ESTUDO COMPARATIVO DE MÉTODOS DE APRENDIZADO DE MÁQUINA NA DETECÇÃO DE REGIÕES PROMOTORAS DE GENES DE *ESCHERICHIA COLI*

LEONARDO G. TAVARES, HEITOR S. LOPES, CARLOS R. ERIG LIMA

Laboratório de Bioinformática
Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial
Universidade Tecnológica Federal do Paraná – UTFPR
Av. 7 de Setembro, 3165- 80230-901 Curitiba/PR
E-mails: leonardo.tavares@unicenp.edu.br, hslopes@pesquisador.cnpq.br, erig@utfpr.edu.br

Abstract— Tools based on Machine Learning methods have been widely used in the bioinformatics, mainly in the construction of classifiers. Detection of genes is still an open problem. Identifying the promoter region laying prior the gene itself is an important aid to detect a gene. This paper aims at comparing several Machine Learning methods applied to the construction of classifiers for detection of promoters of *Escherichia coli* DNA.

Keywords— Bioinformatics, DNA, Machine Learning.

Resumo— Ferramentas baseadas nos métodos de Aprendizado de Máquina têm sido amplamente utilizadas na Bioinformática, principalmente na construção de classificadores. A detecção de genes ainda é um problema em aberto. Identificar a região promotora que antecede o gene propriamente dito é uma importante metodologia para detectar um gene. O objetivo deste artigo é comparar diversos métodos de Aprendizado de Máquina aplicados à construção de classificadores para a detecção de regiões promotoras em DNA da bactéria *Escherichia coli*.

Palavras-chave— Bioinformática, DNA, Aprendizado de Máquina.

1 Introdução

O DNA é um polímero linear constituído de quatro unidades básicas chamadas de nucleotídeos ou bases. São elas: Adenina (A), Citosina (C), Guanina (G) e Timina (T). Acredita-se hoje que o DNA possua embutida na sua seqüência toda informação necessária para o desenvolvimento e funcionamento dos organismos. Um dos esforços da comunidade científica atual é a de conceber ferramentas que possam reconhecer as regiões na seqüência do DNA onde estas informações essenciais estão, os chamados genes. Os genes, por sua vez, são transcritos em proteínas que vão atuar nas células nas mais variadas atividades.

O início da transcrição do gene se dá quando a enzima RNA polimerase se liga a uma região do DNA, conhecida como região promotora. Esta região, simplificadamente, indica à RNA polimerase que a informação gênica está por vir e que a transcrição do DNA deve ser iniciada. Desta forma, uma seção do DNA é transcrita em RNA mensageiro, o qual, após a extração dos introns, levará ao mapeamento de uma proteína (Lehninger et al. 2006). As regras exatas através das quais a RNA polimerase atua não são totalmente conhecidas e, portanto, se faz necessário o estudo de seu comportamento baseando-se em exemplos já conhecidos. A partir desta metodologia e, apoiando-se em métodos de Inteligência Computacional, tenta-se conhecer melhor o funcionamento deste processo bioquímico essencial à vida.

A identificação dos genes presentes na seqüência do DNA e a determinação de suas funções é a tarefa seguinte ao seqüenciamento do DNA de um organismo. De fato, a detecção de genes é, ainda, um problema em aberto e muitas metodologias para este propósito foram propostas. A detecção de regiões promotoras no DNA é um dos recursos mais importantes para a detecção de genes (Mount, 2001).

Atualmente as técnicas mais utilizadas para o reconhecimento de regiões promotoras em DNA são baseadas em métodos de Aprendizado de Máquina tais como redes neurais artificiais, árvores de decisão, modelos ocultos de Markov e outros.

A área de Aprendizado de Máquina é um ramo de estudo da Inteligência Artificial, sendo relacionada ao projeto e desenvolvimento de algoritmos e técnicas capazes de permitirem o aprendizado computacional. De maneira genérica, há dois tipos de aprendizado: indutivo e dedutivo. Os métodos pertencentes ao primeiro tipo são mais populares e extraem regras ou padrões (ou, ainda, conhecimento formatado similarmente) a partir de dados. O foco de Aprendizado de Máquina está na extração de conhecimento de grandes massas de dados de maneira automática, utilizando, para tanto, métodos computacionais e estatísticos. O espectro de aplicações das técnicas de Aprendizado de Máquina é bastante extenso, incluindo, por exemplo, processamento de linguagem natural, reconhecimento de padrões, diagnóstico, detecção de fraudes, reconhecimento de escrita e de fala, análise de investimentos, visão computacional, robótica e bioinformática.

