Classificadores e Multi-classificadores para a Predição de Promotores em *Bacillus subtilis*

Meika I. Monteiro*, Marcílio C. P. de Souto† ,Luiz Marcos Garcia Gonçalves*, Valnaide G. Bittencourt*

*Departamento de Computação e Automação**
Departamento de Informática e Matemática Aplicada†
{meika, lmarcos, valnaide}@dca.ufrn.br, marcilio@dimap.ufrn.br

Resumo

Um dos principais objetivos da bioinformática é a identificação de genes em seqüências descaracterizadas de DNA. A melhoria de técnicas de predição de promotores pode ser um avanço significativo para o desenvolvimento de métodos mais eficientes de predição de genes (ab initio). Neste trabalho, apresentamos uma comparação empírica de técnicas individuais de aprendizado de máquina (Naive Bayes, 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) à tarefa de predição de promotores em Bacillus subtilis. Para isso, foi construída uma base de dados com seqüências de promotores e não-promotores desse organismo.

1. Introdução

O processo de mapeamento de uma seqüência de DNA para a formação de proteínas em organismos eucarióticos e procarióticos envolve algumas etapas [1]. O primeiro passo é a transcrição de um fragmento do DNA em uma molécula de RNA, chamado RNA mensageiro. Esse processo se inicia com a ligação de uma molécula, chamada de RNA polimerase a uma certa posição na molécula do DNA.

O local exato onde o RNA polimerase se liga determina qual a parte do DNA que será lido e onde será feita a transcrição. Existem locais, anteriores a regiões codificadoras de proteínas, que contêm sinais que são reconhecidos pela RNA polimerase: essas regiões são chamadas de promotores [1].

Métodos computacionais de predição de promotores podem ser um avanço para o desenvolvimento de métodos mais eficientes de predição de genes ab initio [2], [3]. Esses métodos também podem ser vistos como uma parte de um processo complexo na descoberta de atividade de genes em redes regulatórias [4]. Tal tarefa de predição é difícil, por causa da variabilidade que os sinais apresentarem entre suas distâncias, além disso há outros fatores que estão envolvidos na regulação do nível de expressão do gene [1], [5].

Apesar dessas limitações, um grande número de programas de predição de promotores tem sido desenvolvido para organismos eucariotos [5], [6], [7],[8]. Entretanto, até agora, são poucos os sistemas que podem ser usados como ferramenta para a predição de promotores em organismos procarióticos, como o programa *Neural Network Promoter Prediction* (NNPP). Alguns outros métodos de predição de promotores de procariotos são baseados em busca de padrões em matrizes com pesos [10] [11] e [12]. Um método novo de predição de promotores em organismos procariotos é apresentado baseado na estabilidade do DNA em [13].

Há uma diferença conceitual, segundo [5], entre tentar reconhecer promotores e reconhecer somente aqueles que serão ativados em seu tipo específico de célula. Alternativamente, o problema pode ser dividido em vários subproblemas. Isto significa que seria necessário construir algoritmos específicos para reconhecer classes específicas de promotores. A maioria dos sistemas referenciados, por exemplo, no parágrafo anterior, foram projetados para propósitos específicos, tal como para a bactéria da *Escherichia coli (E.coli)*.

Por esta razão, aplicamos neste artigo técnicas de Aprendizado de Máquina (AM) para análise de promotores em *Bacillus subtilis (B. subtilis)*. Escolhemos este organismo procarioto por ser amplamente utilizado como modelo em estudos genéticos. Além disso, há, na literatura, muitas análises experimentais comprovando várias de suas seqüências de promotores [14], o que ajudou na seleção e construção da nossa base de dados.

2. Trabalhos relacionados

No contexto de AM, a identificação de seqüências promotoras pode ser colocada da seguinte forma:

- 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.

