Meteorologia
Baixar Monografias e TCC
Baixar livros Multidisciplinar
Baixar livros de Música
Baixar livros de Psicologia
Baixar livros de Química
Baixar livros de Saúde Coletiva
Baixar livros de Serviço Social
Baixar livros de Sociologia
Baixar livros de Teologia
Baixar livros de Trabalho
Baixar livros de Turismo