O objetivo do presente trabalho é avaliar comparativamente diversos métodos de Aprendizado de Máquina para o problema da detecção de regiões promotoras em DNA da bactéria *Escherichia coli*.

Uma descrição da metodologia é apresentada na seção 2, onde são abordados aspectos básicos sobre os métodos computacionais de aprendizado de máquina utilizados. Na seção 3 são apresentados os diversos experimentos realizados e os resultados obtidos neste trabalho. Finalmente, na seção 4 são apresentadas as conclusões do trabalho e propostas para trabalhos futuros.

2 Metodologia

2.1 Base de dados

Este trabalho utiliza uma base de dados compilada por Harley e Reynolds (1987) da Universidade de Wisconsin. O conjunto de dados possui 106 instâncias, das quais 53 contêm regiões promotoras e 53 são trechos de regiões intragênicas e, portanto, não possuem as características das regiões promotoras. Cada instância possui 57 nucleotídeos, sendo que aquelas que correspondem a regiões promotoras se referem às posições -50 até a +7 do início de um gene. Isto é, os 50 nucleotídeos que precedem um gene e os primeiros 7 nucleotídeos de um gene.

Neste trabalho foram removidos os nucleotídeos das posições +1 a +7, sendo apenas consideradas as posições -50 a -1 (a região que precede efetivamente a informação gênica).

2.2 Métodos computacionais

Foram utilizadas duas ferramentas para a avaliação dos métodos de Aprendizado de Máquina: um conhecido software frequentemente empregado em mineração de dados denominado Weka (Witten e Frank, 2005), e o HMMER versão 2.3.2 (Durbin et al. 1998), que é uma ferramenta muito utilizada na área da Bioinformática.

O software Weka foi utilizado por possuir uma grande quantidade algoritmos de classificação de dados reunidos em um único pacote. Para avaliar cada um dos métodos disponíveis no Weka é necessário um arquivo de dados, cujo formato é descrito na documentação do mesmo.

O HMMER foi utilizado para avaliar o método conhecido como Modelo Oculto de Markov (HMM - *Hidden Markov Model*). Muitos trabalhos têm sido publicados nesta linha (Weinert e Lopes, 2004), pois o HMM se mostra bastante eficiente na detecção de padrões em seqüências, pois consegue tratar de forma probabilística a variação estrutural dos elementos. Um Modelo Oculto de Markov é um método estatístico considerado uma rede Bayesiana simples. Num HMM o sistema sendo modelado é assumido como sendo um processo Markoviano cujos parâmetros são desconhecidos. O objetivo é estimar o valor de tais parâmetros a partir de instâncias conhecidas. O modelo assim obtido pode ser então utilizado para análises de outros dados ou para o reconhecimento de padrões (Ephraim e Merhav, 2002).

2.2.1 Weka

Para conversão do arquivo original para o formato ARFF, nativo do Weka, cada posição correspondente a um nucleotideo foi nomeada como m50, m49, m48 e assim sucessivamente até a m1 como uma abreviação à "posição -50", por exemplo. Estas colunas se referem aos 50 atributos previsores. O atributo-meta identifica se a amostra contém ou não uma região promotora. A figura 1 mostra um trecho do arquivo gerado.

```
...
@attribute m8 {A,C,G,T}
@attribute m7 {A,C,G,T}
@attribute m6 {A,C,G,T}
@attribute m5 {A,C,G,T}
@attribute m4 {A,C,G,T}
@attribute m3 {A,C,G,T}
@attribute m2 {A,C,G,T}
@attribute m1 {A,C,G,T}
@attribute PROMOTER {Y,N}

@data
T,A,C,T,A,G,C,A,A,T,A,C,G,C,T,T,G,C,G,T,T,C,
T,G,C,T,A,T,C,C,T,G,A,C,A,G,T,T,G,T,C,A,C,G,
G,T,A,C,T,A,G,A,G,A,A,C,T,A,G,T,G,C,A,T,T,A,
A,A,T,T,G,T,G,A,T,G,T,G,T,A,T,C,G,A,A,G,T,G,
T,C,G,A,T,A,A,T,T,A,A,C,T,A,T,T,G,A,C,G,A,A,
A,G,G,G,G,C,A,A,G,G,A,G,G,A,T,G,G,A,A,A,G,A,
C,A,G,G,G,G,G,T,G,G,A,G,G,A,T,T,T,A,A,G,C,C,
T,T,T,C,T,A,C,A,A,A,A,C,A,C,T,T,G,A,T,A,C,T,
C,G,A,C,T,T,A,A,T,A,T,A,C,T,G,C,G,A,C,A,G,G,
...
```

Figura 1: Formato do arquivo ARFF para o Weka.