Um trabalho que mais se aproxima ao nosso pode ser encontrado em [16]. Neste trabalho um enfoque híbrido baseado em redes neurais e regras simbólicas é aplicado para predizer seqüências promotoras de *E.coli.* O sistema usado, chamado KBANN *(Knowledge Based Neural Network)* usa regra proposicionais formuladas por um biólogo, a qual determina uma topologia neural e pesos iniciais. Fazendo isso, pode-se observar a diminuição do tempo de treinamento da rede e uma melhora em sua generalização [16]. Os dados usados por [16] contêm 53 exemplos de promotores e 53 de não-promotores, onde cada exemplo possuía 57 atributos (por exemplo, 57 nucleotídeos A, T, C ou G). Nos experimentos feitos pelo autor, os resultados obtidos com o KBANN foram comparados com RN do tipo *multi-layer perceptron*, AD (árvores de decisão), *k*-NN (*k*-vizinhos mais próximos), dentre outros. Os melhores resultados obtidos foram da RN e do KBANN, enquanto que a AD e o *k*-NN obtiveram resultados insatisfatórios. O autor não apresenta os resultados das taxas de falso-positivos.

Em [9], um modelo de rede neural de propriedades estruturais e composicionais de uma região promotora eucariótica, o *Neural Network Promoter Prediction* (NNPP), foi desenvolvido e aplicado para análise da *Drosophila melanogaster*. O modelo usa uma arquitetura de *time-delay* que é um caso especial de rede neural *feedforward*. De acordo com os autores, a aplicação desses modelos para testar potenciais sítios promotores obteve um melhor resultado em comparação a outros métodos estatísticos e neurais, como também revelou, indiretamente, propriedades sutis do sinal no início da transcrição. Tal modelo foi estendido para promotores procariotos. Na verdade, é uma das poucas ferramentas disponíveis atualmente para predição de promotores em organismos procariotos(www.fruitfly.org/*seq_tools/promoter.html).*

Recentemente, [21] desenvolveram uma análise comparativa na aplicação de SVM e *transductive support vector machines* (TSVM) na predição de regiões promotoras de eucariotos. De acordo com eles, o segundo método mencionado obteve melhor desempenho.

3. Materiais e métodos

Realizamos uma comparação empírica entre sistemas baseados em regras (com Árvore de Decisão e PART), sistemas de aprendizado estatístico (como *Naive Bayes* – NB, K-Vizinhos Mais Próximos – *k*-NN, Máquinas de Vetores Suporte – SVM e Redes Neurais do tipo VotedPerceptron – VP) e sistemas de multiclassificação (*Bagging* e *Adaboosting*), aplicados à tarefa de predição de promotores em *B.subtilis)*. Todos os métodos usados em nosso estudo foram obtidos do pacote de aprendizagem da máquina do WEKA [15] (www.cs.waikato.ac.nz/~ml/weka/) e tiveram seus resultados comparados entre si.

3.1. Base de dados

Construímos uma base de dados utilizando os mesmos parâmetros usados na elaboração da base de dados da *E.coli,* usada originalmente em [16]. Esta base de dados está disponível em (ftp://ftp.ics.uci.edu/pub/machine–learningdatabases /molecular–biology/promoter–gene–sequences/).
Ela contém 53 exemplos de promotores e 53 exemplos de não-promotores obtidos de uma compilação produzida em [17]. De acordo com [16], exemplos 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 de 57 atributos.

No caso da base de dados do *B.subtilis,* consideramos, inicialmente, apenas os dados experimentalmente comprovados, apresentados na compilação produzidos por [14]. Calculamos a média de comprimento dessas seqüências, que foi de 117 nucleotídeos. Fixamos um comprimento de 117 nucleotídeos para as nossas seqüências de promotores. Seqüências menores que esse número, e ditas como 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 cortadas em suas regiões de *upstream*, de modo ao seu comprimento final ser de 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 em [14].

Para criar os não-promotores para nossa base de dados, selecionamos 112 seqüências contínuas de 117 nucleotídeos de um genoma de um bacteriófago PZA do *B.subtilis* [18], onde não existe seqüência promotora identificada. Estas seqüências foram escolhidas de tal forma que se obtivesse o maior grau de similaridade possível de cada uma delas com as seqüências dos promotores. A média de similaridade entre os promotoras e não-promotoras foi de 27%. No caso da base de dados de *E. coli* [16], essa média é de 24%.

Resumindo, obtivemos uma base de dados de 112 promotores e 112 não-promotores, cada uma possuindo 117 nucleotídeos.

