

UNIVERSIDADE DE CAXIAS DO SUL
CENTRO DE CIÊNCIAS AGRÁRIAS E BIOLÓGICAS
INSTITUTO DE BIOTECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM BIOTECNOLOGIA
NÍVEL DE DOUTORADO

**Redes neurais artificiais aplicadas no reconhecimento
de regiões promotoras em bactérias Gram-negativas**

Scheila de Avila e Silva

Caxias do Sul

2011

Scheila de Avila e Silva

Redes neurais artificiais aplicadas no reconhecimento de regiões promotoras em bactérias Gram-negativas

Tese apresentada ao Programa de Pós-Graduação em
Biotecnologia da Universidade de Caxias do Sul,
visando a obtenção do grau de Doutor em Biotecnologia.

Orientador: Prof. Dr. Sergio Echeverrigaray
Co-orientador: Prof. Dr. Günther J. L. Gerhardt

Caxias do Sul

2011

Dados Internacionais de Catalogação na Publicação (CIP)
Universidade de Caxias do Sul
UCS - BICE - Processamento Técnico

S586r Silva, Scheila de Avila e, 1980-
Redes neurais artificiais aplicadas no reconhecimento de regiões
promotoras em bactérias Gram-negativas / Scheila de Avila e Silva.
- 2011.
vii, 86 f. : il. ; 30 cm.

Tese (Doutorado) – Universidade de Caxias do Sul, Programa de
Pós-Graduação em Biotecnologia, 2011.
Apresenta apêndices e bibliografia.
Orientação: Prof. Dr. Sergio Echeverrigaray; co-orientação:
Prof. Dr. Günther J. L. Gerhardt.

1. Genética molecular. 2. Genes. 3. Redes neurais
(Computação). 4. Bactérias gram-negativas. I. Título.

CDU 2.ed. : 575

Índice para o catálogo sistemático:

1. Genética molecular	575
2. Genes	575.113
3. Redes neurais (Computação)	004.8
4. Bactérias gram-negativas	579.84

Catálogo na fonte elaborada pela Bibliotecária
Márcia Carvalho Rodrigues – CRB 10/1411

SCHEILA DE AVILA E SILVA

**Redes neurais artificiais aplicadas no reconhecimento de regiões
promotoras em bactérias Gram-negativas**

Tese apresentada ao Programa de Pós-Graduação em
Biotecnologia da Universidade de Caxias do Sul, visando a
obtenção de grau de Doutor em Biotecnologia.

Orientador: Prof. Dr. Sergio Echeverrigaray Laguna

Co-orientador: Prof. Dr. Günther J. L. Gerhardt

TESE APROVADA EM 19 DE DEZEMBRO DE 2011.



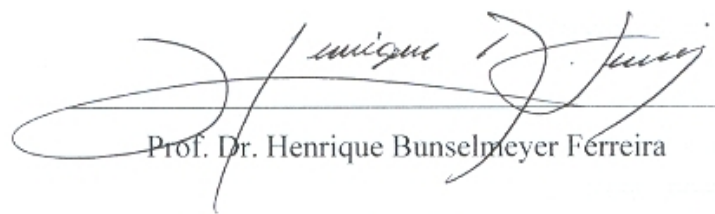
Prof. Dr. Sergio Echeverrigaray Laguna



Prof. Dr. Günther J. L. Gerhardt



Prof. Dr. Ney Lemke



Prof. Dr. Henrique Bunselmeyer Ferreira



Profa. Dra. Helena Graziottin Ribeiro

*No fim você vai ver que as coisas mais leves são
as únicas que o vento não conseguiu levar...*

Mário Quintana.

A João Carlos Sartor,
pelo carinho e companheirismo.

AGRADECIMENTOS

À minha família, pelo apoio em mais esta etapa de minha vida.

Ao meu orientador, Prof. Dr. Sergio Echeverrigaray, pelo apoio e contribuições realizadas ao longo da realização da tese.

Ao Prof. Dr. Günther J. L. Gerhardt, pela orientação durante a bolsa de Iniciação Científica na Graduação, pelo encaminhamento ao curso de mestrado e pelas contribuições realizadas.

Às Prof. Dr^a Ana Paula Longaray Delamare e Prof. Dr^a Helena Graziottin Ribeiro pelo acompanhamento e colaborações pertinentes realizadas.

À Universidade de Caxias do Sul e ao PPG em Biotecnologia pelo apoio ao projeto e ao Núcleo de Pesquisa em Bioinformática.

Ao Prof. Dr. Aldo J. P. Dillon pelo incentivo.

Ao Prof. Dr. Adelmo Cechin (*in memoriam*) pela orientação no desenvolvimento da dissertação de mestrado.

Às bolsistas de iniciação científica que contribuíram neste trabalho: Franciele Forte, Ivaine Taís Sauthier Sartor e Tahila Andrighetti.

Aos alunos de graduação que, ao realizar seu trabalho de conclusão contribuíram para o trabalho: Maurício Adami Mariani, Daniel José dos Santos, Vanessa Davanzo, Daíse Lima da Silva, Rodrigo Cicconet, Marlon Maciel Abreu.

Aos colegas do laboratório de Biotecnologia Vegetal e Microbiologia Aplicada pelo clima descontraído de trabalho e coleguismo nos créditos cursados.

Aos colegas do Colégio La Salle Caxias pela amizade.

À secretária do PPG, Lucimara Serafini, pela cordialidade e eficiência para tratar de questões burocráticas.

LISTA DE ABREVIATURAS

A	Nucleotídeo Adenina
AM	Aprendizado de Máquina
ART	Teoria da Ressonância Adaptativa
BacPP	<i>Bacterial Promoter Prediction</i>
BDBM	Banco de Dados de Biologia Molecular
BP	Algoritmo <i>Backpropagation</i>
C	Nucleotídeo Citosina
DNA	Ácido Desoxirribonucléico
<i>E. coli</i>	<i>Escherichia coli</i>
FN	Falsos Negativos
FP	Falsos Positivos
G	Nucleotídeo Guanina
k-FCV	<i>k-fold-cross-validation</i>
LSSVM	<i>Least Square Support Vector Machine</i>
MLP	<i>Multilayer Perceptron</i>
MPP	Matriz de Posições Ponderadas
NNPP	<i>Neural Networks Promoter Prediction</i>
nt	nucleotídeo
pb	Pares de Bases
RN	Rede Neural Artificial
RNA	Ácido Ribonucleico
RNA_m	Ácido Ribonucleico Mensageiro
RNA_p	Enzima RNA Polimerase
RP_c	Complexo Fechado do Promotor
RP_o	Complexo Aberto do Promotor
SGBD	Sistema de Gerência de Banco de Dados
SVM	Máquinas de Vetor de Suporte (<i>Support Vector Machine</i>)
T	Nucleotídeo Timina
TLS	Sítio de Início da Tradução
TSS	Sítio de Início de Transcrição
VN	Verdadeiros Negativos
VP	Verdadeiros Positivos

Sumário

1	INTRODUÇÃO	1
2	OBJETIVOS.....	4
	2.1Objetivos específicos.....	4
3	REVISÃO BIBLIOGRÁFICA.....	5
3.1	OS PROMOTORES E A TRANSCRIÇÃO DOS GENES.....	6
3.2	RECONHECIMENTO BASEADO EM SINAL.....	12
	3.2.1 Matriz de Posições Ponderadas	12
3.3	ANÁLISE POR APRENDIZADO DE MÁQUINA.....	15
	3.3.1 Máquinas de suporte vetorial – (Support vector machines).....	15
	3.3.2 Redes Neurais.....	17
3.4	METODOLOGIA UTILIZANDO A VALORES DE ESTABILIDADE.....	19
3.5	FUNDAMENTOS de REDES NEURAIAS ARTIFICIAIS.....	22
	3.5.1 Arquitetura das Redes Neurais.....	22
	3.5.2 Treinamento de Redes Neurais.....	25
3.6	EXTRAÇÃO DE REGRAS	27
	3.6.1 Extração de Regras a Partir de Redes Neurais.....	28
	3.6.2 Tipos de regras.....	29
	3.6.3 Regras obtidas a partir dos neurônios da camada oculta.....	29
3.7	CONSIDERAÇÕES ADICIONAIS.....	30
4	METODOLOGIA.....	31
4.1	ORGANISMOS ESTUDADOS.....	31
4.2	BANCOS DE DADOS.....	32
4.3	FERRAMENTAS.....	33
4.4	CRIAÇÃO DE BANCO DE DADOS DE REGIÕES INTERGÊNICAS.....	34
5	RESULTADOS	35

5.1	CAPÍTULO I - <i>Rules extraction from neural networks applied to prediction and recognition of prokaryotic promoters</i>	36
5.2	CAPÍTULO II - <i>BacPP: Bacterial promoter prediction - A tool for accurate sigma-factor specific assignment in enterobacteria</i>	45
5.3	CAPÍTULO III - <i>Neural Networks applied to bacterial promoter prediction based on DNA stability</i>	46
5.4	CAPÍTULO IV – Banco de dados Intergenicdb.....	58
6	CONSIDERAÇÕES FINAIS.....	68
7	REFERÊNCIAS BIBLIOGRÁFICAS.....	70
	APÊNDICE 1 - PATENTE INTERNACIONAL DA FERRAMENTA BacPP.....	75

Índice de tabelas

Tabela 3.1: Descrição das subunidades da RNA polimerase holoenzima de E. coli. (LEHNINGER et al., 2007).....	7
Tabela 3.2: Fatores σ de E. coli (LEWIN, 2008).	8
Tabela 3.3: Resultados obtidos pela metodologia de Jacques et al. (2006).....	14
Tabela 3.4: Resultados obtidos pelo trabalho de Rani et al. (2006).....	19

Índice de ilustrações

Figura 3.1: Dogma central da biologia molecular (LEWIN, 2008).....	6
Figura 3.2: Representação da região promotora para uma única fita de DNA em E. coli. (BURDEN et al., 2005-modificado).....	7
Figura 3.3: Esquema da RNAP de organismos procariotos (LEWIN,2008).....	7
Figura 3.4: Promotor procariótico reconhecido pelo fator $\sigma 70$ (LEHNINGER et al., 2007).....	8
Figura 3.5: Promotores típicos de E. coli reconhecidos pela RNAP holoenzima $\sigma 70$ (MADIGAN, 2010).	9
Figura 3.6: Etapas da iniciação da transcrição de E. coli (LEWIN et al., 2008).....	10
Figura 3.7: Exemplo da transformação da matriz de alinhamento para a matriz de posições ponderadas para a sequência teste AGGTGC.....	13
Figura 3.8: Treinamento de SVM (RUSSEL e NORVIG, 2003).....	15
Figura 3.9: Fluxograma que ilustra a metodologia desenvolvida pelos autores Polat e Günes (2007).....	16
Figura 3.10: Codificação ortogonal para os valores de estabilidade, empregado por Askary et al. (2009).....	18
Figura 3.11: Fluxograma da metodologia descrita por Rangannan e Bansal (2007). 21	
Figura 3.12: Analogia entre neurônios biológicos e artificiais. (WU e MCLARTY, 2000).....	22
Figura 3.13: Modelo de um neurônio artificial (RUSSELL e NORVIG, 2003).....	23
Figura 3.14: Rede MLP com três camadas (RUSSELL e NORVIG, 2003).....	24
Figura 3.15: Ilustração das três regiões definidas na função sigmóide para análise dos dados de entrada e extração de regras.....	30
Figura 4.1: Estrutura da metodologia proposta para o uso de RN no reconhecimento e predição de promotores.....	32

RESUMO

A região promotora é uma sequência de DNA localizada anteriormente à uma região codificante e é responsável por iniciar o processo de transcrição. Deste modo, atua como um elemento regulador. O estudo da regulação da expressão gênica auxilia na compreensão da maquinaria vital dos seres vivos, no conhecimento sobre a funcionalidade dos genes em diferentes espécies, na resposta celular frente às mudanças ambientais, entre outras questões. Embora os métodos computacionais para a predição de genes possuam uma boa acurácia o mesmo não é conseguido para os promotores. Esta dificuldade se deve ao tamanho reduzido do promotor e ao padrão pouco conservado, o que gera resultados com alto número de falsos positivos. Esta tese teve como objetivo a utilização de Redes Neurais Artificiais na predição, caracterização e reconhecimento de promotores de bactérias Gram-negativas. Diferente de outros trabalhos, a predição realizada não foi limitada apenas aos promotores dos genes constitutivos; foi realizada também para as demais classes de sequências promotoras. Além da abordagem clássica utilizando a composição de nucleotídeos foram empregados os valores de estabilidade da sequência. De modo a otimizar o aprendizado da Rede Neural e implementar uma ferramenta própria para a predição de promotores, foram extraídas regras de inferência (baseadas no conhecimento produzido durante o treinamento da rede) que foram ponderadas e implementadas em uma nova ferramenta, chamada BacPP. Até o presente, os resultados obtidos com o BacPP foram satisfatórios e comparáveis com a literatura. Os valores de exatidão obtidos com o BacPP para os fatores σ^{24} , σ^{28} , σ^{32} , σ^{38} , σ^{54} e σ^{70} de *E. coli* foram, 86,9%; 92,8%; 91,5%; 89,3%; 97,0%; 83,6%, respectivamente. Quando a ferramenta foi aplicada em promotores pertencentes a outras bactérias Gram-negativas, a exatidão geral foi de 76%. Considerando a importância da predição de promotores e a ausência de banco de dados com informações para outras bactérias, implementou-se o IntergenicDB, um banco de dados com diversas informações sobre as sequências intergênicas e o valor de classificação destas para os diferentes fatores σ bacterianos, conforme os resultados obtidos com o BacPP.

ABSTRACT

The promoter region is located some few base pairs before a coding region. It is responsible for initiating gene expression process, thus, it can play a regulatory role. The study about gene expression regulation can assist mainly in the comprehension of complex metabolic network presented by several organisms and cellular answer considering the environment changes. The computational methods to gene prediction have a good accuracy, but this is not achieved in promoter prediction. This difficulty occurs because of the length of the promoter and its degenerate pattern. Those features can explain results with a great number of false positives present in the literature. The present thesis has as its main goal the neural networks applied to Gram-negative promoter prediction, recognition and characterization. Beside the classical approach with the nucleotides of the sequence, the prediction was also made by using stability values. Aiming at developing a own tool for bacterial promoter prediction, the rules extraction was carried out and the results were weighted and implemented. This tool, named BacPP, presents results comparable with the related literature. Currently, the BacPP specific accuracy for σ^{24} , σ^{28} , σ^{32} , σ^{38} , σ^{54} and σ^{70} were 86,9%; 92,8%; 91,5%; 89,3%; 97,0%; 83,6%, respectively. Furthermore, when challenged with promoter sequences belonging to other enterobacteria BacPP maintained 76% accuracy overall. Currently, there is no databases dedicated for other Gram-negative promoter than *E.coli*. For this reason, IntergenicDB was modeled and implemented. This database was projected to collect several pieces of information about the sequences and the organisms to which they belong and, the classification results originated from BacPP for each sequence.

1 INTRODUÇÃO

Os fenômenos biológicos são muito complexos e requerem a integração de muitas áreas do conhecimento para a comprovação ou refutação de hipóteses. A interface interdisciplinar mais antiga (e talvez a mais conhecida) entre a Biologia e as Ciências Exatas é a Bioestatística. Gradualmente nos últimos anos, a Biologia tem utilizado, as ferramentas proporcionadas pela Informática e pela Matemática para a resolução de problemas nos mais diversos campos: desde a Genética até a Ecologia (BARRERA *et al.*, 2004).

Um dos maiores desafios da era pós-genômica é a determinação de quando, onde e como os genes são “ligados” e “desligados”. A diferença entre duas espécies está muito mais relacionada com a transcrição de seus genes do que com a estrutura destes em si. Assim, o estudo da regulação gênica contribui para a construção do conhecimento a respeito da funcionalidade dos genes em diferentes espécies, na questão da diferenciação celular em organismos multicelulares, na resposta celular frente às mudanças ambientais, entre outras questões (HOWARD e BENSON, 2003; COTIK *et al.*, 2005).

Dentre as sequências de DNA que atuam como reguladoras da expressão gênica estão incluídas as regiões promotoras. De uma maneira simplificada, pode-se dizer que estas localizam-se anteriormente à região codificante e interagem com a enzima RNA polimerase (RNAP), desencadeando o processo de transcrição (LEWIN, 2008). Fazendo uma analogia, os elementos *downstream* (como os genes) representam a memória de um computador e os elementos *upstream* (como os promotores) os programas que atuam nesta memória. Assim, o estudo dos promotores pode prover modelos sobre a constituição do “programa” e de como este opera (HOWARD e BENSON, 2003).

Em organismos procarióticos, a holoenzima RNAP é formada por cinco subunidades e uma subunidade adicional (que se liga de forma transitória) chamada fator sigma (σ). A coleção de diferentes σ é responsável pela ligação da RNAP em determinadas regiões dos promotores e a consequente expressão de genes específicos de resposta às mudanças ambientais. Os fatores σ são nomeados conforme seu peso molecular (σ^{24} , σ^{28} , σ^{32} , σ^{38} , σ^{54} e σ^{70}) e estão relacionados com determinadas funções metabólicas e/ou fisiológicas. Por exemplo, σ^{32} e σ^{24} desempenham papel na resposta ao estresse por choque térmico, σ^{28} está associado com a expressão de genes produtores de cílios e flagelos, σ^{54} está envolvido na fixação de nitrogênio e σ^{70} está relacionado com a expressão de genes constitutivos (LEWIN, 2008).

A região promotora possui locais específicos e com certo grau de conservação, que auxiliam no reconhecimento e na ligação da RNAP nesta região. Além destes locais, os promotores possuem algumas características estruturais próprias, diferentes das regiões não-promotoras, que podem ser incorporadas nos estudos destes elementos, tais como a deformabilidade, estabilidade e a curvatura. (KANHERE e BANSAL, 2005a; KOZOBAYAVRAHAM *et al.*, 2008).

As técnicas moleculares para a identificação de promotores são custosas e consomem muito tempo, o que permite que as abordagens *in silico* ganhem aplicabilidade (TOWSEY, 2008). As mais variadas abordagens computacionais têm sido empregadas para reconhecer estas regiões e predizer se uma região é ou não promotora. Dentre estas técnicas, pode-se destacar Análise Probabilística, Reconhecimento de Padrões e Aprendizado de Máquina (AM). Embora haja progressos na predição e análise de promotores, estes ainda estão longe de possuir uma alta acurácia (RANI *et al.*, 2006).

A maioria dos trabalhos relacionados são aplicados apenas às sequências promotoras reconhecidas pelo fator σ^{70} . Esta tese tem como tema a aplicação de Redes Neurais Artificiais (RN) na predição, reconhecimento e caracterização de regiões promotoras procarióticas conforme o fator σ que as reconhece. Além da composição de nucleotídeos (nt), suas propriedades estruturais (valores de estabilidade) foram utilizadas no treinamento da RN.

A partir da análise dos resultados obtidos, com as simulações de RN, foi realizada a extração de regras a partir das arquiteturas treinadas para cada fator σ .

A extração de regras é um elemento importante no levantamento de hipóteses pois permite a visualização de como ocorreu o processo de aprendizagem pela rede, uma vez que verifica-se quais elementos da sequência possuem um papel determinante no seu reconhecimento como promotora (ANDREWS *et al.*, 1995). As regras foram ponderadas e implementadas em um programa de predição de promotores procarióticos, chamado de BacPP. Ao analisar uma determinada sequência, o programa atribui um valor de classificação para os fatores σ bacterianos descritos neste trabalho. Considerando a falta de informações sobre outras bactérias Gram-negativas, surgiu a necessidade da implementação de uma base de dados relacionada (TOWSEY *et al.*, 2008). O IntergenicDB foi modelado para armazenar informações relevantes sobre a estrutura e bibliografia das sequências intergênicas de bactérias Gram-negativas, além de armazenar os valores de predição obtidos com a ferramenta BacPP.

O presente trabalho está organizado em 4 seções principais. A seção 3 é constituída de uma revisão bibliográfica geral, na qual são apresentados os conceitos biológicos e computacionais relevantes para a compreensão de como os resultados foram obtidos. Uma visão geral da metodologia é apresentada na seção 4, sendo que os detalhes da metodologia são apresentados nos capítulos da seção dos resultados. A seção 5, mostra os resultados na forma de artigos científicos publicados e/ou a serem submetidos à publicação em periódicos científicos.

2 OBJETIVOS

O objetivo geral deste trabalho é reconhecer, predizer e caracterizar regiões promotoras de diferentes bactérias gram-negativas, integrando dados físico-químicos da molécula de DNA com a composição da sequência por meio de uma abordagem de Redes Neurais Artificiais.