2.2.2 HMMER

Para construção de um modelo HMM é necessário fornecer um arquivo no formato MSF com seqüências alinhadas das informações de DNA ou de proteínas. Neste trabalho as seqüências são os 53 exemplos positivos (com regiões promotoras) da base de dados. Não houve a necessidade de fazer um alinhamento das seqüências (Durbin et al. 1998), pois estas já estavam alinhadas. Um trecho do arquivo MSF gerado é exemplificado na figura 2.

```
Ex1             TACTAGCAAT ACGCTTGCGT TCGGT
Ex2             TGCTATCCTG ACAGTTGTCA CGCTG
Ex3             GTACTAGAGA ACTAGTGCAT TAGCT
Ex4             TCGATAATTA ACTATTGACG AAAAG
Ex5             AGGGGCAAGG AGGATGGAAA GAGGT
Ex6             CAGGGGGTGG AGGATTTAAG CCATC
Ex7             TTTCTACAAA ACACTTGATA CTGTA
Ex8             CGACTTAATA TACTGCGACA GGACG
Ex9             TTTTAAATTT CCTCTTGTCA GGCCG
Ex10            GCAAAAATAA ATGCTTGACT CTGTA
```

Figura 2: Formato do arquivo MSF para o HMMER.

3 Experimentos e Resultados

Utilizando o software Weka, foram testados todos os métodos de classificação supervisionados disponíveis e que suportam os tipos de dados avaliados. Foram avaliados, no total, 30 métodos de classificação, agrupados em métodos Bayesianos, redes neurais, árvores e outros.

Foi adotada a metodologia de validação cruzada de 10 partições (Kohavi, 1995), já prevista no próprio Weka. Validação cruzada é um procedimento estatístico bastante comum no âmbito de mineração de dados e Aprendizado de Máquina e consiste na partição de uma amostra de dados em subconjuntos. A análise é inicialmente realizada em um subconjunto enquanto os demais são guardados para uma subseqüente confirmação e validação da análise inicial. O conjunto de dados inicial é chamado de conjunto de treinamento e os outros, conjuntos de validação ou teste. A validação cruzada de 10 partições significa que uma parte é utilizada para treinamento e as demais partes para validação. O procedimento é repetido circularmente para todas as partições, onde cada uma é utilizada uma única vez para o treinamento. O resultado final é a média das avaliações. A validação cruzada é utilizada basicamente para evitar resultados tendenciosos ao se utilizar um espaço amostral reduzido.

O software Weka gera uma matriz de confusão além de outras informações importantes, conforme mostrado simplificadamente na figura 3 para o algoritmo ID3.

Para avaliação do HMM também foram consideradas somente as posições de -50 a -1 e a metodologia adotada também foi a validação cruzada de 10 partições. Isto foi feito para que os resultados pudessem ser comparados equitativamente com aqueles obtidos pelos classificadores do Weka.

Cada modelo oculto de Markov foi construído utilizando o aplicativo *hmmbuild* e posteriormente calibrado uma vez com o aplicativo *hmmcalibrate*. Para ambos os casos, foram utilizados os parâmetros padrão destes aplicativos.

```
Time taken to build model: 0.09 seconds

=== Stratified cross-validation ===
Correctly Classified Instances          81
Incorrectly Classified Instances        21
UnClassified Instances                   4
Total Number of Instances              106

=== Detailed Accuracy By Class ===
TP Rate   FP Rate   Precision   Recall
  0.863     0.275      0.759     0.863
  0.725     0.137      0.841     0.725

=== Confusion Matrix ===
  a  b   <-- classified as
 44  7 |  a = Y
 14 37 |  b = N
```

Figura 3: Resultado do classificador ID3, utilizando o Weka.