3.2. Avaliação

A comparação de dois métodos de aprendizagem supervisionada é realizada analisando o nível de significância estatística da diferença entre a média da taxa do erro de classificação, em conjuntos independentes de teste, dos métodos avaliados. A fim de avaliar a média da taxa do erro de classificação, 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 de *k-fold cross validation* [19], [20], [15].

Neste trabalho, para utilizarmos a maior quantidade de dados possível no treinamento, utilizamos o método *leave-one-out*, que é uma forma especial de *k-fold cross validation,* onde *k* é o número de instâncias do conjunto de dados. Em nosso caso, *k* é 224, onde *k* é a quantidade total de seqüências promotoras e não-promotoras da base de dados.

4. Experimentos

Nossos experimentos foram realizados apresentando a base de dado construída aos algoritmos de aprendizado de máquina. Devido à metodologia *leave-one-out*, cada método foi executado 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. Então, 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 executado. Caso contrário, o mesmo processo é repetido para o valor do parâmetro com o melhor resultado até então, e assim por diante. Naturalmente, este procedimento 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 [15]):

- *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;
- *Bagging para PART, k-nn, AD, NB, VP e SVM*: todos os parâmetros dos classificadores de bases foram configurados 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 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 (como descrito em [19] e [20]).

5. Resultados

Antes de começarmos nossa investigação da performance dos sistemas multiclassificadores, analisamos a performance dos algoritmos individuais (classificadores base) aplicados ao problema, de acordo com a média da taxa de erro de classificação total.

5.1. Classificadores base (individuais)

A tabela 1 apresenta, para cada algoritmo de aprendizado de máquina, a média e o desvio de padrão da porcentagem incorreta dos exemplos classificados em conjuntos de testes independentes.

De acordo com esta tabela, NB e SVM obtiveram a menor taxa de erro de classificação (18,30%). As hipótese nulas foram rejeitadas em favor do SVM e NB em comparação com *k*-NN, AD e PART com α=0,05, onde α=0,05 é o coeficiente do nível de significância para o teste de hipótese.

Tabela 1. Médias das taxas de erros

Algoritmo	Média	DP
k-NN	34,82%	47,75%
NB	18,3%	38,76%
AD	30,8%	46,27%
PART	31,25%	46,46%
VP	32,14%	46,81%
SVM	18,30%	38,76%

Considerando os melhores resultados obtidos em nossos experimentos, em termos da menor taxa de erro, escolhemos NB e SVM para uma comparação mais detalhada, como mostrado nas Tabelas 2 e 3. Essas tabelas mostram a média de predição e classificação atual.

Tabela 2. Matriz de confusão de NB

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

Tabela 3. Matriz de confusão para SVM

Verdadeiro/Predito	Promotor	Não-promotor
Promotor	72%	24%
Não-Promotor	12,4%	87,5%

Em relação às tabelas anteriores (2 e 3), para uma comparação mais conveniente dos resultados, o nosso esquema de avaliação irá considerar a taxa de verdadeiros positivos, TP (segunda coluna da segunda linha) e a taxa de falso positivo, FP (segunda coluna da terceira linha). Neste caso, NB e SVM apresentam, respectivamente, a seguinte relação VP/FP: 4,31, 6,08 e 2,67. Portando, neste contexto, SVM oferece um melhor compromisso entre generalização e discriminação. Isto também implica em um melhor controle dos falsos positivos. Esta questão é importante devido a existir, como já mencionado, uma alta probabilidade de se achar seqüências que não são promotores similares a promotores.

Também testamos nossa base de dados utilizando o NNPP. O programa NNPP está disponível no *site* (www.fruitfly.org/seq_tools/promoter.html). Todas as predições do NNPP foram executadas com um corte de 0.80. No caso da nossa base de dados, o NNPP predisse corretamente 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 sendo promotoras). Ou seja, para esta base de dados, o método NNPP apresentou uma relação VP/FP de 1,29, o que é bem menor que a taxa apresentada pelo nosso SVM (6,08).

5.2. Multiclassificadores

A principal idéia em usar técnicas de construção de multi-classificadores é de que a combinação de classificadores individuais (base) pode levar a uma melhora na precisão e a estabilidade dos resultados dos algoritmos de classificação. Esta idéia conduz as 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. Esta técnica é mais bem aplicada para classificadores instáveis, tais como Árvores de Decisão e Redes Neurais [22]. Com relação a segunda questão, a maneira mais simples de se fazer isto no caso de classificação é por meio de uma votação (ponderada ou não).