2.1 OBJETIVOS ESPECÍFICOS

- Preparar os dados de entrada para a realização do treinamento;
- Determinar a melhor arquitetura de RNs para a identificação de regiões promotoras de acordo com o fator σ que reconhece a sequência, utilizando a informação dos nt e/ou estabilidade da sequência;
- Extrair regras de cada RN treinada para compreensão dos mecanismos utilizados no reconhecimento de promotores;
- Desenvolver uma ferramenta própria para a predição de promotores com base no aprendizado da RN;
- Aplicar a ferramenta desenvolvida em regiões intergênicas de bactérias Gram-negativas;
- Criar de um banco de dados de possíveis promotores procarióticos utilizando diferentes metodologias disponíveis;

3 REVISÃO BIBLIOGRÁFICA

O DNA ou ácido desoxirribonucléico é a molécula universal mais empregada no armazenamento da informação genética (DE ROBERTIS, 1993). Os genes são um segmento da molécula de DNA que contém a informação necessária para a codificação de seus produtos. Na maioria das vezes, estes produtos são proteínas que realizam uma função específica na célula: estrutural, regulatória ou catalítica. O controle de qual gene deve ser expresso em um determinado momento compreende um conjunto de mecanismos que torna este processo complexo até mesmo para organismos unicelulares, como as bactérias. Este processo é conhecido como regulação da expressão gênica.

O estudo de promotores é um dos aspectos fundamentais para a compreensão da expressão gênica. Ainda que os promotores sejam de importância indiscutível, a habilidade em identificá-los é menos desenvolvida que a de encontrar regiões codificantes. A maior dificuldade no seu reconhecimento *in silico* é que sua sequência é muito curta e não apresenta-se completamente conservada (HOWARD e BENSON, 2003; BURDEN, *et al.*, 2005; KANHERE e BANSAL, 2005b; SIVARAMAN *et al.*, 2005).

As próximas seções descrevem o processo de transcrição dos genes em organismos procariotos, o papel do promotor para o seu desencadeamento e as abordagens *in silico* para a predição de sequências promotoras, sendo que esta seção foi submetida para publicação como capítulo de livro. Além disso, são apresentados os fundamentos sobre as RNs, já que estas foram escolhidas como a técnica de AM da metodologia deste trabalho.

3.1 OS PROMOTORES E A TRANSCRIÇÃO DOS GENES

Quando um gene é expresso, sua informação é copiada na forma de ácido ribonucleico (RNA) que por sua vez, dirige a síntese dos produtos elementares dos genes. Este processo é denominado como dogma central da Biologia Celular, que pode ser visualizado na Figura 3.1.

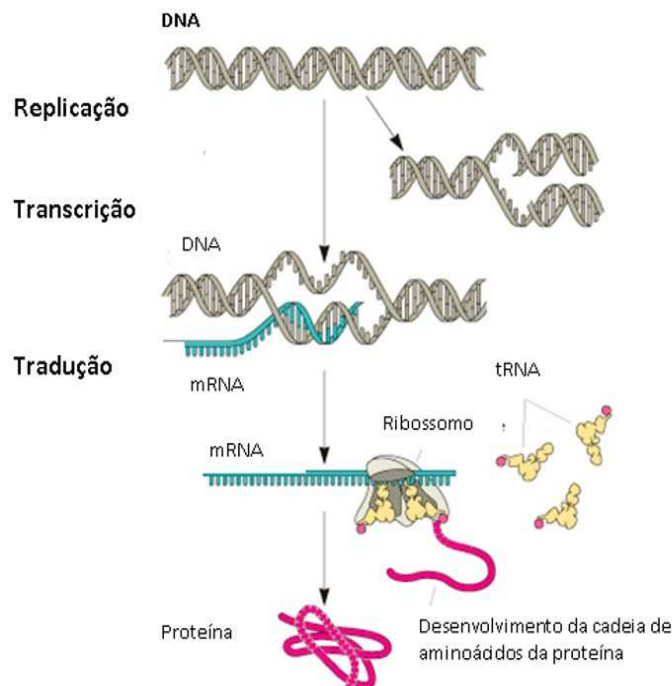


Figura 3.1: Dogma central da biologia molecular (LEWIN, 2008).

A iniciação da transcrição dos genes inicia quando a enzima RNAP liga-se em sequências específicas do DNA, denominadas de promotores. Em *E. coli*, a ligação da RNAP ocorre dentro de uma região que se estende desde cerca de 70 pares de base (pb) antes do sítio de início da transcrição (TSS) até cerca de 30 pb além dele. Por convenção, os pb de DNA que correspondem ao início de uma molécula de RNAm recebem números positivos, sendo esta parte do DNA denominada de região *downstream*. Já a região *upstream* corresponde aos nt que precedem o sítio de início da transcrição recebem números negativos (Figura 3.2).

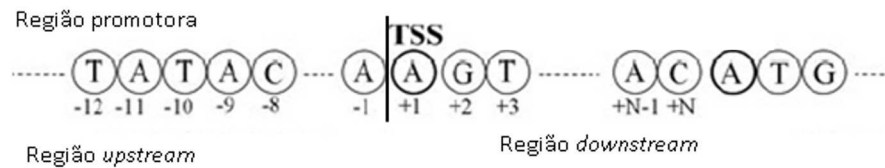


Figura 3.2: Representação da região promotora para uma única fita de DNA em *E. coli*. (BURDEN *et al.*, 2005-modificado).

A enzima RNAP desempenha um importante papel no início da transcrição como reconhecedora das sequências promotoras. Ela contém cinco unidades básicas: duas subunidades α , uma subunidade β , uma β' e uma subunidade ω . Essas cinco subunidades formam o *core* da RNAP. Além destas, há uma subunidade designada fator σ , que liga-se transitoriamente ao *core* e direciona a enzima para sítios de ligação específicos do DNA. Quando o fator σ está associado à RNAP, ela passa a ser chamada de RNAP holoenzima. Na Tabela 3.1, estão as subunidades da RNAP, seu gene codificante e sua função no processo de transcrição e na Figura 3.3 encontra-se um esquema desta enzima.

Tabela 3.1: Descrição das subunidades da RNA polimerase holoenzima de *E. coli*. (LEHNINGER *et al.*, 2007).

Subunidades	Função na RNAP
α	Ligação a proteínas regulatórias
β e β'	Atividade catalítica
σ	Reconhecimento do promotor e especificidade
Ω	Acréscimo na força de associação entre as subunidades

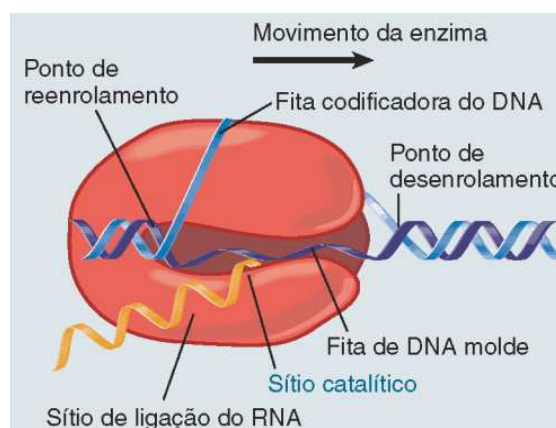


Figura 3.3: Esquema da RNAP de organismos procariotos (LEWIN,2008).

Existem diferentes tipos de fatores σ (Tabela 3.2) que podem se ligar com o *core* da RNAP, sendo cada um associado a uma classe de promotores que regulam a expressão de um determinado conjunto de genes necessários em um dado momento celular.

Tabela 3.2: Fatores σ de *E. coli* (LEWIN, 2008).

Fator σ	Nome do Gene	Função	Consenso -35	separador -10 ¹
28	fliA	Produção de cílios e flagelos	CTAAA 15 pb	GCCGATAA
32	rpoH	Estresse por choque térmico	CCCTTGAA 13-15pb	CCCGATNT
38	rpoS	Resposta a estresse	TTGACA 16-18pb	TATACT
54	rpoN	Assimilação de nitrogênio	CTGGNA 6pb	TTGCA
70	rpoD	Sigma constitutivo	TTGACA 16-18pb	TATAAT
24	rpoE	Estresse por choque térmico	GGAAGTT 15pb	GTCTAA
H	sigH	Estresse osmótico	AGGANPuPu 11-12	GCTGAATCA

¹ Somente para σ^{54} , a região consensual se localiza centrada nos nt -12 e -24. Na sequência consensual, N significa qualquer nucleotídeo e Pu significa nucleotídeo de base púrica.

Um promotor procariótico típico para σ^{70} é constituído de 3 regiões características: uma sequência de 6 nt (hexâmero) centrada em -35 do ponto inicial de transcrição (+1), outro hexâmero centrado em -10 e a sequência que separa os hexâmeros (espaçador), conforme ilustrado na Figura 3.4.



Figura 3.4: Promotor procariótico reconhecido pelo fator σ^{70} (LEHNINGER *et al.*, 2007).

Análises e comparações das sequências da classe mais comum de promotores bacterianos (reconhecidos pela RNAP holoenzima contendo σ^{70}) revelam semelhanças nos dois hexâmeros citados anteriormente. Embora as sequências não sejam idênticas para todos os promotores bacterianos, certos nt comuns em determinadas posições formam uma sequência consenso (Figura 3.5). O modelo biológico padrão para estes promotores é a sequência TTGACA para a região -35, TATAAT para a região -10 e um espaçamento entre estes hexâmeros de 16-18 nt.

Muitas linhas independentes de pesquisa atestam a importância funcional das sequências -35 e -10 (LEHNINGER *et al.*, 2007; KALATE *et al.*, 2003; KANHERE e BANSAL, 2005a). O hexâmero -35 funciona como sinal para reconhecimento pela

RNAP e o hexâmero -10 permite converter o complexo fechado em complexo aberto. Além disso, a distância entre eles parece ser relevante, apesar do tamanho variável e da falta de conservação (LEWIN, 2008).

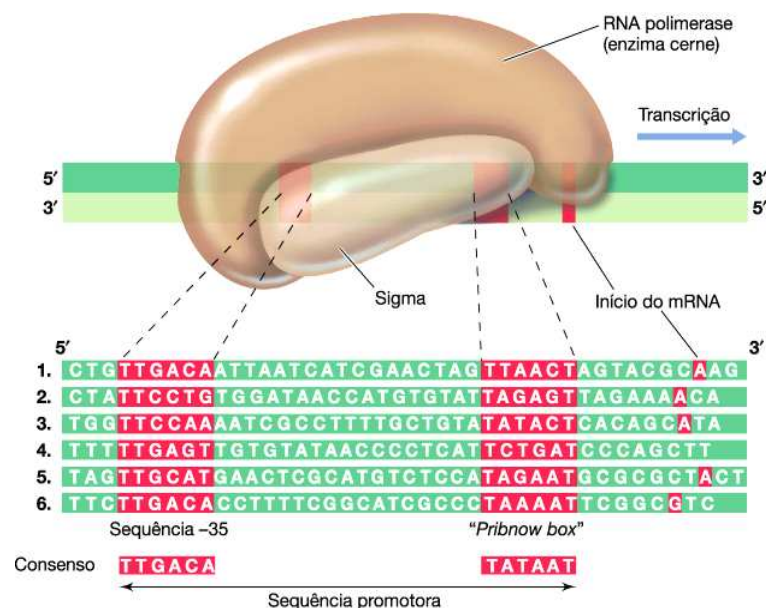


Figura 3.5: Promotores típicos de *E. coli* reconhecidos pela RNAP holoenzima σ^{70} (MADIGAN, 2010).

Variações na sequência consenso podem afetar a eficiência da ligação da RNAP. Uma mudança em apenas um pb pode diminuir a velocidade de iniciação em várias ordens de grandeza. A sequência do promotor, desta forma, estabelece um nível basal de expressão dos genes, sendo considerado um promotor forte aquele mais perto da sequência consensual e um promotor fraco aquele que possui mais de três nt diferentes do consenso (LEWIN, 2008). No entanto, alguns promotores procarióticos permanecem funcionais mesmo na ausência da região -35. Estes promotores possuem uma região chamada -10 estendida (duas bases extra no hexâmero -10). Ainda é incerto se o hexâmero -10 estendido é um antecessor do promotor bipartido ou vice-versa. A explicação para a origem dos promotores bipartidos ainda é desconhecida. A informação da região -10 estendida é importante para o bom funcionamento da RNAP e esta informação foi realocada na região -35 (HOOK-BARNARD *et al.*, 2006; SHULTZABERGER *et al.*, 2007).

A transcrição possui dois momentos principais, cada um com múltiplas etapas. O início do processo ocorre quando a RNAP holoenzima se liga ao promotor,

formando um complexo fechado do promotor – RP_c – no qual o DNA ligado está na forma de fita dupla. Após, há a formação do complexo aberto – RP_o – onde o DNA desta região está parcialmente desenrolado. Em seguida, inicia-se a transcrição deste complexo (etapa de transcrição abortiva) e após a inserção dos dois primeiros nt na molécula de RNA transcrita. O término da transcrição ocorre quando é encontrada uma sequência de nt denominada de região terminadora. Este processo está ilustrado na Figura 3.6.

Apesar da simplicidade com a qual os livros de Biologia Molecular (LEWIN, 2008; LEHNINGER *et al.*, 2007) descrevem os promotores, percebe-se em trabalhos experimentais como os de Naryshkin *et al.* (2000), Burgess e Anthony (2001), Murakami *et al.* (2002), Touloukhonov e Landick (2006), Borukhov e Nudler (2007), que a interação entre a enzima e a sequência promotora é um processo complexo, que envolve a interação de vários locais da RNAP com a sequência.

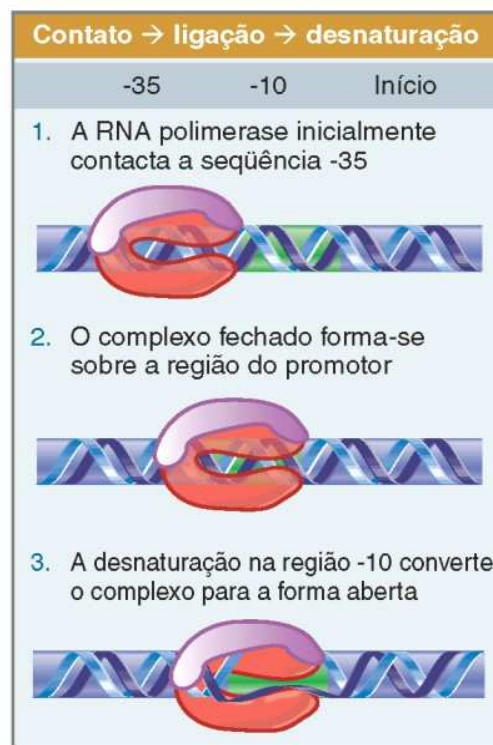


Figura 3.6: Etapas da iniciação da transcrição de *E. coli* (LEWIN *et al.*, 2008).

Outros fatores além das pontes de hidrogênios e proteínas ligantes regulam o reconhecimento da sequência alvo em promotores. Como sugestões iniciais têm-se

as propriedades físicas do DNA, tais como susceptibilidade à DNaseI, deformabilidade, estabilidade e curvatura (GOÑI *et al.*, 2007). Existem diferenças estruturais entre as regiões *upstream* (onde localiza-se a região promotora) e *downstream* (onde localiza-se a região codificante) que podem ser consideradas para melhorar a predição das sequências promotoras e caracterizá-las. É muito difícil acreditar que apenas os motivos consensuais sejam os responsáveis pela interação RNAP-promotor, já que estes motivos são pequenos e não completamente conservados. É possível que as sequências vizinhas a estes motivos também estejam envolvidas neste processo de interação RNAP-promotor (KANHERE e BANSAL, 2005a; RAMPRAKASH e SCWARZ, 2007).

A importância destas propriedades para os promotores e para o processo de transcrição está relacionado com a formação do RP_o, que envolve a separação das fitas de DNA. Esta separação é um processo termodinamicamente desfavorável e ocorre sem nenhuma ajuda energética de fonte externa. Aqui, a pouca estabilidade da sequência promotora pode auxiliar no início de separação das fitas. Trabalhos como os de, Jáuregui *et al.* (2003), Kanhere e Bansal (2005a, 2005b) e Ramprakash e Schwarz (2007) mostram que as regiões promotoras são menos estáveis que as regiões gênicas. Outra propriedade, como a curvatura, pode ser definida como a dupla fita curvada em um axis helicoidal. Muitas sequências, de organismos eucariotos e procariotos mostram que as regiões *upstream* são mais curvadas que as regiões codificantes (BORUKHOV e NUDLER, 2008). Já a deformabilidade, refere-se ao afrouxamento com o qual a molécula pode realizar uma curva em alguma direção. Sabe-se que a deformabilidade é importante para a ligação de fatores de transcrição e evidências experimentais sugerem que a sequência promotora se enrola ao redor da RNAP (KANHERE e BANSAL, 2005a, 2005b; RAMPRAKASH e SCHWARZ, 2007; KOZOBAY-AVRAHAM *et al.*, 2008). Muitos estudos mostram que a deformabilidade e a curvatura tem papel no mecanismo de transcrição, entretanto estes mecanismos ainda não são totalmente compreendidos. Na análise de promotores, algumas questões permanecem sem respostas com respeito à curvatura do DNA: a curvatura é um componente essencial e integral dos promotores? Ela pode ser usada como característica discriminante entre promotores e outras sequências?(PANDEY e KRISHNAMACHARI, 2006).

A flexibilidade da fita de DNA é outra característica física da sequência. O

trabalho de Thiagarajan *et al.* (2006) verificou que na região consensual dos promotores de *E. coli* há dois nt flexíveis entre dois não flexíveis, sendo esta uma característica determinante da região -10 para determinar a força de expressão de um determinado promotor, talvez pela influência de formação do complexo aberto do promotor.

Apesar destas evidências em relação à estrutura da sequência promotora, estas informações não são amplamente utilizadas na predição e reconhecimento dos promotores, conforme descrito nas próximas seções, que apresentam os principais trabalhos referentes à análise de promotores procarióticos *in silico*. A literatura apresenta muitas abordagens para o reconhecimento e predição de promotores. Dentre estas pode-se citar: (i) metodologia baseada em sinal, que opera no reconhecimento de sinais relativamente conservados através de alinhamento e homologia entre promotores previamente identificados; (ii) AM, que usa conjunto de informações estruturais e funcionais disponíveis sobre as sequências promotoras para “aprender” a reconhecê-los automaticamente e produzir hipóteses relevantes sobre estas sequências. Aqui, encontram-se as metodologias de RNs e *Support Vector Machines* (SVM).

A seguir, estão apresentados os principais métodos computacionais encontrados na literatura de predição de promotores acompanhados da discussão das implicações que tornam este campo de pesquisa ainda latente.

3.2 RECONHECIMENTO BASEADO EM SINAL

A metodologia de reconhecimento de promotores baseado em sinal emprega principalmente a comparação do conteúdo de diferentes sequências promotoras. Os trabalhos clássicos e alguns mais recentes estão apresentados a seguir.

3.2.1 Matriz de Posições Ponderadas

A metodologia de matriz de posições ponderadas (MPP) consiste em alinhar um conjunto de sequências identificadas previamente como promotoras e pesquisar por regiões conservadas em seu conteúdo. Conforme Hertz e Stormo (1999), as matrizes de posições ponderadas assumem que cada linha corresponde a um dos nt e cada coluna a um alinhamento. Os elementos da matriz são os pesos utilizados para pontuar uma sequência teste, conforme uma medida que quantifica a aderência

ao modelo. A pontuação é dada pela soma dos pesos de cada letra alinhada em cada posição (Figura 3.7).

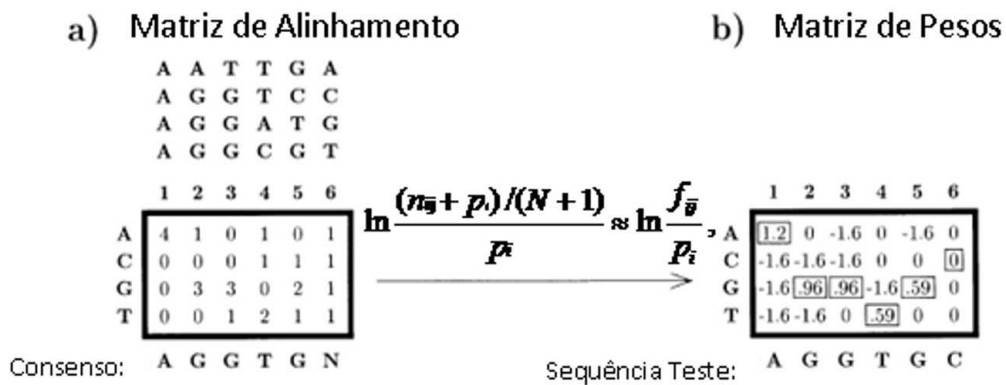


Figura 3.7: Exemplo da transformação da matriz de alinhamento para a matriz de posições ponderadas para a sequência teste AGGTGC.