O limiar de detecção foi escolhido iterativamente de modo a minimizar o número de erros, sendo o mesmo estabelecido em -8,4. Um trecho de um arquivo HMM gerado é mostrado na figura 4.

```
HMMER2.0  [2.3.2]
NAME  prm5
LENG  23
ALPH  Nucleic
RF    no
CS    no
MAP   yes
COM   hmmbuild prm5.hmm prm5.msf
COM   hmmcalibrate prm5.hmm
NSEQ  48
DATE  Wed Mar 14 18:50:10 2007
CKSUM 3505
XT      -9967     -1  -1000  -1000  -9967
NULT      -1  -9967
NULE       0      0      0      0
EVD    -5.948497   0.714402
HMM         A      C      G      T
          m->m   m->i   m->d   i->m   i->i
           -29      *  -5644
     1  -3700  -1982   -548   1578    15
     -      0      0      0      0
     -     -1 -10779 -11821   -894  -1115
     2  -1621   -962  -1999   1542    16
     -      0      0      0      0
     -     -1 -10779 -11821   -894  -1115
     3  -3700  -1692   1640  -1008    17
     -      0      0      0      0
     -     -1 -10779 -11821   -894  -1115
     4   1057      1  -2689   -389    18
     -      0      0      0      0
     -     -1 -10779 -11821   -894  -1115
     5     92   1004  -1475   -814    19
     -      0      0      0      0
     -     -1 -10779 -11821   -894  -1115
     6    705  -1285   -324    215    20
     -    173    -90  -1010    522
     -  -5943    -24 -11821     -4  -8503
     7    374   -281   -141    -38    22
     -   -618   -181     31    531
     -  -5943    -24 -11821     -4  -8503
     8   -525   -164    -57    537    24
     -    132   -632   -212    481
     -  -5943    -24 -11821     -4  -8503
```

Figura 4: Resultado da saída do *hmmer*.

O maior interesse deste trabalho é na precisão preditiva dos classificadores, mensurada pela taxa de acerto. Assim, não foram comparados os quesitos de tempo de execução, utilização de memória e outros.

A tabela 1 mostra os resultados obtidos. A mesma está organizada por grupos de técnicas adotadas. Dentro de cada grupo, as técnicas estão organizadas em ordem decrescente da qualidade dos resultados. A tabela está organizada da seguinte maneira: na primeira coluna está o nome do grupo a que pertence o método (segundo a classificação do WEKA), na segunda coluna está o nome do método, da terceira à sexta colunas estão os dados da matriz de confusão, na sétima coluna está a taxa percentual de acerto.

Os dados da matriz de confusão são:

- tp: *true positive* – verdadeiro positivo: se refere ao número de exemplos positivos que foram corretamente classificados como tal;
- fn: *false negative* – falso negativo: se refere ao número de exemplos positivos que foram erroneamente classificados como negativos;
- fp: *false positive* – falso positivo: se refere ao número de exemplos negativos que foram erroneamente classificados como positivos;
- tn: *true negative* – verdadeiro positivo: se refere ao número de exemplos negativos que foram corretamente classificados como tal.

Na tabela se observa que, considerando apenas a taxa de acerto, o HMM e os métodos Bayesianos obtiveram as melhores taxas de acerto na classificação.

Grupo	Método	tp	fn	fp	tn	%
HMM	HMM	50	3	5	48	92,45
Bayes	HNB	49	4	3	50	93,40
	NaiveBayes	48	5	3	50	92,45
	NaiveBayesSimple	48	5	3	50	92,45
	NaiveBayesUpdateable	48	5	3	50	92,45
	AODE	50	3	7	46	90,56
Function	MultiplayerPerceptron	49	4	3	50	93,40
	SMO	49	4	4	49	92,45
	RBFNetwork	48	5	6	47	89,62
	Logistic	45	8	5	48	87,73
	VotedPerceptron	46	7	10	43	83,96
Meta	LogitBoost	47	6	5	48	89,62
	MultiBoostAB	47	6	7	46	87,73
	MultiClassClassifier	45	8	5	48	87,73
	ThresholdSelector	44	9	5	48	86,79
	ADABoost	46	7	8	45	85,84
Trees	NBTree	47	6	5	48	89,62
	LMT	47	6	6	47	88,67
	ADTree	47	6	8	45	86,79
	J48	45	8	12	41	81,13
	ID3	44	7	14	37	76,41
Lazy	LBR	48	5	3	50	92,45
	IB1	49	4	15	38	82,07
	Kstar	48	5	14	39	82,07
	IBk	49	4	16	37	81,13
	LWL	41	12	14	39	75,47
Rules	PART	44	9	11	42	81,13
	DecisionTable	41	12	11	42	78,30
	Ridor	41	12	11	42	78,30
	JRip	42	11	13	40	77,35
	NNge	31	22	2	51	77,35

Tabela 1: Resultados obtidos para HMM e outros 30 métodos de aprendizado de máquina.