Bagging e *AdaBoosting*, as técnicas de geração de multi-classificadores usadas neste trabalho, usam estas estratégia. Tais técnicas se diferenciam, basicamente, na forma em que geram os classificadores base. *Bagging* (*Bootstrap Aggregating*) constrói os classificadores a partir de conjuntos sucessivos e independentes de amostras de dados. Estas amostras são geradas a partir de um conjunto de dados de treinamento com *m* itens. Aleatoriamente, são extraídos *m* exemplos com substituição a partir do conjunto original, ou seja, ocorrendo portanto a repetição de exemplos na amostra. Desta maneira, exemplos utilizados em uma amostra aparecem novamente em outra. Isto colabora para que os classificadores sejam diferentes, devido a esta variação de exemplos nas amostras. A idéia geral é que a combinação destes classificadores através da votação da classe predita com maior freqüência, leve a uma melhor acurácia.

O *Adaboosting* (*Adaptive Boosting*) faz parte da classe de algoritmos que alteram a distribuição do conjunto de treinamento, baseando-se na performance das classificações anteriores. Isto deve-se ao fato da característica básica do seu funcionamento, onde os classificadores são gerados seqüencialmente. A cada passagem os pesos dos

exemplos são alterados em função do sucesso de sua classificação (os exemplos que não são classificados tem seu peso aumentado). Por fim, após *n* passagens que são previamente definidas, é gerado um classificador final formado por um esquema de votação, sendo que o peso de cada classificador depende da sua performance no conjunto de treinamento em que ele foi construído.

As Tabelas 5 e 6 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 todos os métodos individuais previamente apresentados na Tabela 1, com exceção do KNN e NB que nos experimentos preliminares não se mostraram adequados à tarefa de predição de promotores.

Tabela 5. Taxa de erro da técnica *Bagging*

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 6. Taxa de erro da técnica *Boosting*

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

Em relação aos resultados obtidos com a técnica *Bagging* (Tabela 5), não é verificada nenhuma diferença estatisticamente significativa entre eles (α=0,05). Em relação aos resultados da técnica *Adaboosting* (Tabela 6), também não há evidência de diferença estatisticamente significativa entre os resultados desses métodos, apesar da menor taxa de classificação incorreta (18,30%). Devido ao elevado desvio padrão apresentado por todos os métodos.

6. Considerações finais

Neste trabalho, apresentamos uma comparação empírica de técnicas individuais de aprendizado de máquina (*Naive Bayes*, Árvores de Decisão, Máquinas de Vetores de Suporte, Redes Neurais do tipo *VotedPerceptron*, PART e K-Vizinhos Mais Próximos) e sistemas multiclassificadores (*Bagging*, *Adaboosting*) à tarefa de predição de promotores em *B. subtilis*. Para isto, como uma de nossas contribuições, construímos inicialmente uma base de dados de promotores e não-promotores para este organismo.

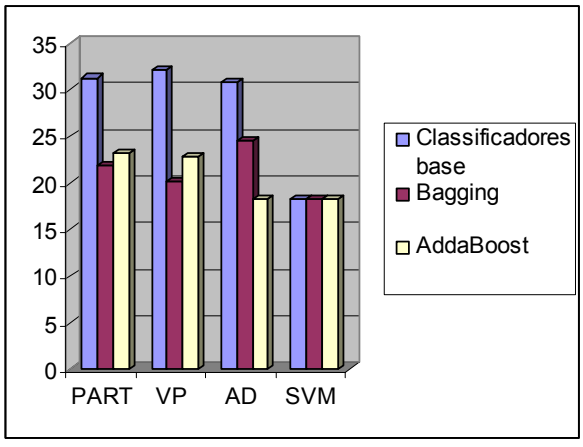
Dos diferentes classificadores base analisados, o SVM obteve o melhor resultado em comparação aos outros em termos da relação TP/FP (6,08). De fato, este método obteve melhor performance quando comparado ao NNPP. Uma das razões para que isto tenha ocorrido está no fato de termos construído um classificador específico para a bactéria *B. subtilis,* enquanto que o NNPP é um classificador geral.