A transformação ilustrada na Figura 3.7, foi criada a partir da fórmula 3.1:

$$\ln \frac{(n_{ij} + p_i)/(N + 1)}{p_i} \approx \ln \frac{f_{ij}}{p_i}, \quad (3.1)$$

No exemplo mostrado (Figura 3.7), em (a) está uma tabela de alinhamento para os 4 hexâmeros localizados no topo da figura. Embaixo da matriz está a sequência consenso correspondente ao alinhamento (N indica que não há nucleotídeo preferencial). Após a aplicação da fórmula 3.1, foi gerada a matriz de pesos (b), derivada da matriz de alinhamento (a). Os números destacados são os responsáveis pela pontuação global da sequência (Hertz e Stormo, 1999).

Jaques *et al.* (2006), desenvolveu uma nova abordagem de predição de promotores com base nas matrizes de representação da distribuição genômica de hexanucleotídeos. Esta metodologia foi utilizada para dez organismos procarióticos (Tabela 3.3). A sensibilidade das matrizes geradas para cada organismo variou de 29.4% (*C. glutamicum*) a 90,9% (*B. japonicum*), conforme mostrado na Tabela 3.3. Para a matriz gerada para *E. coli*, a sensibilidade apresentada foi de 42,4%.

Quando os resultados deste trabalho são comparados com a literatura, percebe-se que a sensibilidade da predição de promotores de *E. coli* não é melhorada. No entanto, ele demonstra os recentes esforços para ampliar a predição para outras bactérias e, além disso, mostra que a distribuição genômica dos

elementos regulatórios é significativamente diferente dos elementos não-regulatórios.

Tabela 3.3: Resultados obtidos pela metodologia de Jacques *et al.* (2006)

Organismos	Conteúdo de GC (%)	Sensibilidade (%)
<i>E. coli</i>	50,8	42,4
<i>B. subtilis</i>	43,5	56,8
<i>C. glutamicum</i>	53,8	29,4
<i>M. pneumoniae</i>	40,0	43,3
<i>M. tuberculosis</i>	65,6	57,1
<i>S. coelicolor</i>	72,1	58,8
<i>H. pylori</i>	38,9	47,1
<i>C. jejuni</i>	30,5	42,9
<i>B. japonicum</i>	64,1	90,9
<i>S. aureus</i>	32,9	37,5

A MPP foi utilizada, também, no software *Virtual Footprint* (MUNCH *et al.*, 2005). Esta ferramenta tem o objetivo de reconhecer padrões em sequências de DNA e está disponível no site <http://prodoric.tu-bs.de/vfp/index2.php> (acessado em 29 de julho de 2010). A ideia central do trabalho baseia-se no princípio que os promotores são mais representados nas regiões intergênicas do que no resto do genoma. O valor de classificação foi calculado na semelhança entre a matriz de representação da distribuição genômica dos promotores encontrados na literatura e a matriz de representação da distribuição de prováveis promotores.

Uma variação da MPP foi desenvolvida por Li e Lin (2006) que obteve sensibilidade de 91% e especificidade de 81% usando 683 sequências experimentalmente identificadas como promotores de *E. coli* reconhecidos pelo fator σ^{70} . A matriz desenvolvida por eles foi chamada de “Matriz de Valores Posição-Correlação” e baseia-se na medida da conservação das sequências.

Quando compara-se a MPP com o simples alinhamento de sequências, percebe-se que a ponderação melhora os resultados obtidos. A importância e o pioneirismo deste trabalho são indiscutíveis para a análise dos promotores, já que existem recentes publicações que mostram um certo grau de conservação dos motivos (Cotik *et al.*, 2005; Sivaraman *et al.*, 2005). No entanto, somente a análise dos nt da sequência para a descoberta de novos promotores é uma abordagem limitada, já que: (i) a variação dos nt é grande; (ii) assume a independência entre bases adjacentes; (iii) não permite a presença de múltiplos elementos dos

promotores, inserções, deleções ou espaço variável entre os elementos e (iv) o resultado pode variar de acordo com o método de alinhamento (SONG *et al.*, 2007).

3.3 ANÁLISE POR APRENDIZADO DE MÁQUINA

Nesta seção são descritas as metodologias de SVM e RN já que estas são as metodologias as mais empregadas na predição de promotores e mostram resultados promissores.

3.3.1 Máquinas de suporte vetorial – (*Support vector machines*)

O algoritmo das Máquinas de Vetor de Suporte ou Máquinas de Suporte Vetorial foi proposto por Boser *et al.* (1992) e pode ser utilizado para classificações de padrões e regressão linear. Basicamente, as SVM são uma máquina linear com algumas propriedades muito interessantes. No caso das classificações, a idéia principal é construir um hiperplano como superfície de decisão, de tal forma que a margem de separação entre exemplos positivos e negativos seja máxima (Figura 3.8). As SVMs podem fornecer um bom desempenho de generalização em problemas de classificação de padrões, apesar de não incorporarem conhecimento do domínio do problema e apresentam limitações com a escolha do *kernel* (HAYKIN, 1999).

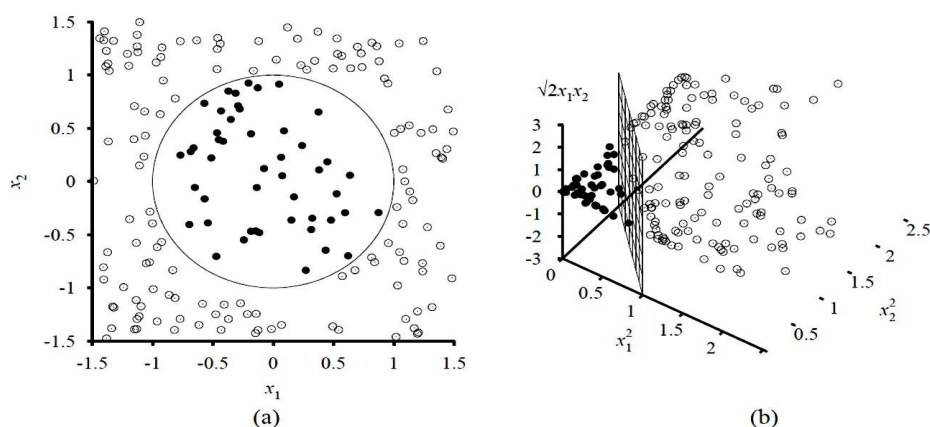


Figura 3.8: Treinamento de SVM (RUSSEL e NORVIG, 2003).

Em (a) treinamento de duas dimensões com os exemplos positivos representados pelos círculos pretos e os exemplos negativos pelos círculos brancos. Em (b) o mesmo conjunto de dados após mapeamento em um espaço tridimensional.

Polat e Günes (2007) usaram uma combinação de seleção de características e LSSVM (*least square support vector machine*), conforme ilustrado na Figura 3.9. Esta metodologia mostra uma acurácia de 84,6%, sensibilidade de 90% e especificidade de 80%. Apesar destes índices serem elevados, ressalta-se que neste trabalho foram empregadas apenas 57 sequências promotoras. Este pequeno número não abrange todas as características do universo de sequências promotoras disponíveis, que são de aproximadamente 740 para as sequências reconhecidas pelo fator σ^{70} de *E. coli*. Se todo o conjunto disponível fosse considerado, é possível que mais de 57 atributos fossem selecionados como caracterizadores das sequências e, provavelmente, os valores de desempenho diminuiriam.

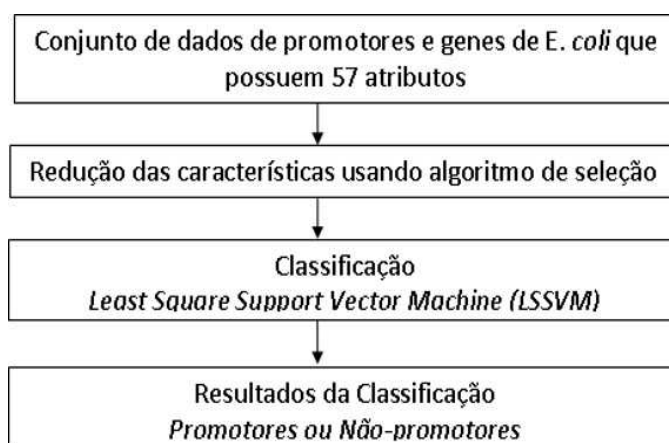


Figura 3.9: Fluxograma que ilustra a metodologia desenvolvida pelos autores Polat e Günes (2007).

Gordon *et al.* (2003) usaram uma SVM com núcleo de uma função de alinhamento. Neste trabalho, foram tomados dois conjuntos de dados: (i) promotores e regiões codificadoras e (ii) promotores e regiões intergênicas. A metodologia empregada por eles mostra uma média de erro de 16,5% e de 18,6%, respectivamente aos conjuntos de dados usados.

A SVM foi também utilizada para a predição *in silico* da TSS e seus promotores constitutivos associados em *E. coli* por Gordon *et al.* (2006). O método conseguiu uma acurácia de acordo com o estado da arte (erro médio de 11,6%). Mais tarde, o mesmo grupo de pesquisa (TOWSEY *et al.*, 2008), usou a SVM treinada anteriormente em outras sequências procarióticas (*B. subtilis* e *Chlamydia trachomatis*). Os valores de performance (acurácia, precisão, sensibilidade ou

especificidade) não são apresentados no trabalho. No entanto, os autores ressaltam que sua metodologia foi capaz de encontrar outras informações relevantes além dos motivos consensuais -10 e -35, sendo descrito um motivo localizado na região +15 ao +25.

Para verificar a existência de alguma correlação entre o grau de conservação da sequência promotora e a expressão do seu respectivo gene, Kiryu *et al.* (2005) utilizou as SVMs e, como resultado, estes autores não encontraram correlação entre a sequência promotora e o nível de expressão gênica.

3.3.2 Redes Neurais

As RNs são um sistema de AM inspirado no funcionamento de redes neurais biológicas. Pode-se afirmar que as RNs aprendem a partir dos exemplos e apresentam alguma capacidade de generalização do conjunto de treinamento (WU e MCLARTY, 2000).

As primeiras aplicações de RNs na predição de promotores, como apresentados nos trabalhos de Demeler e Zhou (1991) e O'Neill (1991), apesar da arquitetura simples, obtiveram uma alta acurácia, mas um número de falsos positivos igualmente alto. Outras abordagens foram apresentadas por Mahadevan e Ghosh (1994) que usaram uma combinação de duas RNs para a identificação de promotores de *E. coli*. Todos os promotores deste trabalho tinham espaçamento entre 15-21 nt entre os hexâmeros característicos. A primeira RN predizia os hexâmeros consensuais, enquanto a segunda foi designada para o reconhecimento da sequência inteira (65 nt), sendo o espaço entre os hexâmeros variável. Uma vez usada a informação da sequência inteira ocorreu dependências entre as bases em várias posições. Isto refletiu em um treinamento pobre e uma predição realizada por duas redes sem neurônios na camada oculta.

Pedersen e Engelbrecht (1995) predisseram a TSS e identificaram novos sinais característicos correlacionados com o local de início da transcrição. Para isso, foram usados dois diferentes esquemas de codificação, um com janelas 1 até 51 nt e outro com uma janela de 65 nt. Uma ideia interessante, neste trabalho, foi a medida do conteúdo de informação relativa dos dados de entrada, pelo uso da habilidade da RN para aprender corretamente, como avaliado pelo coeficiente de correlação do teste máximo.

Uma ferramenta disponível na internet e baseada em RNs é o *Neural Networks Promoter Prediction* (NNPP). Burden *et al.* (2005) incorporou à rede a informação sobre a distância entre o sítio de início de transcrição (TSS) e o sítio de início da tradução – TLS – (primeiro nucleotídeo da região codificadora). Com um conjunto de dados de 771 promotores, eles conseguiram uma precisão de 54% e uma sensibilidade de 86%.

Askary *et al.* (2009) descrevem uma arquitetura de RN chamada de N4, capaz de prever a TSS de promotores de *E. coli* reconhecidos pelo fator σ^{70} . A sensibilidade e a precisão da rede foram superiores a 94%. Esta rede neural recebeu os valores de estabilidade das sequências convertidos em codificação ortogonal (Figura 3.10). Assim, a camada de entrada possuiu 6608 neurônios (413 grupos x 16 combinações de valores de estabilidade). Esta rede possuiu duas camadas de neurônios ocultos, totalizando 402 neurônios e um neurônio na camada de saída. Apesar da complexidade da arquitetura apresentada (o que torna a rede computacionalmente pesada), este trabalho mostra o potencial de utilização dos valores de estabilidade das sequências como parâmetro de classificação.

tt	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
tc	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
ta	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	
tg	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	
ct	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	
cc	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	
ca	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	
cg	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	
at	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	
ac	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	
aa	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	
ag	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
gt	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	
gc	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	
ga	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
gg	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	
	-1.02	-1.46	-0.60	-1.38	-1.16	-1.77	-1.38	-2.09	-0.73	-1.43	-1.02	-1.16	-1.43	-2.28	-1.46	-1.77
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓

Figura 3.10: Codificação ortogonal para os valores de estabilidade, empregado por Askary *et al.* (2009).

Utilizando a informação da quantidade de dinucleotídeos da sequência promotora, Rani *et al.* (2006) treinaram uma RN. Neste trabalho foram utilizados promotores σ^{70} de *E. coli* como exemplos positivos e quatro conjuntos diferentes de exemplos negativos: (i) sequências codificantes, (ii) sequências codificantes e intergênicas, (iii) sequências aleatórias com 60% de AT e (iv) sequências aleatórias com 50% de AT. Os resultados de especificidade e sensibilidade obtidos estão

apresentados na tabela 3.4.

Tabela 3.4: Resultados obtidos pelo trabalho de Rani et al. (2006)

Conjunto de dados	Sensibilidade	Especificidade
Promotores + sequências codificantes	80%	79%
Promotores + sequências codificantes e intergênicas	63%	88%
Promotores + sequências aleatórias com 60% de AT	93%	88%
Promotores + sequências aleatórias com 50% de AT	95%	99%

Estes valores podem ser explicados pela quantidade grande de dinucleotídeos AT nos promotores reconhecidos pelo σ^{70} , conforme os resultados mostrados no capítulo II dos resultados desta tese. Quando esta metodologia for aplicada em sequências reconhecidas por outros fatores σ , que possuem conteúdo AT mais baixo, estes mesmos valores podem não ser alcançados.

A vantagem das RNs em relação a outras técnicas de AM é que elas podem aprender a reconhecer padrões degenerados, imprecisos e incompletos, os quais são característicos dos promotores. Além disso, permitem rápido desempenho em grandes sequências genômicas (COTIK *et al.*, 2005). Como desvantagem, pode-se citar a subjetividade da escolha dos parâmetros e da arquitetura da rede, uma vez que há falta de recomendações teóricas sobre estes e também sobre o tamanho do conjunto de treinamento.

3.4 METODOLOGIA UTILIZANDO A VALORES DE ESTABILIDADE

Kanhere e Bansal (2005b) desenvolveram uma metodologia baseada nas diferenças de estabilidade (ΔG^0) entre as regiões promotoras e codificantes (Figura 3.11). Esta ferramenta foi modificada e melhorada por Rangannan e Bansal (2007). Eles calcularam a energia livre (estabilidade) entre duas regiões do genoma de um organismo, conforme as equações (3.2), (3.3), (3.4) e (3.5). Os resultados obtidos por eles mostram que a estabilidade é uma medida melhor que os motivos conservados para diferenciar regiões promotoras e não-promotoras.

$$D(n) = E1(n) - E2(n) \quad (3.2)$$

onde,

$$E1(n) = \frac{\sum_{n}^{n+99} \Delta G^{\circ}}{100} \quad (3.3)$$

$$E2(n) = \frac{\sum_{n+99}^{n+119} \Delta G^{\circ}}{100} \quad (3.4)$$

$$E1'(n) = \frac{\sum_{n}^{n+49} \Delta G^{\circ}}{50} \quad (3.5)$$

onde, n é o nucleotídeo da sequência promotora.

Assim, $E1(n)$ e $E2(n)$ representam a média de energia livre em uma janela de 100 nt começando de n com uma vizinhança de 100 nt. $E1'(n)$ representa a média de energia livre em uma região de 50 nt. $E1'$ é usado no lugar de $E1$ no ciclo de refinamento dos falsos negativos. O valor de D representa a diferença de energia livre em duas regiões vizinhas. Uma sequência de DNA é designada como portadora de um promotor somente se a média da energia livre da região de 100 nt ($E1$) e a diferença (D) na energia livre forem maiores que o $E\text{-cutoff}$ e $D\text{-cutoff}$ escolhido, respectivamente. A metodologia desenvolvida pelos autores para a análise e predição de promotores está esquematizada na Figura 3.11.

Esta metodologia consegue uma sensibilidade de 98%, mas uma precisão de 55% (número de verdadeiros positivos/(número de verdadeiros positivos + número de falsos positivos)). Uma desvantagem desta metodologia é que ela se aplica somente em grandes sequências (do nucleotídeo -150 até o nucleotídeo +50) e foram analisadas 251 sequências, o que representa aproximadamente um terço do total das sequências disponíveis para σ^{70} da *E.coli*.

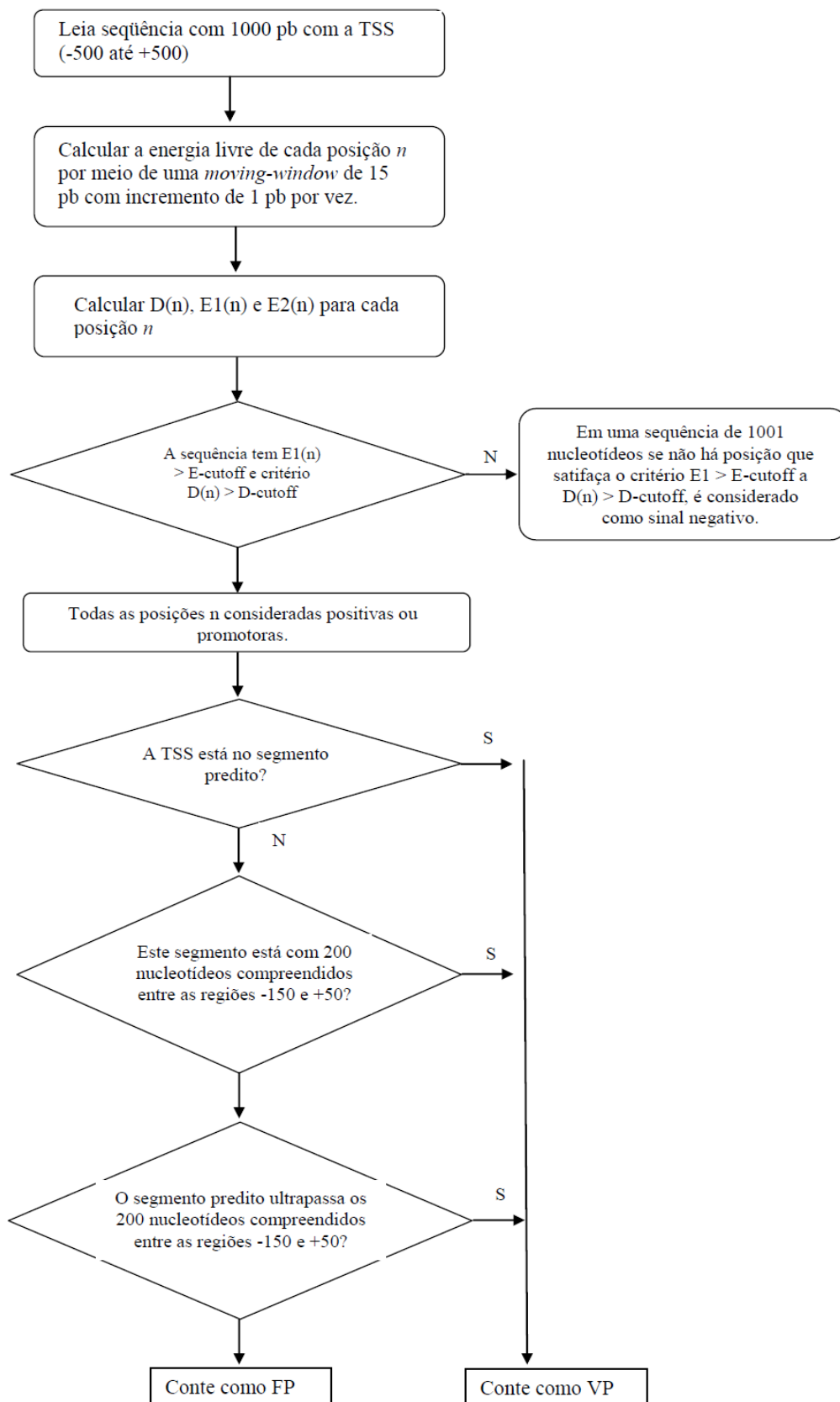


Figura 3.11: Fluxograma da metodologia descrita por Rangannan e Bansal (2007).