Além de comparar as diversas técnicas de classificação de dados entre si, é importante comparar os resultados aqui obtidos com um referencial externo. Para tanto, são mostrados os resultados obtidos por Towell et al. (1990) utilizando a mesma base de dados, mas com outras técnicas. Neste trabalho os autores propuseram uma abordagem híbrida baseada em redes neurais e regras simbólicas que ficou conhecida como KBANN (*Knowledge Based Neural Network*). Também, no mesmo trabalho, o KBANN foi comparado com uma rede neural do tipo *perceptron* multicamadas, uma árvore de decisão induzida pelo algoritmo ID3, o algoritmo k-NN (k-vizinhos mais próximos) e a técnica conhecida como O'Neill. A tabela 2 reproduz estes resultados, onde observa-se que os mesmos foram bastante próximos àqueles obtidos no presente trabalho.

Técnica	Taxa de acerto (%)
KBANN	96,22
Perceptron multicamadas	92,45
O'Neill	88,68
k-NN	87,74
Árvore de decisão com ID3	82,08

Tabela2: Comparação entre diferentes métodos por Towell et al. (1990) utilizando o mesmo banco de dados.

No trabalho de Towell as redes neurais se mostraram bem mais eficientes que as demais técnicas utilizadas como o k-NN e a árvore de decisão, que são consideradas referenciais na área de Aprendizado de Máquina.

4 Discussão e conclusão

O presente trabalho mostrou uma comparação de diferentes métodos de Aprendizado de Máquina para o problema de detecção de regiões promotoras em DNA de *Escherichia coli*. Observou-se que, em geral, os métodos probabilísticos, tais como o HMM e métodos Bayesianos, obtiveram os melhores resultados. Uma característica interessante do HMM é que a escolha do limiar de decisão é livre e, portanto, pode ser alterada para um ajuste na especificidade do classificador, por exemplo. Após os métodos probabilísticos, os melhores resultados estão associados às redes neurais, possivelmente por sua capacidade de estabelecer hiperplanos complexos no espaço de separação das classes do problema.

Uma outra característica observada é o fato das técnicas de indução de árvores de decisão como ID3 e C45 (J48) não apresentarem um resultado tão bom quanto o observado em outras aplicações semelhantes. Estes classificadores são, em geral, referenciais importantes na área de mineração de dados e de Aprendizado de Máquina.

Para trabalhos futuros pretende-se estudar a associação de diferentes classificadores (técnicas de *boosting* e *bagging*), além de aplicar mesma metodologia a regiões promotoras de outros organismos, principalmente eucariotos, como parte de um sistema de detecção de genes.

Referências Bibliográficas

Durbin, R., Eddy, S., Krogh, A. e Mitchison, G. (1998). *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, Cambridge.

Ephraim, Y. e Merhav, N. (2002). Hidden Markov processes. *IEEE Transactions on Information Theory* **48**:1518–1569.

Harley, C. e Reynolds, R. (1987). Analysis of E. coli promoter sequences. *Nucleic Acids Research* **15**:2343–2361.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pp. 1137–1143.

Lehninger, A.L. (2006) Princípios de Bioquímica, 4°. edição. Ed. Sarvier, São Paulo.

Mount, D.W. (2001). *Bioinformatics: Sequence and Genome Analysis*, CSHL Press.

Souto, M.C.P., Lorena, A.C., Delbem, A.C.B. e de Carvalho, A.C.P.L.F. (2003). Técnicas de Aprendizado de Máquina para Problemas de Biologia Molecular. In: Anais da III Jornada de de Inteligência Artificial, pp. 103–152. Editora SBC

Towell, G., Shavlik, J. e Noordewier, M. (1990). Refinement of approximate domain theories by knowledge-based artificial neural networks. In: *Proceedings of the 8th National Conference on Artificial Intelligence* (AAAI-90), pp. 861–866.

Weinert, W., Lopes, H.S. (2004) Neural networks for protein classification. *Applied Bioinformatics* **3**(1):41–48.

Witten, I.H. e Frank, E. (2005). *Data Mining: Practical machine learning tools and techniques*, 2nd Edition, Morgan Kaufmann, San Francisco.

.