A tabela 7 resume os melhores resultados obtidos com os métodos PART, VP, AD e SVM além de seus respectivos sistemas de multiclassificação com as técnicas *Bagging* e *Adaboosting* empregadas.

Observando a tabela 9 pode-se verificar uma considerável melhoria nos resultados obtidos com métodos de multiclassificação que usaram as ADs como o classificador de base. Observa-se que o resultado original obtido com a AD, como o 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 aquele conseguido com o *Bagging*.

Em relação ao PART e o VP, os resultados obtidos com as técnicas de multiclassificação não mostraram a mesma melhoria que na AD. Por o exemplo, nenhuma diferença estatística foi detectada entre o Bagging PART e o PART Adaboosting. No caso do *Adaboosting VP e o Bagging* VP, o multiclassificador gerado mostraram um desempenho estatisticamente superior quando comparado ao VP individual. Porém não houve hipótese nula entre Adaboosting VP e o Bagging VP. Para o PART, nenhuma diferença estatística foi detectada em relação ao Bagging PART e AdaBoosting PART. Somente o Bagging PART validou estatiticamente em relação ao PART individual. O SVM permaneceu com os mesmos resultados somente variando seu desvio-padrão.

Tabela 9. Resultados gerais



7. Referências

[1] B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, and J. Watson, *The molecular biology of the cell*, second edition ed. New York: Garland Publishing, 1989.

[2] P. Baldi and S. Brunak, *Bioinformatics: the Machine Learning Approach*, second edition ed. MIT Press, 1998.

[3] M. W. Craven and J. Shavlik, "Machine learning approaches to gene recognition," *IEEE Expert*, vol. 9, pp. 2–10, 1994.

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

[5] A. G. Pedersen, P. Baldi, Y. Chauvin, and S. Brunak, "The biology of eukaryotic promoter prediction - a review." *Comput. Chem.*, vol. 23, pp. 191–207, 1999.

[6] J. W. Fickett and A. G. Hatzigeorgiou, "Eukaryotic promoter recognition," *Genome Res.*, vol. 7, pp. 861–78, 1997.

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

[8] T. Werner, "The state of the art of mammalian promoter recognition," *Brief. Bioinform.*, vol. 4, pp. 22–30, 2003.

[9] M. G. Reese, "Application of a time-delay neural network to promoter annotation in the drosophila melanogaster genome," *Comput. Chem.*, vol. 26, no. 1, pp. 51–56, 2001.

[10] R. Staden, "Computer methods to locate signals in nucleic acid sequences," *Nucleic Acids Res.*, vol. 12, pp. 505–519, 1984.

[11] M. Mulligan, D. K. Hawley, R. Entriken, and W. McClure, "Escherichia coli promoter sequences predict in vitro RNA polymerase selectivity," *Nucleic Acids Res.*, vol. 12, pp. 789–800, 1984.

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

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

[14] J. D. Helmann, "Compilation and analysis of Bacillus subtilis of extended contact between RNA polymerase and upstream promoter DNA," *Nucleic Acids Research*, vol. 23, no. 13, pp. 2351–2360, 1995.

[15] I. H. Witten and E. Frank, *Data mining: practical machine learning tools and techniques with Java implementation*. USA: Morgan Kaufman Publishers, 2000.

[16] G. G. Towell, "Symbolic knowledge and neural networks: insertion, refinement and extraction," PhD thesis Computer Science, University of Wisconsin, 1991.

[17] C. B. Harley and R. P. Reynolds, "Analysis of E. coli promoter sequences," *Nucleic Acids Research*, vol. 15, pp. 2343–2360, 1987.

[18] V. Paces, C. Vlcek, P. Urbanek, and Z. Hostomsky, "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*, vol. 44, no. 1, pp. 115–120, 1986.

[19] T. Mitchell, *Machine Learning*. New York: McGraw Hill, 1997.

[20] T. G. Dietterich, "Approximate statistical test for comparing supervised classification learning algorithms," *Neural Computation*, vol. 10, no. 7, pp. 1895–1923, 1998.

[21] N. Kasabov and S. Pang, "Transductive support vector machines and applications in bioinformatics for promoter recognition," *Neural Information Processing - Letters and Reviews*, vol. 3, no. 2, pp. 31–37, 2004.

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