3.5 FUNDAMENTOS DE REDES NEURAIS ARTIFICIAIS

Conforme Baldi e Brunak (2001), as RNs foram originalmente desenvolvidas com o objetivo de modelar o processamento de informação e aprendizagem do cérebro. Trata-se de um modelo computacional aplicável a uma ampla variedade de áreas, como Engenharia, Economia e Biologia. Nesta última, principalmente em problemas de análise de sequências e reconhecimento de padrões. Nas demais áreas, por exemplo, as RNs podem ser aplicadas na síntese e reconhecimento de fala, interface adaptativa entre humanos e sistemas físicos complexos, aproximação de funções, entre outros.

Esta seção apresenta conceitos fundamentais sobre as RNs que auxiliam na compreensão da metodologia empregada e possibilitam uma melhor discussão dos resultados obtidos.

3.5.1 Arquitetura das Redes Neurais

As RNs, conforme Wu e McLarty (2000), consistem de grupos ou camadas (*layers*) de unidades de processamento com (ou algumas vezes sem) conexões entre os grupos. A unidade básica de uma camada é um neurônio artificial. Estas unidades, como os neurônios reais, têm conexões de entrada (dendritos) e conexões de saída (axônios). Também como neurônios reais, as unidades da rede neural também têm alguma forma de processamento interno, que cria um sinal de saída como uma função do sinal de entrada. Entretanto, diferentemente dos neurônios reais, o neurônio artificial tem como saída um número e apresenta mudanças somente em um intervalo discreto de tempo, conforme é ilustrado na Figura 3.12.

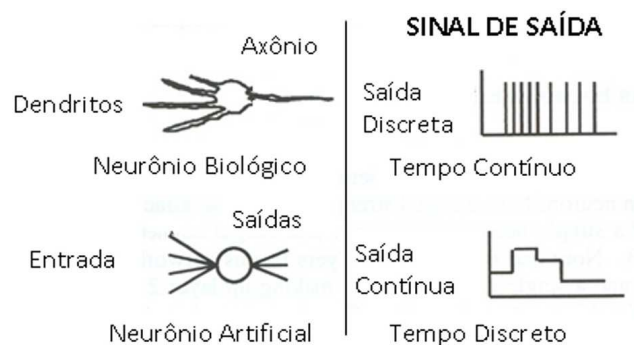


Figura 3.12: Analogia entre neurônios biológicos e artificiais. (WU e MCLARTY, 2000).

Uma RN é caracterizada pelo (i) padrão de conexões entre os neurônios (chamado de arquitetura), (ii) método de determinação de pesos nas conexões (chamado de treinamento ou aprendizagem) e (iii) sua função de ativação. Esses parâmetros estão descritos ao longo desta e das próximas seções.

Os neurônios (Figura 3.13) são conectados por vínculos orientados. Assim, pode-se representar uma RN como um grafo direcionado com peso ou arquitetura (MOUNT, 2000). Um vínculo da unidade j para a unidade i serve para propagar a ativação a_j desde j até i . Cada vínculo também tem um peso numérico W_{ji} associado a ele, o qual determina a intensidade e o sinal da conexão. Especificamente, um sinal a_j na entrada da sinapse i conectada ao neurônio j é multiplicada pelo peso sináptico W_{ji} (HAYKIN, 1999; RUSSEL e NORVIG, 2003).

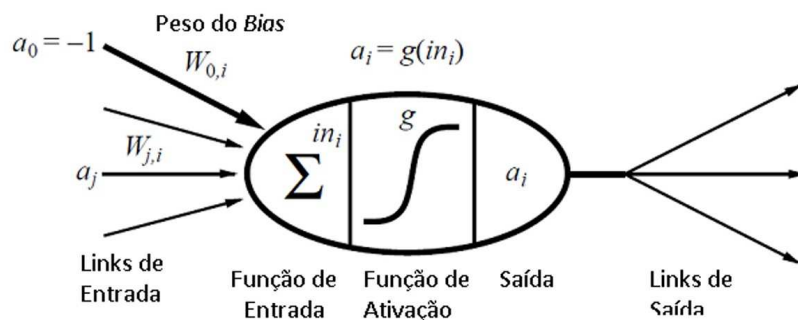


Figura 3.13: Modelo de um neurônio artificial (RUSSELL e NORVIG, 2003).

A ativação da saída da unidade é $a_i = g(\sum_{j=0}^n W_{ji} a_j)$

onde a_j é ativação de saída da unidade j e W_{ji} é o peso no vínculo da unidade j até essa unidade.

Após, cada unidade i calcula inicialmente uma soma ponderada de suas entradas:

$$in_i = \sum_{j=0}^n W_{ji} a_j \quad (3.6)$$

Então ela aplica uma função de ativação g a essa soma para derivar a saída:

$$a_i = g(in_i) = g\left(\sum_{j=0}^n W_{ji} a_j\right) \quad (3.7)$$

É importante ressaltar que há a inclusão de parâmetro externo do neurônio

artificial, um *bias* W_{0i} (Figura 3.12) conectado a uma entrada fixa $a_0 = -1$. O termo W_{0i} define o limite real para a unidade, no sentido de que a unidade é ativada quando a soma ponderada de entradas “reais” $\sum_{j=1}^n W_{ji} a_j$ excede W_{0i} . A função de ativação g é projetada para atender duas aspirações: primeiro, a unidade de estar “ativa” (próxima de +1) quando as entradas positivas forem recebidas e negativas (próxima a 0) quando as entradas “erradas” forem recebidas. Em segundo lugar, a ativação precisa ser não-linear, caso contrário a RN inteira entrará em colapso, tornando-se uma função linear simples (HAYKIN, 1999; RUSSEL e NORVIG, 2003).

Uma arquitetura com melhor capacidade de generalização constitui-se de redes com múltiplas camadas, chamadas de Multilayer Perceptron (MLP), sendo o caso mais comum aquelas que envolvem uma única camada oculta, conforme ilustrado na Figura 3.14. Segundo Hornik (1989), as RNs com uma única camada oculta são aproximadores universais, pois aproximam qualquer função com precisão arbitrária. A função dos neurônios ocultos é intervir entre a entrada externa e a saída de maneira útil. A vantagem de adicionar camadas ocultas é que ela aumenta o espaço de hipóteses que a rede pode representar e, assim, é capaz de extrair estatísticas de ordem elevada. Isto é particularmente valioso quando o tamanho da camada de entrada é grande.

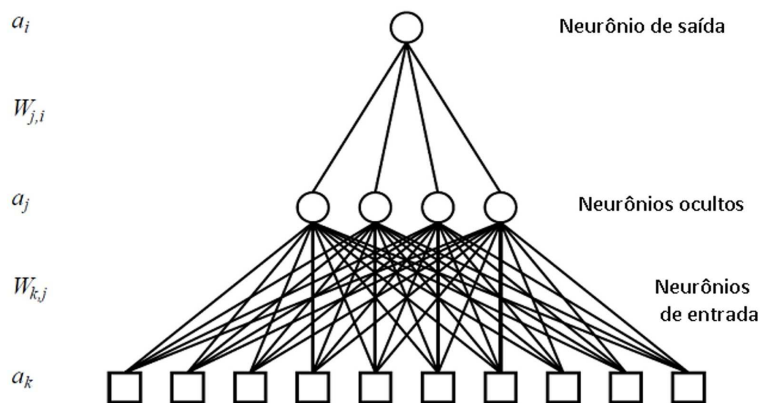


Figura 3.14: Rede MLP com três camadas (RUSSELL e NORVIG, 2003).

Redes com muitas camadas ocultas são menos eficientes pois requerem maior tempo de computação e apresentam menor capacidade de generalização quando comparadas às redes com uma camada oculta. Além disso, a extração das regras da rede se torna mais difícil.

3.5.2 Treinamento de Redes Neurais

Segundo Wu e McLarty (2000) a ideia fundamental do aprendizado ou treinamento, para todas as arquiteturas de RN, é atribuir valores a um conjunto de pesos (inicializado normalmente de forma aleatória), aplicar os dados de entrada à rede e verificar como esta responde a determinados conjuntos de pesos. Se o desempenho não for satisfatório, então os pesos devem ser modificados pelo algoritmo específico da arquitetura e repetir o procedimento. Este procedimento deve ser repetido até que algum critério de parada pré-especificado seja atingido.

A passagem de todos os vetores dos dados de entrada através da rede é chamado de época. Alterações nos pesos podem ser feitas a cada padrão processado (treinamento *on-line*) ou após uma época inteira (treinamento em lote), sendo esta última o procedimento mais utilizado. O objetivo do treinamento é encontrar o conjunto de parâmetros (número de camadas, número de neurônios nas camadas e pesos entre as camadas) que minimize a diferença entre os valores de saída da rede e os valores desejados. No entanto, se a rede tiver uma arquitetura com muitas camadas ocultas ou for treinada por muitas épocas (a quantidade de épocas neste caso varia de acordo com os dados a rede envolvida), ela será capaz de memorizar todos os exemplos. Isto é chamado de *overtraining*, já que a rede forma uma extensa tabela de busca mas não realiza boas generalizações para entradas que não foram vistas antes.

Uma das maneiras de testar a exatidão da rede é tentar várias arquiteturas e, com a técnica de validação cruzada, verificar qual apresenta os melhores resultados. A técnica de validação cruzada, ou *k-fold-cross-validation (k-FCV)*, consiste em particionar aleatoriamente o arquivo de padrões em k partes de mesmo tamanho. Assim, ocorre a geração dos arquivos para treinamento e validação. As etapas de treinamento e validação são repetidas k vezes, sendo utilizados para treinamento $k-1$ arquivos e para validação o k -ésimo arquivo não utilizado no treinamento. A cada iteração, o arquivo de validação possui um k diferente.

Outros métodos de validação que podem ser citados são: *holdout* e *jackknife*. O método *holdout* consiste em separar, de forma aleatória, o arquivo de padrões em dois arquivos. O de treinamento tipicamente conterá dois terços dos dados e o de validação o um terço restante. Já o método *jackknife*, conhecido com *leave-one-out*, é semelhante ao k -FCV, mas k é igual ao número de linhas do arquivo de padrões.

Com isto, cada arquivo de validação conterá somente uma linha em cada etapa do processo.

O procedimento utilizado para realizar o processo de aprendizagem é chamado algoritmo de aprendizagem, e sua função é modificar os pesos sinápticos de forma a alcançar o objetivo desejado. Os algoritmos de aprendizado podem ser supervisionados ou não-supervisionados, embora aspectos de cada um possam co-existir em uma dada arquitetura. O treinamento supervisionado é acompanhado pela apresentação de uma sequência no vetor de treinamento associada com um vetor de saída alvo. Um ingrediente essencial neste tipo de aprendizado é a disponibilidade de um “professor” externo. Em termos conceituais, podemos pensar que o professor tem o conhecimento da saída desejada. O conhecimento disponível pelo professor é então transferido à RN através de ajustes iterativos para minimizar o erro de acordo com o algoritmo de aprendizado (WU, 1997). Como exemplo, os algoritmos: *Back-propagation* (BP), *Resilient Proapagation*, *Cascade Correlation*, *Kohonen* e *Quickprop*. A principal diferença entre eles está, principalmente, no modo como os pesos da rede são ajustados.

Um algoritmo de aprendizado supervisionado tem o objetivo de minimizar a diferença entre o valor de saída da rede e o valor desejado. Uma típica função de erro a ser minimizada é:

$$E = \sum_{i=1}^n (y_i - h_w(x))^2 \quad (3.8)$$

onde n é o número de padrões de entrada, y_i é a saída da rede (para um dado conjunto de parâmetros w) e $h_w(x)$ o valores esperado de saída. Se uma rede possui mais que uma unidade na camada de saída, então a equação 3.8 se torna:

$$E = \sum_{i=1}^n \sum_{j=1}^k (y_i - h_w(x))^2 \quad (3.9)$$

onde k é o número de unidades na camada de saída (WU e McLARTY, 2000).

O treinamento não-supervisionado ou aprendizado auto-organizável não possui um professor externo para verificar o processo de aprendizado. O algoritmo normalmente é guiado pela medida de similaridade sem um vetor alvo de

especificação. As redes auto-organizáveis modificam os pesos até que os vetores mais similares sejam designados ao mesmo grupo de saída (clusterização), o qual é representado por um vetor-exemplo. Como exemplo de algoritmo de aprendizado não-supervisionado, pode-se ser citados os mapas auto-organizáveis de *Kohonen* e a teoria da ressonância adaptativa (ART) (WU, 1997).

Rumelhart *et al.* (1986) criaram um método intuitivo que aprende rapidamente, revolucionando o campo das RNs. O método foi chamado de BP porque o erro é propagado da saída para a entrada da rede, ou seja, a propagação do erro pode ser efetuada da camada de saída para a camada oculta e desta para a camada de entrada. O erro nas camadas ocultas parece misterioso, porque os dados de treinamento não informam que valor os neurônios ocultos devem ter. O processo de propagação de retorno emerge diretamente de uma derivação do gradiente de erro global e da aplicação da regra da cadeia (WU e McLARTY, 2000; RUSSELL e NORVIG, 2003).

Uma RN multicamadas tem três características distintas:

1. O modelo de cada neurônio da rede inclui uma função de ativação não-linear, como a função logística.

2. A rede contém uma ou mais camadas de neurônios ocultos, que não são parte da entrada ou da saída da rede. Estes neurônios capacitam a rede a aprender tarefas complexas extraindo progressivamente as características mais significativas dos vetores de entrada.

3. A rede exibe um alto grau de conectividade, determinado pelas sinapses da rede.

Estas características conferem o poder computacional da MLP, mas também são responsáveis pelas deficiências na compreensão do comportamento da rede (HAYKIN, 1999).

3.6 EXTRAÇÃO DE REGRAS

A metodologia de RNs possui uma grande aplicabilidade nos mais diversos problemas, mas uma de suas desvantagens é que o conhecimento adquirido por elas não é diretamente acessível. Com objetivo de diminuir esta dificuldade, muitos algoritmos para extração de regras a partir das RNs treinadas têm sido

desenvolvidos. Assim, as redes tornam-se mais atrativas que outras técnicas de AM já que fornecem uma explicação de como cada decisão é feita (ODAJIMA *et al.*, 2007).

3.6.1 Extração de Regras a Partir de Redes Neurais

Uma das características mais atrativas das RNs é que elas não requerem um conhecimento prévio da aplicação do problema para a construção do modelo. Assim, para tornar esta metodologia realmente compreensível ao usuário, é desejável extrair conhecimento a partir de redes treinadas (WU e MCLARTY, 2000). Muitas vezes, as RNs são denominadas de “caixa preta”, em particular por não fornecer ao usuário nenhuma informação sobre o conhecimento adquirido. Embora isto geralmente seja verdadeiro, especialmente para redes multicamadas, existem métodos para analisar RNs e extrair regras ou características. Estas regras incluem: regras de inferência (*if-then-else*), árvores de decisão, regras difusas, entre outras.

Conforme Andrews *et al.* (1995), a extração de regras pode oferecer alguns benefícios listados a seguir:

- descoberta de novos relacionamentos e/ou características importantes a partir das regras extraídas;
- expressão do conhecimento de modo formal;
- capacidade de gerar explicações para as decisões tomadas internamente pela RN, de modo que facilite a aceitação do uso da rede pelos usuários;
- integração com sistemas simbólicos e a possibilidade de descobrir em que situações a rede pode cometer erros de generalização;
- identificação de regiões no espaço de entrada que não se fizeram representar no conjunto de treinamento.

Além disso, as regras extraídas a partir das RNs podem ser apresentadas para análise de um especialista. Assim, as regras corretas podem ser usadas para gerar padrões de treinamento adequados, os quais podem melhorar a capacidade de generalização (CLOETE e ZURADA, 2000). Uma vez que este trabalho visa extrair regras a partir de RNs, a próxima seção descreve brevemente alguns tipos de regras, com ênfase maior às regras do tipo *if-then*.

3.6.2 Tipos de regras

A extração de regras a partir de RNs é baseada no comportamento dos neurônios, sendo a relação entre as entradas e as saídas usualmente analisada (CLOETE e ZURADA, 2000; HUANG e XING, 2005). Conforme Andrews *et al.* (1995), há muitos tipos de regras que podem ser extraídos das RNs, mas o desenvolvimento de técnicas de extração de regras tem sido mais direcionado à apresentação da saída como um conjunto de regras expressas, usando a forma convencional de lógica simbólica na forma *if...then...else...*

Neste tipo de regra, a parte SE especifica um conjunto de condições sobre valores de atributos previsoires e a parte ENTÃO especifica um valor previsto para o atributo de saída. Os atributos previsoires são as premissas da regra que devem ser obedecidas, para assim obter um atributo classe.

IF < condição> *THEN* <conclusão> (<confiança>)

A “condição” é, tipicamente, uma expressão lógica que contém variáveis relevantes das quais os valores podem ser inferidos a partir das bases de fatos ou fornecidos pelo usuário. A “conclusão” determina o valor de alguma variável que corresponde a “condição” ser satisfeita. O grau de certeza ou validade da regra é expressa pelo seu percentual de confiança (CLOETE e ZURADA, 2000). A extração de regras é realizada através da interpretação dos pesos da rede neural.

As regras do tipo *if-then* podem ser utilizadas posteriormente em um sistema de inferência lógica para a resolução de problemas. Um segundo uso destas regras pode ser a geração de regras para um sistema baseado em conhecimento. Deve-se observar, também, que quanto mais curtas as regras (em termos de números de cláusulas) melhor, pois regras curtas geralmente podem ser aplicadas a mais situações (CLOETE e ZURADA, 2000).

3.6.3 Regras obtidas a partir dos neurônios da camada oculta

Para a obtenção de regras a partir dos neurônios da camada oculta da RN treinada, o programa denominado FAGNIS (CECHIN, 1998), analisa o valor de ativação dos neurônios na camada oculta e os classifica em três regiões, conforme ilustrado na Figura 3.15. Aqui, encontra-se uma breve explicação desta ferramenta. Uma descrição mais detalhada pode ser encontrada no capítulo II dos resultados desta tese.

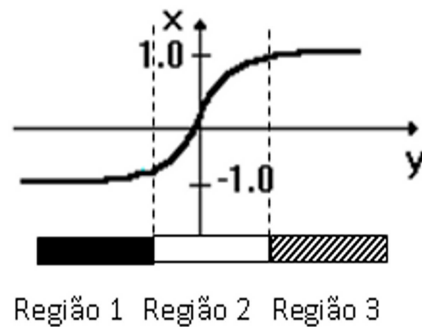


Figura 3.15: Ilustração das três regiões definidas na função sigmóide para análise dos dados de entrada e extração de regras.

A ferramenta FAGNIS, verifica em qual das regiões a ativação dos neurônios ocultos se enquadram para cada entrada da rede, O número máximo de combinações possíveis é $3n$, onde n simboliza o número de neurônios na camada oculta. No entanto, nem todas estas combinações ocorrem e, somente as combinações mais frequentes são consideradas, pois melhor representam os dados. Como resultado, temos o protótipo da regra, o qual é definido como a média das entradas de cada grupo (combinação das regiões). Assim, a escrita formal da regra possui a forma de uma equação linear: “SE $X \approx$ protótipo ENTÃO $Y =$ constante da equação linear + (os coeficientes da equação linear) * X .” Aqui, X é o exemplo de entrada; Y corresponde à saída da RN e os coeficientes da equação linear representam a influência dos exemplos na saída da RN.

3.7 CONSIDERAÇÕES ADICIONAIS

Os promotores são importantes reguladores da expressão gênica e, a revisão bibliográfica sobre o estado da arte mostrou os esforços realizados para melhorar a acurácia da predição e a importância de estender o estudo para outras espécies bacterianas além de *E. coli*. Nesta seção não há descrição de trabalhos relacionados com a extração de regras a partir de RNs aplicadas à predição de promotores, pois não foi encontrado nenhum artigo até o término da revisão bibliográfica. Este trabalho pretende extrair regras de inferência das RNs treinadas para compreender o processo de classificação e, a partir das regras criar uma ferramenta própria para a predição de promotores de bactérias Gram-negativas.

4 METODOLOGIA

Nesta seção, uma visão geral da metodologia desenvolvida é descrita. Para facilitar a compreensão dos procedimentos realizados, apresenta-se um fluxograma (Figura 4.1) com todas as etapas da metodologia. A descrição das etapas apresentadas na Figura 4.1 são descritas nos capítulos I à IV dos resultados.

4.1 ORGANISMOS ESTUDADOS

Os organismos escolhidos foram as bactérias *E. coli*, as do gênero *Shigella*, *Pseudomonas*, *Salmonella* e *Aeromonas*. Assim, abrange-se uma ampla variedade de representantes das bactérias Gram-negativas. Salienta-se que neste estudo, as bactérias Gram-positivas não foram consideradas por apresentarem diferentes características em relação às Gram-negativas, no que diz respeito à composição química, estrutura e permeabilidade da parede celular, além de diferenças fisiológicas, de metabolismo e patogenicidade.

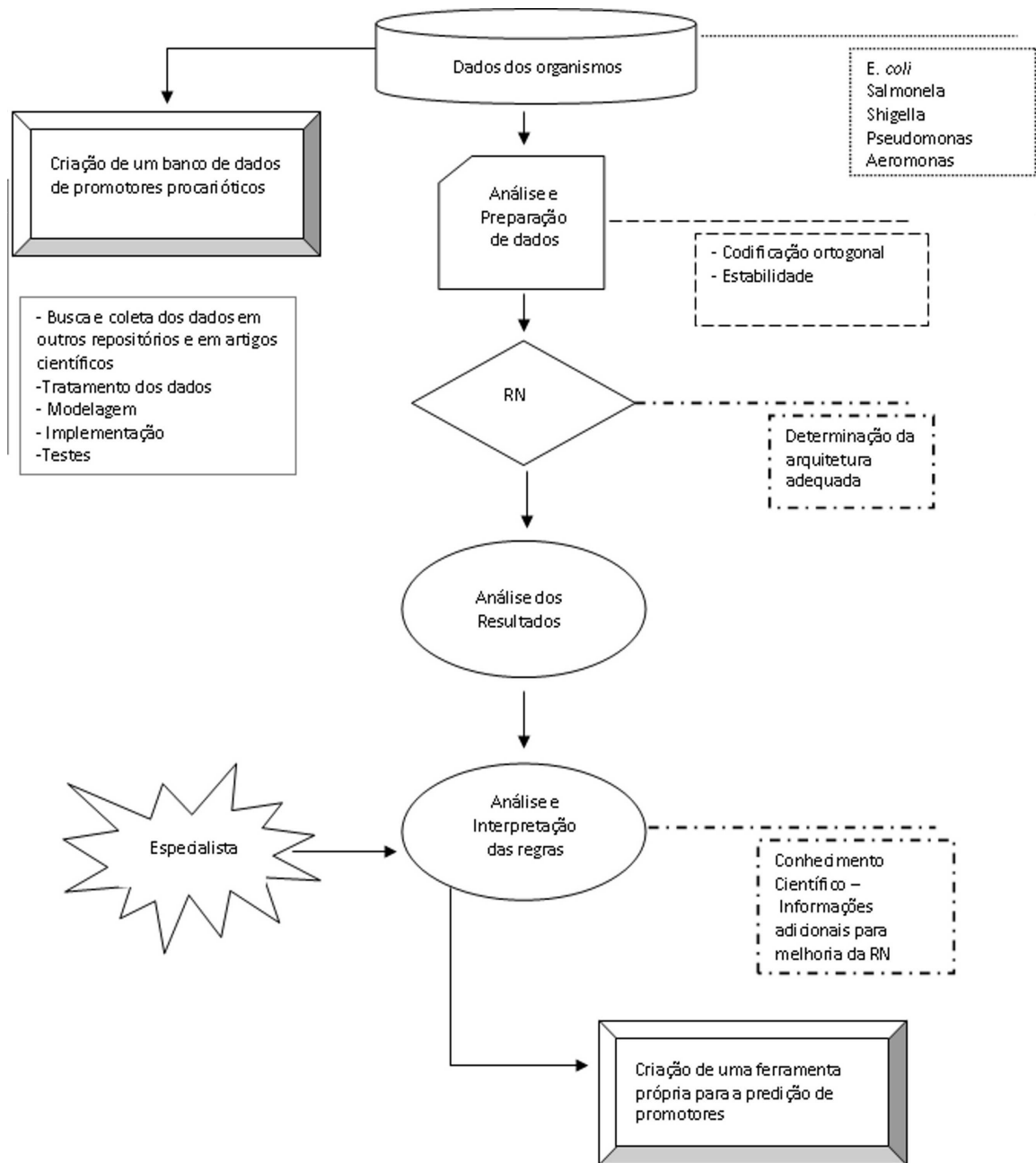


Figura 4.1: Estrutura da metodologia proposta para o uso de RN no reconhecimento e predição de promotores.

4.2 BANCOS DE DADOS

As regiões promotoras, as regiões intergênicas e os dados relacionados às características dos promotores foram retirados de bancos de dados biológicos e de artigos científicos. Os bancos de dados públicos utilizados foram:

- **CMR**: banco de dados de genomas procarióticos (PETERSON *et al.*, 2001). Nele se encontram dados de genomas completos, de regiões específicas (genes, promotores, regiões intergênicas, homologias), entre outras ferramentas. As informações estão disponíveis no endereço de internet: <http://cmr.jcvi.org/cgi-bin/CMR/CmrHomePage.cgi>

- **NCBI**: maior base de dados pública de sequências genéticas. Desta podem-se extrair sequências de genes, proteínas, genomas completos, dados de homologia e expressão gênica, além de possuir informações sobre os artigos relacionados a cada descoberta genética (WHEELER *et al.*, 2008). As informações estão disponíveis no endereço de internet: <http://www.ncbi.nlm.nih.gov>.

- **RegulonDB**: base de dados que contém informações acuradas sobre a rede regulatória de *E. coli* com conhecimento experimental. Há dados sobre a organização de operons, promotores e seu fator sigma associado, entre outros (GAMA-CASTRO *et al.*, 2008). As informações estão disponíveis no endereço de internet: <http://regulondb.ccg.unam.mx/index.html>.

4.3 FERRAMENTAS

As principais ferramentas computacionais utilizadas foram:

- **Python**: linguagem de programação escolhida para desenvolver programas que automatizem a preparação de dados para as etapas de treinamento e teste (PYTHON SOFTWARE FOUNDATION, 2009).

- **R**: software para manipulação e análise de dados. Este software permite a realização de análise estatística, treinamento de RNs, extração de regras, entre outras funções (R DEVELOPMENT CORE TEAM, 2005).

- **SPSS**: software para análise estatística. Permite a realização de gráficos e outras funções (SPSS).

- **Tisean**: software de domínio público que realiza a suavização de dados através de um filtro passa-baixa – *LowPass* – (HEGGER *et al.*, 1999).

- **WEBLOGO**: aplicação da web (de uso livre) para a geração de sequencias logo. Estas são uma representação gráfica de aminoácidos ou ácidos nucleicos de múltiplos alinhamentos. Cada logo consiste em empilhamento dos símbolos, um para cada posição da sequência. O tamanho geral dos empilhamentos indica o grau

de conservação da sequência em determinada posição, enquanto que o tamanho do símbolo indica a frequência relativa do aminoácido ou nucleotídeo em cada posição (CROOKS *et al.*, 2004).

4.4 CRIAÇÃO DE BANCO DE DADOS DE REGIÕES INTERGÊNICAS

A criação do banco de dados se faz necessária uma vez que não existem repositórios públicos de dados sobre os promotores pertencentes a outras bactérias Gram-negativas que não *E. coli*. Assim, desta necessidade, implantamos uma base de dados com as regiões intergênicas. Esta base de dados arquiva as sequências intergênicas e outras informações associadas, como porcentagem de GC, genes associados, tamanho, localização na fita de DNA, entre outras informações. Esta base de dados foi desenvolvida em conjunto com professores e alunos do Centro de Computação e Tecnologia da Informação e está descrita no capítulo IV.

5 RESULTADOS

Esta seção apresenta os resultados obtidos na forma de artigos científicos, sendo organizada em cinco capítulos, nos quais são apresentados:

Capítulo I- *Rules extraction from neural networks applied to prediction and recognition of prokaryotic promoters.*

Capítulo II - *BacPP: Bacterial promoter prediction - A tool for accurate sigma-factor specific assignment in enterobacteria.*

Capítulo III- *Neural Networks applied to bacterial promoter prediction based on DNA stability.*

Capítulo IV- Banco de dados IntergenicDB.

5.1 CAPÍTULO I - RULES EXTRACTION FROM NEURAL NETWORKS APPLIED TO PREDICTION AND RECOGNITION OF PROKARYOTIC PROMOTERS

Este capítulo apresenta o artigo "*Rules extraction from neural networks applied to the prediction and recognition of prokaryotic promoters*", publicado na revista *Genetics and Molecular Biology*. Esta revista possui fator de impacto 0,08 (para os últimos 3 anos, conforme informação disponível no web site da revista [http://statbiblio.scielo.org//stat_biblio/index.php?state=19&lang=en&country=scl&iissn=1415-4757&CITED\[\]=GENETICS+AND+MOLECULAR+BIOLOGY&YNG\[\]=2011](http://statbiblio.scielo.org//stat_biblio/index.php?state=19&lang=en&country=scl&iissn=1415-4757&CITED[]=GENETICS+AND+MOLECULAR+BIOLOGY&YNG[]=2011)) e é classificada pela CAPES como B1 na área de avaliação Interdisciplinar. O trabalho pode ser acessado *on-line* pelo doi dx.doi.org/10.1590/S1415-47572011000200031. Este artigo descreve os resultados das simulações de RNs utilizando a codificação ortogonal e os valores de estabilidade e mostra as regras extraídas de cada arquitetura.



Rules extraction from neural networks applied to the prediction and recognition of prokaryotic promoters

Scheila de Avila e Silva, Günther J.L. Gerhardt and Sergio Echeverrigaray

Programa de Pós-Graduação em Biotecnologia, Universidade de Caxias do Sul, Caxias do Sul, RS, Brazil.

Abstract

Promoters are DNA sequences located upstream of the gene region and play a central role in gene expression. Computational techniques show good accuracy in gene prediction but are less successful in predicting promoters, primarily because of the high number of false positives that reflect characteristics of the promoter sequences. Many machine learning methods have been used to address this issue. Neural Networks (NN) have been successfully used in this field because of their ability to recognize imprecise and incomplete patterns characteristic of promoter sequences. In this paper, NN was used to predict and recognize promoter sequences in two data sets: (i) one based on nucleotide sequence information and (ii) another based on stability sequence information. The accuracy was approximately 80% for simulation (i) and 68% for simulation (ii). In the rules extracted, biological consensus motifs were important parts of the NN learning process in both simulations.

Key words: neural network, promoter, rule extraction.

Received: March 26, 2010; Accepted: January 11, 2011.

Introduction

The determination of how and when genes are turned on and off is a challenge in the post-genomic era. Differences between two species are often more related to gene expression and regulation than to their structures (Howard and Benson, 2002). An adequate comprehension of the complex metabolic networks present in various organisms, including cellular differentiation and cellular responses to environmental change, can be facilitated by studying of promoter sequences, *i.e.*, short sequences located before the transcription start site (TSS) of a gene (Jáuregui *et al.*, 2003; Pandey and Krishnamachari, 2006).

Promoters act as gene expression regulators through their ability to interact with the enzyme RNA polymerase, thereby initiating transcription. The σ factor moiety of the RNA polymerase, of which there are several types, are involved in the recognition and primary interaction with the promoters. Various bacterial σ factors interact with different promoter sequences that are characterized by particular consensus motifs and properties. Most prokaryotic promoters have two consensus hexameric (six nucleotides) motifs: one centered at position -35 and another centered at position -10 relative to the TSS. For factor σ^{70} , the pattern sequences for these motifs are 'TTGACA' and TATAAT' for

positions -35 and -10, respectively, and are separated by ~17 non-conserved nucleotides (Lewin, 2008).

As an analogy, the downstream sequences (genes) represent the "computer memory" while the upstream sequences (promoters) represent the "computer program" that acts on this memory. The study of promoters can provide new models for developing computer programs and for explaining how they operate (Howard and Benson, 2002). Despite the importance of promoters in gene expression, the shortness of their sequences, many of which are not highly conserved, makes them difficult to detect when compared to genes sequences. This characteristic limits the accuracy of *in silico* methods because many nucleotide alterations may not be significant in terms of promoter functionality (Howard and Benson, 2002; Burden *et al.*, 2005; Kanhere and Bansal, 2005b).

There are many machine learning approaches for promoter recognition and prediction, including Hidden Markov Models – HMM (Pedersen *et al.*, 1996), Support Vector Machines – SVM (Gordon *et al.*, 2003) and Neural Networks – NN. The earliest NN used for promoter prediction had a simple architecture (Demeler and Zhou, 1991; O'Neill, 1991). In these papers, the prediction had good accuracy but the number of false positives was high. Mahadevan and Ghosh (1994) used two NN: one to predict motifs and another to recognize the complete sequence. The *Neural Networks Promoter Prediction* (NNPP) program was implemented by Oppon (2000) and improved by Burden *et al.* (2005), who included information about the

distance between TSS and the first nucleotide translated, thereby decreasing the number of false positives.

Apart from consensus motifs, promoters have certain physical features, such as stability, curvature and bendability, that make them different from gene sequences, *i.e.*, they are less stable, more curved and more bendable (Kanhere and Bansal, 2005a). The latter authors subsequently used promoter stability information to develop a procedure that recognizes promoters in whole sequences (Kanhere and Bansal, 2005b). However, despite the importance of these physical features, they have not been widely used in NN promoter prediction.

Neural networks are suitable for promoter prediction and recognition because of their ability to identify degenerated, imprecise and incomplete patterns present in these sequences. In addition, NNs perform well when processing large genome sequences (Kalate *et al.*, 2003; Cotik *et al.*, 2005). A further feature is that there is no need for prior knowledge when building a suitable model. An important procedure in NN methods is rule extraction from trained networks that can assist the user in identifying biological rules from the input data (Andrews *et al.*, 1995). In this paper, we describe the use of a NN to predict and recognize prokaryotic promoters by comparing two data sets: (i) nucleotide sequence information and (ii) stability sequence information of *E. coli* promoters, regardless of the σ factor that recognizes the sequence.

Material and Methods

The promoter sequences used were obtained from the January 2006 version of the RegulonDB database (Gama-Castro *et al.*, 2008). Nine hundred and forty promoters and 940 random sequences were used to train and test the NN. The promoters and sequences represented positive and negative examples, respectively. The random sequences were generated with a probability of 0.22 for guanine (G) or cytosine (C) nucleotides and 0.28 for adenine (A) or thymine (T) nucleotides, based on the distribution of these nucleotides in real promoter sequences (Kanhere and Bansal, 2005a). The examples were shuffled and allocated to one of ten files in order to generate the train and test set. Two simulations were done, one based on nucleotide sequences and the other on stability information. The procedures are described below.

Simulation based on nucleotide sequences

In the simulation using nucleotide sequences (referred to as the sequence-based simulation) the promoters and random sequences were initially aligned with the software ClustalW (Thompson *et al.*, 1994) to accommodate the variable sequence length between the motifs. Without this initial alignment, the NN does not provide good accuracy. The alignment introduced gaps in the sequences, represented by a short line (-). The gaps were inserted where necessary (at the beginning, middle or end of a sequence) (Figure 1). The short line (-) was removed from the beginning and end of the sequence to avoid incorrect learning by the NN. Consequently, the resulting promoter sequences contained 72 nucleotides. After alignment, the nucleotides and gaps were encoded using a set of four binary digits as described by Demeler and Zhou (1991): A = 0100, T = 1000, C = 0001, G = 0010 and “-” = 0000.

The architecture used to classify the sequences had 288 input neurons (72 bp x four digits for each nucleotide), two neurons in the hidden layer and one neuron in the output layer (Figure 2a). The presence of a large number of neurons in the hidden layer or in the output layer did not increase the accuracy of the procedure.

Simulation using promoter sequence stability

The stability of DNA molecules can be expressed in terms of their free energy (ΔG), which in turn depends on the mononucleotide and dinucleotide composition (SantaLucia and Hicks, 2004). The stability of a DNA duplex can be predicted from its sequence based on the contribution of each nearest-neighbor interaction (SantaLucia and Hicks, 2004; Kanhere and Bansal, 2005a). The contribution of each dinucleotide is described in SantaLucia and Hicks (2004).

To do the simulation using the free energy information, denoted as the stability-based simulation, ΔG was calculated using the following formula, described in SantaLucia and Hicks (2004) and Kanhere and Bansal (2005a):

$$\Delta G^0 = \Delta G_{ij} \quad (1)$$

where ΔG^0_{ij} is the standard free energy change for dinucleotides of type *ij*. The original formula described in Kanhere and Bansal (2005a) was modified to adjust its adequacy to the goals of this paper. The best architecture ob-

-----CA TCTATCA TCTAAAAAACC--A GAAAAA CAAATAAC-A TCATGT-----	hycA
-----TTAAAAATCTCTTTAA TAACAATAAAT--TAAAA GTTGGCACA A-AAAATGC -----	rpsUp3
-----CGGTGCTTTACAAA GCA GCAGCAA TTGCAGTAAATTC CGCACCATTTTGA---	fusAp
-----GATAAA TCCA TGGCTCTGCGCCTGGCGAA CGAACTTTCTGATGCTGCA ---	gugpp2
-----TTCCCTCA CCCACGCCGTCACCGCCTTGTCATCTTTCTGACACCT -----	purH

Figure 1 - Examples of promoter sequences aligned by ClustalW software.

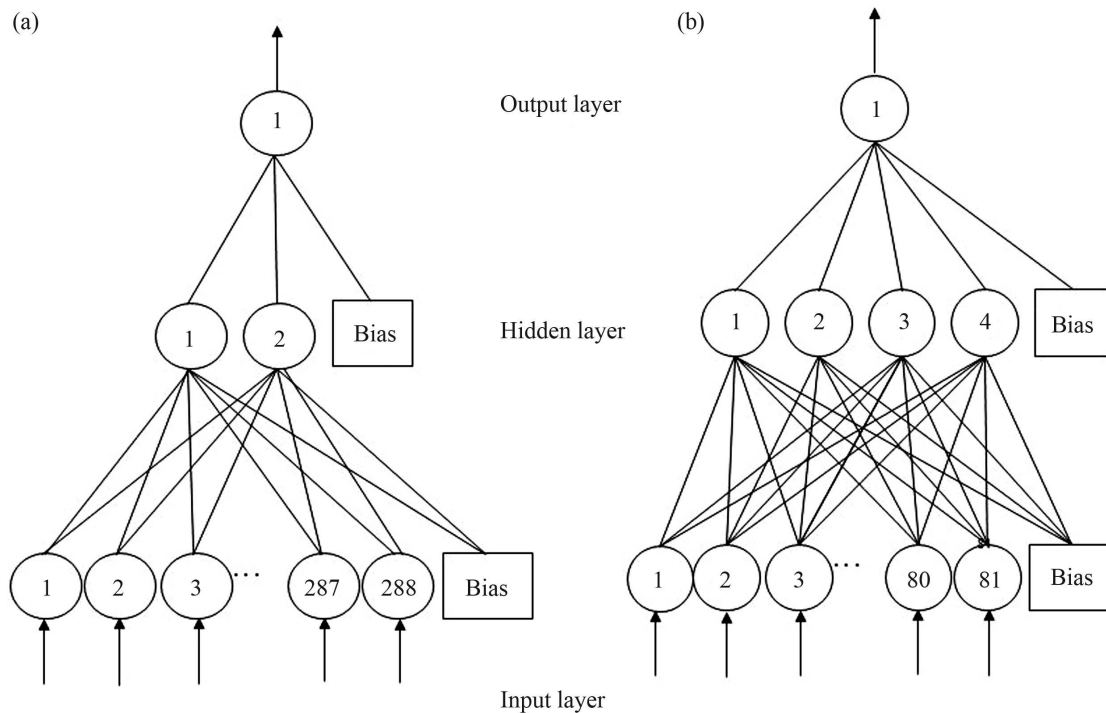


Figure 2 - (a) Architecture of the best NN used to classify promoter sequences in the sequence-based simulation. There were 288 neurons in the input layer, two neurons in the hidden layer and one neuron in the output layer. (b) Architecture of the best NN used to classify promoters in the stability-based simulation. There were 81 neurons in the input layer, four neurons in the hidden layer and one neuron in the output layer.

tained to classify the sequences had 81 neurons in the input layer, four hidden neurons and one output neuron (Figure 2b).

Training and analysis procedures

Both simulations were done in the R Environment (R Development Core Team, 2005). The algorithm *back-propagation* (BP) was chosen because it is the most popular algorithm for training feedforward networks (Kalate *et al.*, 2003). NNs based on the BP training algorithm have been successfully used for various applications in biology involving non-linear input-output modeling and classification (Mahadevan and Gosh, 1994; Kalate *et al.*, 2003; Burden *et al.*, 2005). The ten-fold cross-validation method was used to obtain statistically valid results. The k -fold cross-validation (k -FCV) technique consists in randomly sharing the examples' archive in k equal portions. The train and validation were repeated k times, using $k-1$ archives to train and k^{th} archives for validation. In each interaction, the validation archive had a different k (Polate and Günes, 2007).

The accuracy (A), specificity (S) and sensitivity (SN) were calculated from the number of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN). The TP were promoter sequences classified as promoters, TN were random sequences recognized as non-promoters, FP were random sequences classified as

promoters and FN, promoters classified as non-promoter sequences. The formulas used are given below:

$$A = \frac{TP + TN}{TN + TP + FN + FP} \quad (2)$$

$$S = \frac{TN}{TN + FP} \quad (3)$$

$$SN = \frac{TP}{TP + FN} \quad (4)$$

An input sequence was classified as a promoter if its output lay between 0.5 and 1.0. Otherwise, it was considered as a non-promoter (Kalate *et al.*, 2003).

Rule extraction

Neural networks are applicable to many different problems, but the learning process is complex (Andrews *et al.*, 1995). How a NN classifies a given sequence as promoter or non-promoter can be understood based on rule extraction. Here, we extracted rules using two approaches:

(1) *Rules based on hidden neurons*: The sigmoid function was divided into three regions (Figure 3). For each input, the region of the sigmoid function corresponding to the best fit of the activation function of the hidden neurons was identified. The maximum number of combinations was 3^n , where n is the number of neurons in the hidden layer. However, all of the possible combinations do not occur, and only the more frequent combinations were considered

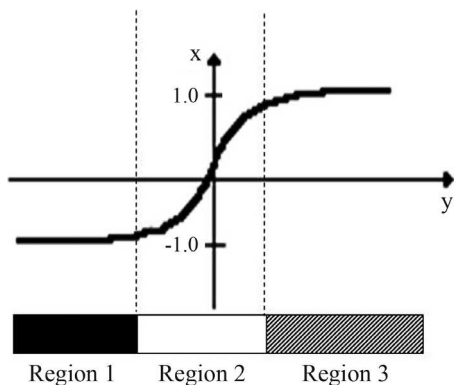


Figure 3 - The three regions defined in the sigmoid function to analyze the input data and to extract rules from the trained NN.

since they best represented the input data. The result of this approach was a rule prototype, which we defined as the input data set average. The rule can be written as a linear equation: “If $x \cong$ prototype then $y =$ constant of a linear equation + (coefficients of the linear equation)”. Here, x is an input example, y corresponds to the NN output and the coefficients of the linear equation are the nucleotides of the sequence. This approach is referred to as FAGNIS, according to Cechin (1998). Rule extraction was done in the R Environment (R Developed Core Team, 2005).

(2) *Rules by a decision tree*: These rules were obtained using the software Weka with the algorithm J-48 (Witten and Frank, 2005). The decision tree is an analytical tool to find rules and relations by subdividing information in the data analyzed. The tree consists of nodes that represent attributes and arches from the nodes that were assigned possible values for these attributes. The first node corresponds to the root from which the other nodes were derived. These derived nodes are referred to as leaf-nodes and repre-

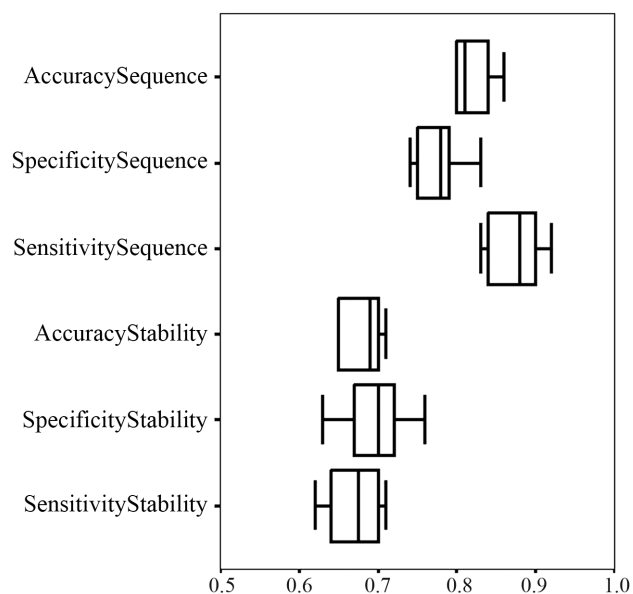


Figure 4 - Box plot for accuracy, specificity and sensitivity in the ten-fold cross-validation for both simulations.

sent the distinct classes of each training set. The possible ways of running the tree can be written in an *if-then* rule format.

Results and Discussion

Classification analysis

Analysis of the results initially involved a root mean square (RMS) evaluation. Figure 5 shows the RMS plot of the best NN architecture for both simulations. In the sequence-based simulation, the lowest RMS was achieved with 30 train epochs, *i.e.*, this number of epochs yielded the best accuracy, which was 0.8 (80%) with a standard deviation of 0.04 (4%). For the stability-based simulation, the best NN yielded an RMS with 40 train epochs (Figure 3), an accuracy of 0.68 (68%) and a standard deviation of 0.023 (2.3%).

The quality of the classification is shown by the confusion matrix for both simulations (Table 1). The specificity and sensitivity of the results for the sequence-based simulation was 0.9 (90%) and 0.65 (65%), respectively. For the stability-based simulation, the values for these two parameters were 0.7 (70%) and 0.67 (67%), respectively.

The box plot (Figure 4) shows the distribution of the values for accuracy, specificity and sensitivity in the ten-fold cross-validation. The central line represents the median, the base of the rectangle is proportional to the number of cases, and the lower and upper boundaries of the box show the lower and upper quartiles, respectively. The length of the box therefore corresponds to the inter-quartile range, which is a convenient and popular measure of the spread. For both simulations, the small length of the boxes indicated low variation in accuracy and sensitivity; specificity showed the greatest variation.

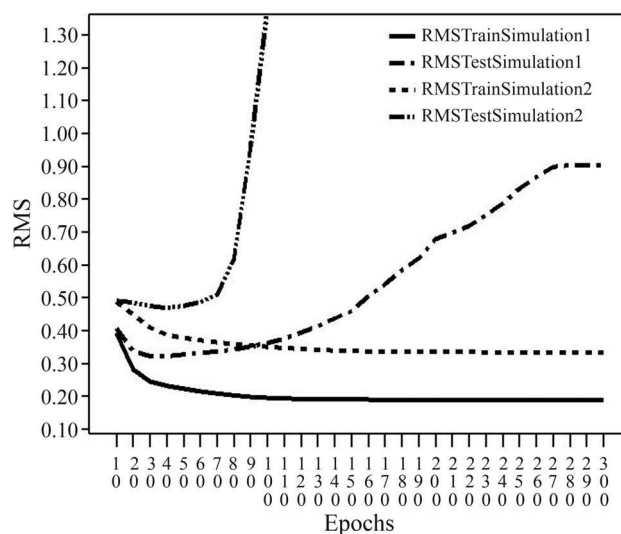


Figure 5 - Plot of the RMS for 300 train epochs. The RMS train showed a slow decrease in both simulations. In this test, the lowest RMS value was observed at epoch 30 in the sequence-based simulation and epoch 40 in the stability-based simulation.

Table 1 - Confusion matrix for the NN architecture described in the text. The sequences were classified as promoters and non-promoter (random) sequences.

	Sequence-based simulation		Stability-based simulation	
	Classified as promoter	Classified as non-promoter	Classified as promoter	Classified as non-promoter
Promoter	66	28	63	31
Non-promoter	9	85	28	66

The classification results showed that the NN provided a good generalization for the input data in the sequence-based simulation. The NN classified random sequences more correctly than it did promoter sequences (Table 1). This fact probably reflected the incomplete conservation of the consensus hexameric sequence of the promoters and the presence of several consensus promoter sequences for each σ factor (Lewin, 2008). The NN was unable to learn a single pattern for the input data because of different motifs present in the σ factor family, *e.g.*, the consensus sequences for σ^{24} are 'CTAAA' for the -35 region and 'GCCGATAA' for the -10 region. Consequently, the NN created a general classification rule based on similar features for all promoter sequences. For this reason, the sensitivity observed here was lower than that observed in other papers. This finding was reflected in the low number of epochs necessary for learning, an indication that the input data had noise typical of biological data (Losa *et al.*, 1998). In contrast, the results from the stability-based simulation showed that the NN was unable to correctly classify the random sequences (Table 1). This finding can be explained by the lack of data synchronization since it was not possible to pre-align the sequences. Sequence alignment was not feasible because it was impossible to obtain stability values for the gaps inserted during alignment.

The results for the sequence-based simulation were very similar to those reported by others (Table 2). Burden *et al.* (2005) reported a specificity of 0.6 (60%) and sensitivity of 0.5 (50%) for their NN-based analysis. The usefulness of this tool (referred to as NNPP) was improved when the estimated probability that a given sequence was a true promoter was reduced by 60%. Gordon *et al.* (2003) developed an SVM-based approach using a sequence alignment kernel and reported an accuracy of 0.84 (84%), a specificity of 0.84 (84%) and a sensitivity of 0.82 (82%). Web tools such

as BPROM claim an accuracy of 0.8 (80%). The papers or web tools described are only for σ^{70} promoter sequences whereas our NN used all known promoter sequences. In addition, in most previous studies the number of sequences used was lower than that used here. The results of the stability-based simulation were poor, but this simulation can be useful for subsequent predictions and can expand the range of tools for promoter prediction.

Rule extraction in the sequence-based simulation

In this simulation, five rules were extracted by the FAGNIS method. The decision tree was obtained by using the J-48 algorithm (Figure 7). To facilitate comprehension, only promoter rules will be discussed. The rules from FAGNIS yielded the promoter prototype shown in Figure 6 and identified the nucleotides that were most important in the learning process.

All of the nucleotides underlined in Figure 6 were the most important for the learning process. The nucleotides located in regions -35 and -10 (read left to right) are indicated in bold. The similarity of the nucleotides identified by the prototype with the consensus biological sequence was clear since most of the sequences belonging to the data set were recognized by σ^{70} . In addition to these consensus nucleotides, there are other nucleotides that are crucial for NN but they are not located in regions of known biological importance. The nucleotides in parentheses all have equal importance for NN. The pattern shown here was also observed with the decision tree discussed below.

The ten-fold-cross-validation method was used to obtain statistically valid results for the extraction rule based on the decision tree. The resulting tree had 31 nodes and 25 leaves (Figure 7a). The frequency of correctly classified sequences was 63% and the promoter precision was 68%. The trees showed nucleotide 25 (located in the -35 region) as the

Table 2 - Comparison of different methods used to calculate accuracy, specificity and sensitivity.

Procedure	Accuracy	Specificity	Sensitivity	Author
Sequence-based simulation	0.8	0.9	0.65	This paper
Stability-based simulation	0.68	0.7	0.67	This paper
Sequence alignment kernel	0.84	0.84	0.82	Gordon <i>et al.</i> (2003)
NNPP	N.I	0.6	0.5	Oppon (2000)
NNPP 2.2	0.49	N.I	N.I	Burden <i>et al.</i> (2005)

The numbers in bold indicate the highest scores for each parameter. N.I. = no information.

If Promoter then

```

1   2   3   4   5   6   7   8   9   10  11  13  13  14  15  16
T (TA) T A T (AT) (AT) A A A A (AT) (AT) T T T

17  18  19  20  21  22  23  24  25  26  27  28  29
T (ATG) (AT) (AT) (AT) A A (AT) (AT) (AT) T T T

30  31  32  33  34  35  36  37  38  39  40  41  42  43  44
(ATC) A A A (TG) T T (AT) A T T (ATC) (AC) T A

45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61
T (ATC) A T A T A T T A T A A T T (AT) A

62  63  64  65  66  67  68  69  70  71  72
T (AT) (AT) TG (AT) (AT) (AT) A T A A

```

Figure 6 - Rule prototype for promoter sequence obtained from NN learning.

root. The presence of guanine at this position was sufficient to identify a given sequence as a non-promoter. The other nucleotides present in the rules were located in the -10 region that included nucleotides 46 to 54, approximately. Some of the rules identified by this approach included:

- a) *If Promoter then* nucleotide_25 = A, nucleotide_45 = T and nucleotide_46 = A or G
- b) *If Promoter then* nucleotide_25 = T and nucleotide_47 = A or T
- c) *If Promoter then* nucleotide_25 = T and nucleotide_47 = C and nucleotide_50 = A or T
- d) *If Promoter then* nucleotide_25 = C and nucleotide_45 = T

These rules shared many similarities with the prototype obtained with the trained NN. Clearly there is a strong relationship among the nucleotides located in the biological

motifs. Despite the incomplete conservation of these motifs, they are still an important feature used by NN for learning. The rules generally agreed with current biological knowledge.

Rule extraction in stability-based simulations

For this simulation, rule extraction using FAGNIS generated seven rules, of which only one classified sequences as a non-promoter (rule 1). The prototypes of the rules are shown as plots for better comprehension (Figure 8). In four promoter prototypes (rules 4 to 7) there was a decrease in the ΔG values in the -10 region (located between nucleotides 45 and 52). These rules were valid for 135 promoter sequences, which were classified based on these rules. The two promoter prototypes that accounted for the majority of promoter sequences (total of 533) showed

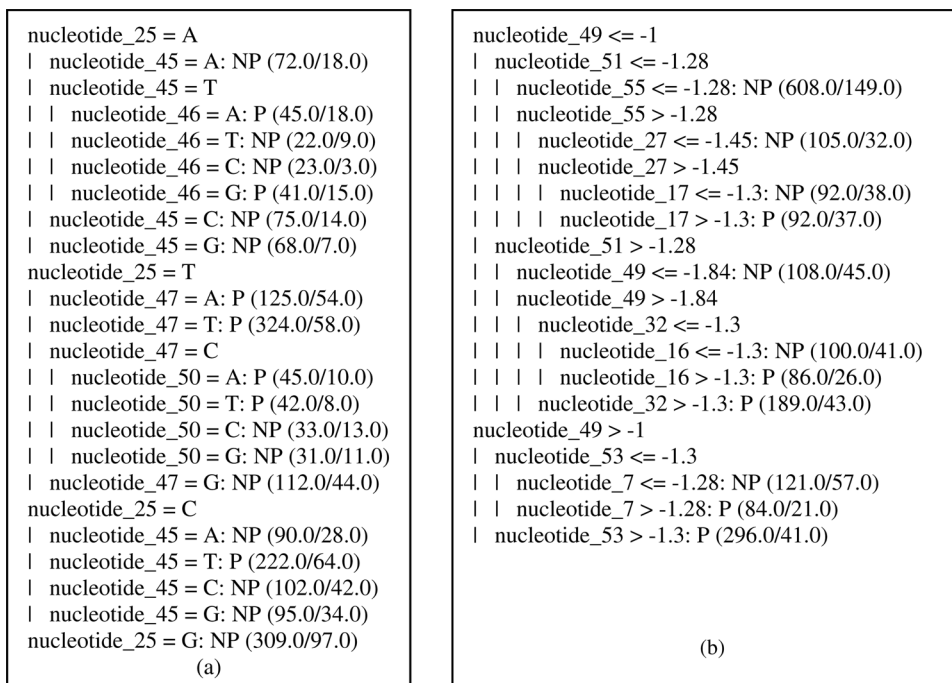


Figure 7 - Decision tree based on the J-48 algorithm. (a) Decision tree for the sequence-based simulation data. (b) Decision tree for the stability-based simulation data.

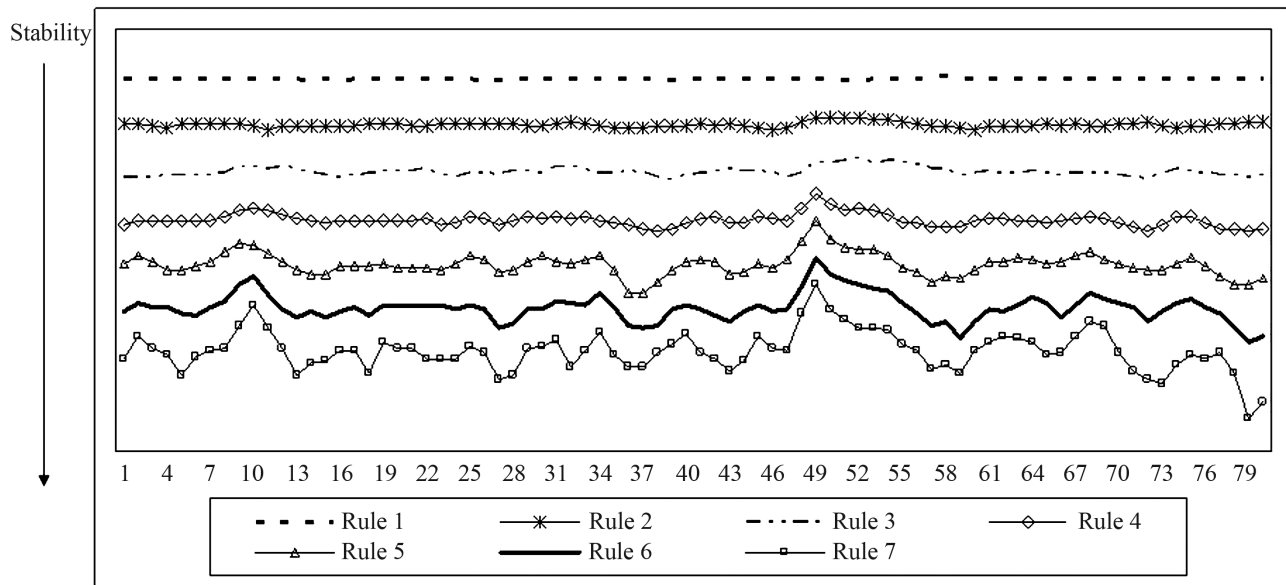


Figure 8 - Plot for the stability-based simulation data. This plot represents the rule prototype obtained with the FAGNIS method for rule extraction. Only rule 1 is for non-promoter sequences; the other rules are for prototype promoter sequences.

no evident decrease in this region. Four patterns of promoters were identified in the plots, in contrast to NN learning, and this explains the poor rate of correct sequence classification.

The rules obtained using the J-48 algorithm were extracted from the decision tree shown in Figure 7b. The ten-fold cross-validation method was also used. The resulting tree had 21 nodes and 11 leaves. The success rate for correctly classified sequences was 66.8% and the promoter precision was 68%. The rules that classified a sequence as a promoter are shown in Figure 7b. These rules showed that there was a relationship between the two consensus motifs of promoters. The root of the tree was nucleotide 49 (located in the -10 region), but there were other important nucleotides in the -10 and -35 regions. Some nucleotides (7, 13, 16, 17 and 32) occurred at positions with no known biological function. This analysis also revealed the stability low value of the nucleotides and the absence of guanine at position 49. This fact can explain the high ΔG value that this nucleotide has when it occurs as a neighbor of another nucleotide.

In conclusion, the usefulness of NN for promoter prediction and recognition was assessed using two data sets. The accuracy of the sequence-based simulation was 0.80 ± 0.04 while that of the stability-based simulation was 0.68 ± 0.02 . These results were comparable to those reported in the literature. The rules extracted from NN learning can help to identify the most important nucleotide promoter patterns. The pattern obtained is representative of all sequences, despite the σ factor that recognizes each promoter. The data obtained by this approach can help in promoter prediction and increase our knowledge of the biological role of promoters. Generally, NN-based methods and

machine learning techniques for promoter prediction rely on the stability of promoter sequences less frequently than on nucleotide sequence information. The confusion matrix showed that NN could differentiate promoters and random sequences based on nucleotide information, but this was not the case when stability information alone was used. The results of this study indicate that the use of nucleotide sequence and structural characteristics as input data may help to improve the prediction of bacterial promoters. This finding should provide a stimulus for developing more efficient algorithms for predicting such promoters.

Acknowledgments

We are grateful to Universidade de Caxias do Sul (UCS) for financial support.

References

- Andrews R, Diederich J and Tickle AB (1995) A survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-Based Syst* 6:373-389.
- Burden S, Lin Y-X and Zhang R (2005) Improving promoter prediction for the NNPP2.2 algorithm: A case study using *Escherichia coli* DNA sequences. *Bioinformatics* 21:601-607.
- Cechin AL (1998) *The Extraction of Fuzzy Rules from Neural Networks*. Shaker Verlag, Aachen, 149 pp.
- Cotik V, Zaliz RR and Zwir I (2005) A hybrid promoter analysis methodology for prokaryotic genomes. *Fuzzy Sets Syst* 1:83-102.
- Demeler B and Zhou G (1991) Neural network optimization for *E. coli* promoter prediction. *Nucleic Acids Res* 19:1593-1599.
- Gama-Castro S, Jimenez-Jacinto V, Peralta-Gil M, Santos-Zavaleta A, Peñaloza-Spinola MI, Contreras-Moreira B, Segura-Salazar J, Muñiz-Rascado L, Martinez-Flores I, Salgado H, *et al.* (2008) RegulonDB (v. 6.0): Gene regulation model

- of *Escherichia coli* K-12 beyond transcription, active (experimental) annotated promoters and text press navigation. *Nucleic Acids Res* 36:D120-D124.
- Gordon L, Chervonenkis A, Gammerman AJ, Shahmuradov IA and Solovyev VV (2003) Sequence alignment for recognition of promoter regions. *Bioinformatics* 19:1964-1971.
- Howard D and Benson K (2002) Evolutionary computation method for pattern recognition of *cis*-acting sites. *BioSystems* 72:19-27.
- Jáuregui R, Abreu-Goodger C, Moreno-Hagelsieb G, Collado-Vides J and Merino E (2003) Conservation of DNA curvature signals in regulatory regions of prokaryotic genes. *Nucleic Acids Res* 31:6770-6777.
- Kalate R, Tambe SS and Kulkarni B (2003) Artificial neural networks for prediction of mycobacterial promoter sequences. *Comput Biol Chem* 27:555-564.
- Kanhere A and Bansal M (2005a) Structural properties of promoters: Similarities and differences between prokaryotes and eukaryotes. *Nucleic Acids Res* 33:3165-3175.
- Kanhere A and Bansal M (2005b) A novel method for prokaryotic promoter prediction based on DNA stability. *BMC Bioinformatics* 6:1471-2105.
- Lewin B (2008) *Genes IX*. Artmed, Porto Alegre, 955 pp.
- Losa GA, Merlini D, Nonnenmacher TF and Weibel ER (1998) *Fractals in Biology and Medicine. Vol. II, Mathematics and Bioscience in Interaction*. Birkhäuser, Basel, 321 pp.
- Mahadevan I and Ghosh I (1994) Analysis of *E. coli* promoter structures using neural networks. *Nucleic Acids Res* 22:2158-2165.
- O'Neill, MC (1991) Training back-propagation neural networks to define and detect DNA-binding sites. *Nucleic Acids Res* 19:313-318.
- Oppon EC (2000) Synergistic Use of Promoter Prediction Algorithms: A Choice for a Small Training Dataset? Doctorate in Computational Science, South African National Bioinformatics Institute, 238 pp.
- Pandey SP and Krishnamachari A (2006) Computational analysis of plant RNA Pol-II promoters. *Biosystems* 83:38-50.
- Pedersen AG, Baldi P, Brunak S and Chauvin Y (1996) Characterization of prokaryotic and eukaryotic promoters using hidden Markov models. In: *Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology*, St. Louis, pp 182-191.
- Polate K and Günes S (2007) A novel approach to estimation of *E. coli* promoter gene sequences: Combining feature selection and least square support vector machine (FS_LSSVN). *Appl Math Comput* 190:1574-1582.
- SantaLucia J and Hicks D (2004) The thermodynamics of DNA structural motifs. *Annu Rev Biophys Biomol Struct* 33:415-440.
- Thompson JD, Higgins DD and Gibson TJ (1994) Clustal W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting position-specific gap penalties and weight matrix choice. *Nucleic Acids Res* 22:4673-4680.
- Witten IH and Frank E (2005) *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufman, San Francisco, 560 pp.

Internet Resources

- R Development Core Team (2005) R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. <http://www.R-project.org> (November 20, 2005).

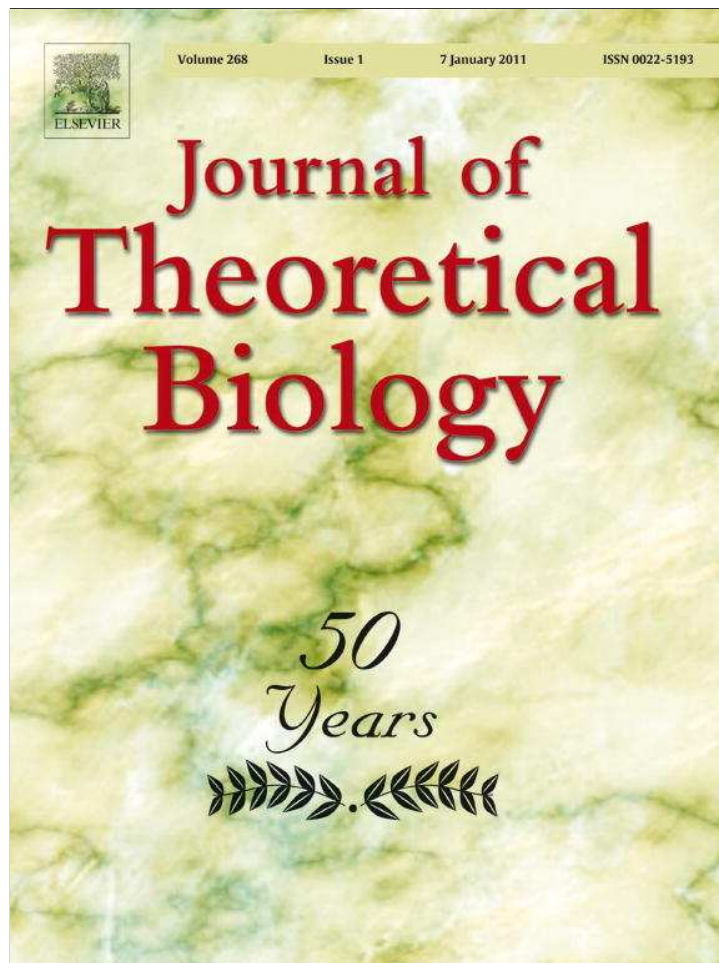
Associate Editor: Luciano da Fontoura Costa

License information: This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

5.2 CAPÍTULO II - *BACPP: BACTERIAL PROMOTER PREDICTION - A TOOL FOR ACCURATE SIGMA-FACTOR SPECIFIC ASSIGNMENT IN ENTEROBACTERIA*

Este capítulo apresenta o artigo científico “*BacPP: Bacterial promoter prediction - A tool for accurate sigma-factor specific assignment in enterobacteria*”. Nele, há a descrição da metodologia de implementação da ferramenta apresentada no capítulo anterior e os resultados obtidos. O artigo foi publicado na revista *Journal of Theoretical Biology*, período com fator de impacto 2,371 e classificado pela CAPES como qualis A1 na área de avaliação interdisciplinar. Este documento pode ser acessado pela internet pelo [doi:10.1016/j.jtbi.2011.07.017](https://doi.org/10.1016/j.jtbi.2011.07.017). Este software tem o depósito de patente realizada no *United States Patent and Trademark Office*. Os formulários e o código-fonte da ferramenta encontram-se no apêndice 1.

Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.



(This is a sample cover image for this issue. The actual cover is not yet available at this time.)

This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

<http://www.elsevier.com/copyright>



Contents lists available at ScienceDirect

Journal of Theoretical Biology

journal homepage: www.elsevier.com/locate/yjtbi

BacPP: Bacterial promoter prediction—A tool for accurate sigma-factor specific assignment in enterobacteria

Scheila de Avila e Silva^{a,*}, Sergio Echeverrigaray^a, Günther J.L. Gerhardt^{b,1}

^a Universidade de Caxias do Sul, Programa de Pós-Graduação em Biotecnologia, Av. Francisco Getúlio Vargas, 1130-CEP 95070-560, Caxias do Sul-RS, Brazil

^b Universidade de Caxias do Sul, Departamento de Física e Química, Av. Francisco Getúlio Vargas, 1130-CEP 95070-560, Caxias do Sul-RS, Brazil

ARTICLE INFO

Article history:

Received 29 October 2010

Received in revised form

20 May 2011

Accepted 21 July 2011

Available online 3 August 2011

Keywords:

Neural network

Rule extraction

Gene expression regulation

ABSTRACT

Promoter sequences are well known to play a central role in gene expression. Their recognition and assignment *in silico* has not consolidated into a general bioinformatics method yet. Most previously available algorithms employ and are limited to $\sigma 70$ -dependent promoter sequences. This paper presents a new tool named BacPP, designed to recognize and predict *Escherichia coli* promoter sequences from background with specific accuracy for each σ factor (respectively, $\sigma 24$, 86.9%; $\sigma 28$, 92.8%; $\sigma 32$, 91.5%; $\sigma 38$, 89.3%, $\sigma 54$, 97.0%; and $\sigma 70$, 83.6%). BacPP is hence outstanding in recognition and assignment of sequences according to σ factor and provide circumstantial information about upstream gene sequences. This bioinformatic tool was developed by weighing rules extracted from neural networks trained with promoter sequences known to respond to a specific σ factor. Furthermore, when challenged with promoter sequences belonging to other enterobacteria BacPP maintained 76% accuracy overall.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

Prediction and recognition of transcription sites *in silico* from the overwhelming bulk and diverse background genomic sequence is an active research topic in molecular biology and a challenge in bioinformatics. This knowledge essentially defines our ability to generate hypotheses in the context of the fundamental topic of bacterial gene function and regulation. Promoters are *cis*-acting sequence elements located upstream of the transcription start site (TSS) of open reading frames (ORF). These elements choreograph the initiation of gene expression through their recognition by RNA polymerase (RNAP). In bacteria, RNAP holoenzyme consists of five subunits (2α , β , β' , ω) and an additional sigma (σ) subunit factor (Borukov and Nudler, 2003). A small collection of different σ subunits act as key regulators of bacterial gene expression by dictating the RNAP sequence-specific binding at the level of the promoter where melting of the DNA double strand occurs (Borukov and Nudler, 2003; Li and Lin, 2006). Bacterial cells use alternative σ factors for subsets of promoters, endowing them ability of adapting gene expression to environmental changes (Borukov and Nudler, 2003). In the model enterobacterium *Escherichia coli* the most prevalent σ

subunit are labeled by molecular weight: $\sigma 24$, $\sigma 28$, $\sigma 32$, $\sigma 38$, $\sigma 54$ and $\sigma 70$. Each σ factor subfamily has been assigned a global functional role and are characterized by their recognition of a distinct consensus promoter sequence. For example, $\sigma 32$ plays a role in heat-shock response, $\sigma 28$ is associated with expression of flagellar genes during normal growth and the better known $\sigma 70$ is the major factor responsible for the bulk housekeeping transcriptional activity in standard laboratory conditions (Borukov and Nudler, 2003; Lewin, 2008). Regardless of the σ factor in question, all promoters can be dissected into two functional recognition sites, known as the -35 and -10 regions upstream of the TSS codon. These sequence motifs are poorly conserved and differ widely among the promoter different σ factor which recognizes them. Furthermore, the consensus motifs recognized by $\sigma 24$ and $\sigma 38$ have not been reported owing to low conservation and/or the limited number of confirmed promoters. The canonical consensus for known -35 and -10 regions and the number interspersing nucleotides are (Lewin, 2008)

$\sigma 32$ –5'–CCCTTGAA 13–15 bp CCCGATNT–3'

$\sigma 28$ –5'–CTAAA 15 bp GCCGATAA–3'

$\sigma 70$ –5'–TTGACA 16–18 bp TATAAT–3'

$\sigma 54$ –5'–CTGGNA 6 bp TTGCA–3'

The variation among consensus sequences recognized by each σ factor, in particular the relative positions of the conserved

* Corresponding author. Tel.: +55 54 3218 2100x2075.

E-mail addresses: sasilva6@ucs.br (S. de Avila e Silva),

selaguna@ucs.br (S. Echeverrigaray), gunther_lew@yahoo.com.br (G.J. Gerhardt).

¹ Tel.: +55 54 3218 2100x2607.

motifs, limits the efficacy of prediction by a global analysis approach. A limited analysis of a putative promoter sequence by comparison with the $\sigma 70$ promoter consensus motif can lead to an unacceptable rate of false negatives and incorrect assignments. Hence, global promoter prediction requires separate in-depth analysis for each σ -dependent promoter sequences.

Promoter databases have allowed the development of bioinformatic tools which predict the location of promoter regions on the basis of homology to consensus sequences or a reference list of promoters (Polate and Günes, 2007). The classical approach for promoter prediction involves the development of algorithms that use position-weight matrices (PWMs). This methodology yields results by aligning examples of reference sequences and estimating the base preference at each position of a matrix (Li and Lin, 2006; Gordon et al., 2006).

In recent years, Machine Learning approaches have been applied for promoter recognition and prediction. Among these, applications of Support Vector Machines (SVM) and Neural Network (NN) have produced promising results. The SVM methods use a training algorithm and can represent complex nonlinear functions, with the objective of separating the dataset into two classes by a hyperplane (Polate and Günes, 2007). The SVM can be applied to identify important biological elements including protein (Chen et al., 2009; Zakeri et al., 2011; Zeng et al., 2009), promoters (Polate and Günes, 2007), transcription start sites (Gordon et al., 2006), among others.

The NNs are computational tools with complex nonlinear functions. They have been applied to many bioinformatic tools including promoter prediction (Demeler and Zhou, 1991; Burden et al., 2005; Rani et al., 2007), gene expression (Janga and Collado-Vides, 2007) and protein analysis (Gama-Castro et al., 2008). The NNs are particularly adequate for promoter prediction and recognition due to their ability to identify degenerated, imprecise and incomplete patterns merged within those sequences, and can achieve high performance when processing extended genome sequences (Cotik et al., 2005). Moreover, the NN methodology allows rule extraction from trained networks, which can assist to derive biologically-relevant concepts from the input data (Andrews et al., 1995).

Herein we propose a bacterial promoter prediction tool, denoted as BacPP, not limited to exclusively employing $\sigma 70$ sequences for the prediction of all promoters. BacPP is based on rules derived from NN learning process for $\sigma 24$, $\sigma 28$, $\sigma 32$, $\sigma 38$, $\sigma 54$ and $\sigma 70$ dependent promoter sequences. The information obtained from the rules was weighted to maximize promoter prediction and classify them according to σ factor which recognize the sequence.

2. Methods

2.1. Dataset

E. coli promoter sequences obtained from the RegulonDB database (Gama-Castro et al., 2008) (version available in April, 2009) were used as positive examples for NN training. A total of 1034 sequences, subdivided according to their σ factor were employed (Table 1). As negative examples for NN training an equal number of random sequences was generated with a probability of 0.28 for nucleotides adenine (A) and thymine (T) and probability of 0.22 for cytosine (C) and guanine (G), according to (Kanhere and Bansal, 2005). To confirm the specificity of BacPP, a set of 1034 randomly chosen intergenic regions (> 80 bp upstream) were also used as negative examples.

As elucidated in a recent comprehensive review (Chou, 2011), to avoid homology bias and remove the redundant sequences from the benchmark dataset, a cut-off threshold of 25%

Table 1
Number of sequences employed in the simulation for each σ factor.

σ factor	Number of promoters sequences
$\sigma 24$	69
$\sigma 28$	21
$\sigma 32$	71
$\sigma 38$	99
$\sigma 54$	38
$\sigma 70$	740

imposed in (Chou and Shen, 2010) to exclude those proteins from the benchmark datasets that have equal to or greater than 25% sequence identity to any other in a same subset. However, in this study we did not use such a stringent criterion because the currently available data do not allow us to do so. Otherwise, the numbers of promoter sequences for some subsets would be too few to have statistical significance.

Promoter sequences from other *Enterobacteriaceae* were obtained from available literature, since currently the only web databases resources are for *E. coli* and *Bacillus subtilis*. Thus, a collection of 82 promoter sequences belonging to *Citrobacter*, *Enterobacter*, *Klebsiella*, *Proteus*, *Salmonella*, *Shigella*, *Yersinia* genera were also employed (Ching and Inouye, 1986; Kutsukake et al., 1990; Mares et al., 1992; Tobe et al., 1993; Smith and Somerville, 1997; Sulavik et al., 1997; Beach and Osuna, 1998; Wösten and Groisman, 1999; Penfound and Foster, 1999; Castellanos et al., 2000; Hu et al., 2000; Ibanez-Ruiz et al., 2000; Ramírez-Santos et al., 2001; Aldridge et al., 2006; Maxson and Darwin, 2006; Skovierova et al., 2006; Yang et al., 2008; Perez and Groisman, 2009).

2.2. Neural network simulation

NN simulations were carried out for each σ dependent promoter sequences. The nucleotides were encoded by four binary digits: A=0100, T=1000, C=0001 and G=0010 (Brunak et al., 1991). An input sequence was classified as presumptive promoter if its output lay value was between 0.5 and 1.0, according to the σ -dependent promoter sequence. Otherwise, it was considered as a non-promoter.

The simulations were carried out in the R Environment (R Development Core Team, 2008). We selected the *back-propagation* (BP) algorithm with a *k-fold-cross validation*, in order to obtain statistically valid results. In this technique the dataset was divided into *k* subsets. At each iteration one of the *k* subsets was used as the test-set and the others formed a training set. The average error across all *k* trials was then computed (Polate and Günes, 2007). The *k* values determined by the number of promoter sequences available were 10 for $\sigma 70$ promoters; 2 for $\sigma 28$ and $\sigma 54$ promoters; 3 for $\sigma 24$, $\sigma 32$ and $\sigma 38$ promoters.

Among the independent dataset test, sub-sampling (e.g., 2, 5 or 10-fold cross-validation) test, and jackknife test, which are often used for examining the accuracy of a statistical prediction method (Chou and Zhang, 1995), the jackknife test was deemed the most objective that can always yield a unique result for a given benchmark dataset, as elucidated in (Chou and Shen, 2008) and demonstrated by Eq. (50) of (Chou and Shen, 2007). Therefore, the jackknife test has been increasingly and widely adopted by investigators to test the power of various prediction methods (Chen et al., 2009; Zakeri et al., 2011; Zeng et al., 2009; Kandaswamy et al., 2011; Lin et al., 2009; Mohabatkar, 2010; Nanni and Lumini, 2009; Xiao et al., 2008; Xiao et al., 2009; Xiao et al., 2011a; Xiao et al., 2011b) as well as a long list of references cited in a recent review (Chou, 2011). However, to reduce the

computational time, we adopted the k -fold cross-validation in this study as done by many investigators with NN as the prediction engine.

The results were evaluated comparing the parameters accuracy (A), specificity (S) and sensitivity (SN) calculated from Eqs. (1)–(3), respectively,

$$A = \frac{TP+TN}{TN+TP+FN+FP} \quad (1)$$

$$S = \frac{TN}{TN+FP} \quad (2)$$

$$SN = \frac{TP}{TP+FN} \quad (3)$$

where, TP (true positive) are promoter sequences classified as promoter; TN (true negative) are random sequences recognized as non-promoters; FP (false positive) are random sequences classified as promoter and FN (false negative) are promoters classified as non-promoter sequences. The same formulas were used for analyzing BacPP simulations, described in the next section.

The NN is applicable to a variety of problems, but the learning process is complex (Andrews et al., 1995). How NN learns to classify sequences as promoter or non-promoter can be understood by rules extraction and the explanation of how each NN decision is made increases the knowledge about the sequences themselves (Andrews et al., 1995). In this paper, we extracted the rules based on the value of the hidden neurons by the Fuzzy Automatically Generated Neural Inferred System (FAGNIS) technique. A brief explanation about FAGNIS is provided and further explanation and details would be obtained in Battistella and Cechin (2004).

This technique consists in segmenting a sigmoid function in three regions. For each input, it is verified in which region of the sigmoid the hidden neurons would fit. Because of peculiarities of data generally the activation functions (f_A) works on a small input range. The maximum number of combinations is 3^n , where n is a number of neurons in the hidden layer. Nevertheless, all the possible combinations do not occur and only the more frequent combinations are considered, which are the best representation of the input data. FAGNIS is based in to substitute the activation functions for a set of linear segments. The value of f_A can be approximated by a set of linear segments, using a relationship as demonstrated in

$$f_A(a_j) \sim \sum_i [F_i(a_j)(p_i a_j + q_i)] \quad (4)$$

where $f_A(a_j)$ is the original nonlinear function, a_j is the activation signal (weighted sum of the input vector), $F_i(a_j)$ is a function that links each value of a_j to correspondent linear segment(s), and $p_i a_j + q_i$ are linear segments. To improve the precision of the approximation, $F_i(a_j)$ must be a fuzzy number.

Therefore, the results are conveniently presented by a rule prototype, which we defined as the average model of the input dataset. The rule can be written as a linear equation: "If $x \cong$ prototype then $y =$ constant of linear equation + (coefficients of the linear equation)". Here, x is an input example, y corresponds to the NN output and the coefficients of the linear equation are the nucleotides of the sequence.

This methodology has been successfully used to extracted rules from other biological problems by using NNs (Battistella and Cechin, 2004).

2.3. BacPP implementation

The BacPP tool was implemented in Python programming language. An overview of this approach is given in Fig. 1. The objective of this tool was to weight the score obtained from NN

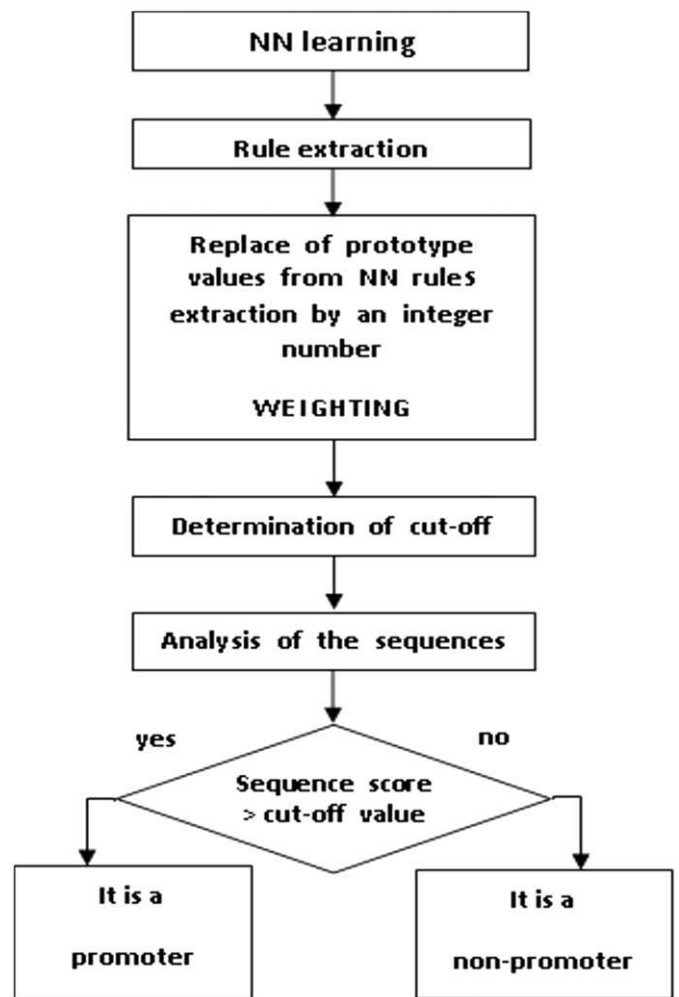


Fig. 1. The flow chart of BacPP approach.

Table 2
The best weighting set to BacPP classification results.

Score from NN rule prototype	Integer number replace it
Upon 0.6	+6
0.5–0.59	+4
0.4–0.49	+2
0.3–0.39	+1
0.2–0.29	0
0.1–0.19	–1
Lower 0.1	–3

prototypes of rules extracted for each σ -dependent promoter sequences. Thereafter, we used these scores as models to determine and classify promoters according to σ factor which recognize them. Several weighing schemes were evaluated. The weights were defined using integer numbers between -10 and $+10$. For a given nucleotide, if the prototype score lied above 0.3 or below 0.2 , the values were replaced by a positive or negative number, respectively. If the prototype score lied between 0.29 and 0.2 , the prototype values were replaced by zero. The best weights are presented in Table 2. The cut-off values were determined by the intersection of the histogram plot. The values of this plot are the scores obtained from BacPP for the promoters and non-promoter sequences. For these reasons, each σ factor presents a different cut-off value.

3. Results

In the NN simulation, the architecture that best classified the input set of sequences for each σ -dependent promoter is presented in Table 3. A greater number of neurons in the hidden layer did not significantly increase accuracy, specificity or sensitivity.

Using the best architecture for each σ -dependent promoter, the NN achieved an average accuracy of 71.67%, a specificity of 71.08% and a sensitivity of 72.98%, with low variation among σ factors (Table 4). The similarity between the specificity and sensitivity values within each σ is an indicative of the consistence of the NN learning process. The accuracy, specificity and sensitivity values obtained for $\sigma 70$ are lower but comparable with those previously reported using NN methodologies (Demeler and Zhou, 1991; Burden et al., 2005; Rani et al., 2007; Askary et al., 2009). The ROC curve is presented in Fig. 2.

The prediction validity is often examined by observing its ROC curve (receiver operator characteristic curve), which shows the trade-off between sensitivity and specificity. The definition and theory of ROC has been described in the literature. The ROC curve is a plot of the sensitivity against the specificity for different cut-off points. Each point on the ROC curve represents a sensitivity/specificity pair corresponding to a particular decision threshold (Xu et al., 2010). Fig. 2 shows the classification of the best NN simulation for all σ -dependent promoter sequences. The most evident characteristic is observed for $\sigma 70$ results, which passes through the upper left corner. The results for the others σ -dependent promoter sequences are very similar.

The NN prototypes were extracted from the trained NN by the FAGNIS algorithm. This procedure was carried out for each σ -sequence dependent and the results are shown as a frequency plot in Fig. 3 (Crooks et al., 2004).

The prototype obtained for $\sigma 24$ (Fig. 3a) did not show any conserved motifs, but as most promoters, it is characterized by a high prevalence of AT nucleotides (Lewin, 2008). As far as we know, no conserved motifs have been described for $\sigma 24$ promoters. Conversely, the $\sigma 28$ promoter prototype (Fig. 3b) exhibited two conserved motifs, one between -15 and -7 , TGCCGATAA, and the other between -33 and -25 , TAAAGTTT, that match those previously described (Song et al., 2007).

Table 3
The best architecture for every σ factor family simulation.

σ factor	Number of neurons in input layer	Number of neurons in hidden layer	Number of neurons in output layer
$\sigma 24$	324	4	1
$\sigma 28$	324	2	1
$\sigma 32$	324	2	1
$\sigma 38$	324	2	1
$\sigma 54$	324	2	1
$\sigma 70$	324	5	1

Table 4
The best results for every σ factor family in NN simulation.

σ factor	Accuracy	Specificity	Sensitivity
$\sigma 24$	71.6	69.1	73.9
$\sigma 28$	70.2	66.6	73.8
$\sigma 32$	72.4	71.4	73.4
$\sigma 38$	67.7	68.5	66.8
$\sigma 54$	73.5	73.2	73.8
$\sigma 70$	77.0	76.7	77.2

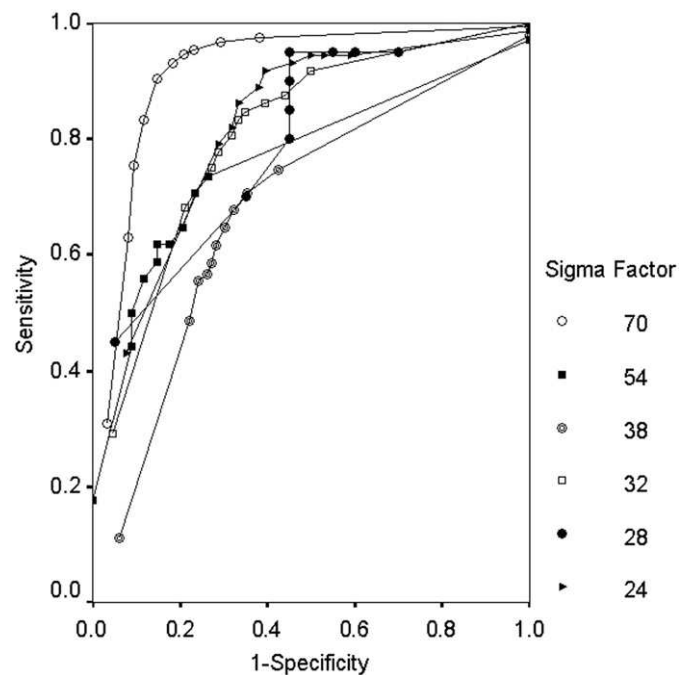


Fig. 2. The ROC curve for the best NN simulation for each σ -dependent promoter sequences.

The prototype obtained for the $\sigma 32$ promoters (Fig. 3c) was characterized by the presence of two partially conserved motifs. One from -7 to -15 with a consensus sequence CYC YAWWWW, and one from -28 to -35 with YTKRWWW sequence. According to IUPAC code, the letters Y, W, K, R represents: C or T, A or T, G or T, A or G, respectively. These motifs are similar to those described by (Wang and deHaseth, 2003). On the other hand, no conserved motifs were obtained for $\sigma 38$ promoters (Fig. 3d). However, a W-rich region could be identified in the first 11 nucleotides, with a highly conserved T at -7 . Low conservation and W-rich motifs on $\sigma 38$ promoters were reported by Typas et al. (2007).

The prototype obtained for $\sigma 54$ promoters (Fig. 3e) showed two conserved motifs: WWCGTT between -10 and -15 , and ACGGT between -22 and -26 . These motifs are similar to those described by Barrios et al. (1999). In Fig. 3f, the $\sigma 70$ promoter prototype was characterized by a high A/T/W content (88%) compared with the other promoters, which varied between 30% ($\sigma 32$) and 58% ($\sigma 38$). High AA, AT and TT dinucleotide frequency was reported by Kanhere and Bansal (2005) for *E. coli* $\sigma 70$ promoters. However, the typical -10 and -35 conserved motifs were not evident on the prototype extracted from the NN. This fact can be rationalized due to the variation on the number of nucleotides between the $+1$ and the first motif, as well as among motifs of $\sigma 70$ promoters (Shultzaberger et al., 2007).

In general, the prototype rules extracted from the NN showed the conserved motifs previously described for each σ -dependent promoter sequences, indicating that the NN learning has biological relevance.

Considering the efficiency of the NN learning, the rules were weighted and used to develop a prediction tool for bacterial promoter sequences, separated by σ factor.

As it can be observed in Tables 4 and 5, BacPP showed higher accuracy, specificity and sensitivity than the original NN using random sequences as negative controls. Moreover, there are no significant differences between the BacPP results presented employing either random or intergenic regions as negative examples. Using BacPP, the accuracy obtained for $\sigma 54$, $\sigma 28$ and $\sigma 32$ were all above 90%. These σ -dependent promoter sequences

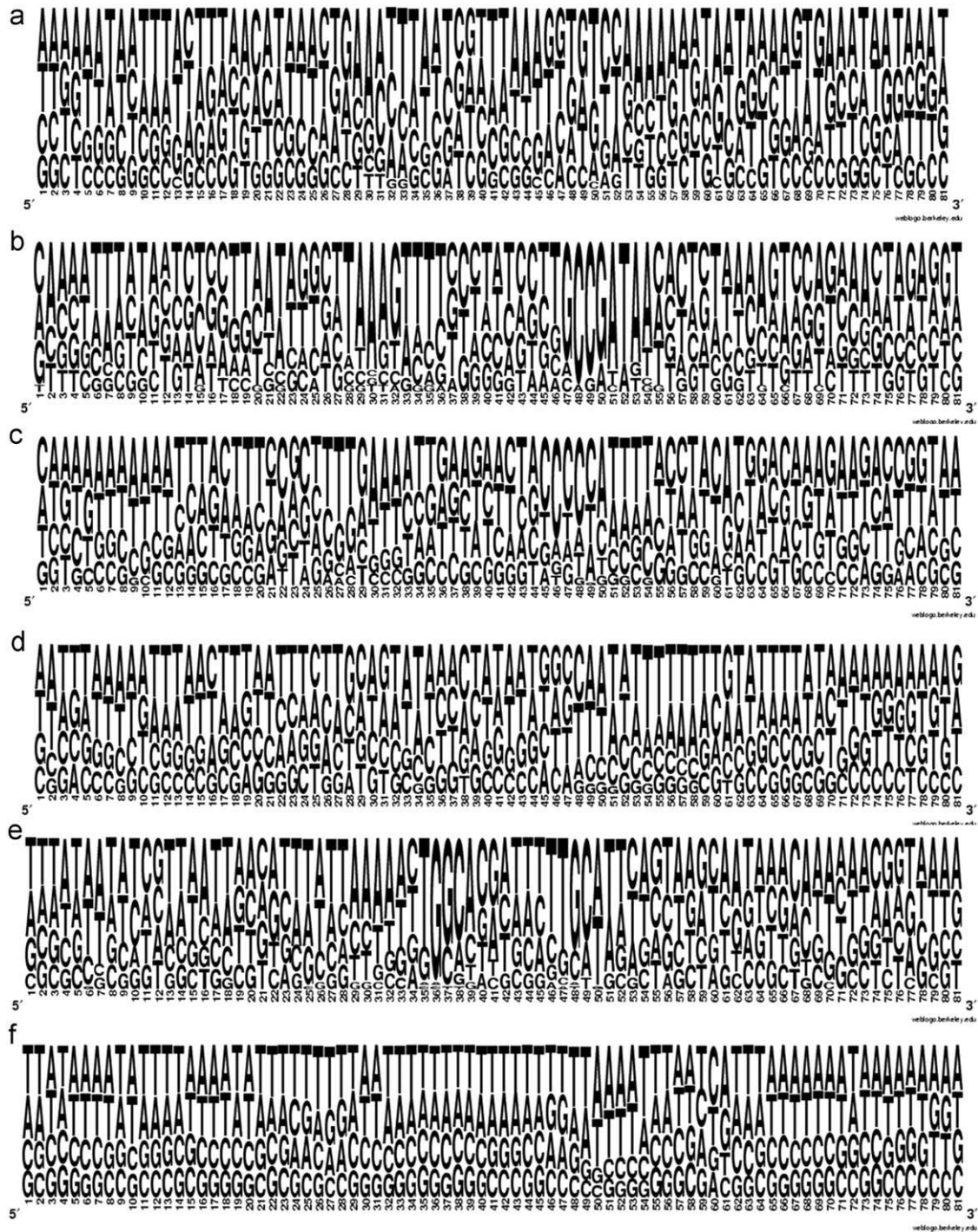


Fig. 3. The WebLogo frequency plot of NN prototypes for: (a) σ_{24} dependent-promoters; (b) σ_{28} dependent-promoters; (c) σ_{32} dependent-promoters; (d) σ_{38} dependent-promoters; (e) σ_{54} dependent-promoters; and (f) σ_{70} dependent-promoters. Here, the nucleotide 61 is the +1 nucleotide.

exhibited highly conserved motifs, Fig. 3d, b and c, respectively, which may explain the efficacy of the prediction. Conversely, the lowest accuracy was obtained for σ_{70} , the least conserved of σ promoters (Fig. 3f). Although the σ_{70} promoters are the most abundant in bacterial genomes, they have more deviations from the canonical consensus promoters (Janga and Collado-Vides, 2007).

To evaluate the efficiency of BacPP, the same set of promoters and random sequences were used for promoter prediction using BacPP and NNPP, the unique neural network based algorithm (Burden et al., 2005) available on the web until the finished of this

paper. As it can be seen in Table 4, BacPP showed higher accuracy, specificity and sensitivity, with values in the order of 90% for BacPP and 65% NNPP, respectively. The parameters were also higher for σ_{70} promoters compared to NNPP modeling.

In another simulation, BacPP was challenged in a cross-test to evaluate the sensibility of promoter classification by σ factor. For example, the algorithm for σ_{70} (tester) was applied to promoter sequences belonging to σ_{24} , σ_{28} , σ_{32} , σ_{38} and σ_{54} (queries). For all permutations, the highest sensibility was observed for the propriator σ factor (Table 6), indicating that BacPP can efficiently classify σ -dependent promoter sequences. It is worth noting that

Table 5

The best results for every σ factor family in BacPP simulation.

σ factor	Accuracy (%)			Specificity (%)			Sensitivity (%)		
	BacPP		NNPP ^a	BacPP		NNPP ^a	BacPP		NNPP ^a
	*RS	*IR		*RS	*IR		*RS	*IR	
σ 24	86.9	86.1	67.2	95.6	94.2	60.8	78.2	78.0	50.7
σ 28	92.8	97.1	68.2	90.4	98.1	75	95.2	96.2	50
σ 32	91.5	92.3	68.4	92.9	94.0	60.5	90.1	90.7	64.7
σ 38	89.3	86.6	68.2	83	80.8	62.6	93.9	92.4	64.6
σ 54	97	95.2	67.8	100	96.6	60	94.1	93.8	48.5
σ 70	83.6	80.5	74.4	85.4	80.8	68.7	81.8	80.3	80

*RS=random sequences as negative examples, *IR=intergenic region as negative examples.

^a Tested with random sequences as negative examples.

Table 6

Sensibility values of the cross-test for promoter sequences.

Tested tester	σ 24			σ 28			σ 32			σ 38			σ 54			σ 70		
	A	S	SN	A	S	SN	A	S	SN	A	S	SN	A	S	SN	A	S	SN
σ 24	86.9	95.6	78.2	70.8	71.2	70.4	58.4	53.5	63.3	57.0	45.4	68.6	57.3	64.7	50.0	58.1	21.7	94.5
σ 28	63.7	62.3	65.2	92.8	90.4	95.2	66.9	76.0	57.7	59.0	49.4	68.6	36.7	29.4	44.1	59.1	59.5	58.6
σ 32	62.3	68.1	56.5	71.4	61.9	80.9	91.5	92.9	90.1	72.2	68.6	75.7	72.0	64.7	79.4	66.1	65.6	66.6
σ 38	69.5	69.5	69.5	64.2	61.9	66.6	74.6	81.6	67.6	89.3	83	93.9	64.7	64.7	64.7	72.9	73.6	72.1
σ 54	67.3	76.8	57.9	78.5	71.4	85.7	70.4	76.0	64.7	61.6	49.4	73.7	97	100	94.11	68.4	63.1	73.7
σ 70	65.2	73.9	56.5	54.7	71.4	38.0	69.0	59.1	78.8	75.7	61.6	89.8	72.0	58.8	85.2	83.6	85.4	81.8

higher specificity was obtained when a σ factor models were applied on other σ promoter sequences (Table 6) compared to random sequences (Table 5). This is indicative of the degree of kinship among promoter sequences beyond σ factor and explains the relative efficiency of prediction models based only on σ 70 to adventitiously identify of other σ promoter sequences (Burden et al., 2005; Gordon et al., 2003).

We carried out a comparison of BacPP with previously reported machine-learning based programs. To our satisfaction, the performance average in terms of accuracy (90.2%), specificity (91.0%) and sensitivity (89.5%) of BacPP is highly competitive and comparably exceeds the parameters of the most efficient approaches published thus far in the context of σ 70 promoters. For example, (Oppon, 2000) reported a neural network promoter prediction program (NNPP) that exhibited 60% specificity and 50% sensitivity for σ 70 promoters, noting later improvements in specificity (Burden et al., 2005). Secondly, SVM based approach using Sequence Alignment Kernel obtained an accuracy of 84%, specificity of 84% and sensitivity of 82% (Gordon et al., 2003). Furthermore, (Polate and Günes, 2007) reported least-square support vector machines which obtained an accuracy of 84.6%, sensitivity of 90.9% and specificity of 80%.

A noteworthy NN method which uses dinucleotide pairs as input can achieve an accuracy of 96%, specificity of 98% and sensitivity of 93% for σ 70 promoters using negative examples of AT-rich (60%) sequences (Rani et al., 2007). Although this approach is highly efficient it is limited to the AT-rich sigma sequences like σ 70. Similarly, Rangannan and Bansal (2007) used DNA duplex stability to predict promoter sequences and obtained modest accuracy of 52.2% with sensitivity of 98%. By using stability promoter information in a NN simulation, (Askary et al., 2009) presents stability and sensitivity values of 94%. NN based on SIDD (stress-induced duplex destabilization) (Bland et al., 2010) showed better sensitivity (F-score—62.3) than threshold based methods (F-score—56.8).

A position-correlation scoring matrix (PCSM) applied to *E. coli* σ 70 showed 81% specificity and 91% sensitivity (Li and Lin, 2006).

A combination of position correlation score function and increment of diversity with modified Mahalanobis Discriminant used for eukaryotic and prokaryotic promoter prediction resulted in 91.4% specificity and 84.9% sensitivity for *E. coli* σ 70 (Lin and Li, 2011).

Finally, when evaluated in the context of a set of 82 promoter sequences of diverse enterobacterial species, BacPP exhibited an accuracy of 80.5%, with 86% sensitivity and 75% specificity, indicating that BacPP is reliable for promoter prediction and classification beyond the model Gram-negative organisms.

4. Conclusions

In this work, we have presented a novel and valuable approach for prediction and classification of bacterial promoters based on weighting promoter prototypes obtained from rules extracted from NNs. By separating the promoter sequences according their σ factor which recognize them, we have demonstrated that the current boundaries of prediction and classification of promoters can be dissolved. The finished bioinformatic tool, BacPP has reliable accuracy parameters across the σ families of *E. coli* promoter sequences (σ 24, σ 28, σ 32, σ 38, σ 54 and σ 70 accuracies of 86.9%, 92.8%, 91.5%, 89.3%, 97.0% and 83.6%, respectively). Furthermore, when applied to a set of promoters from diverse enterobacteria, the accuracy of BacPP was 76%, indicating that this tool can be reliably extended beyond the Gram-negative model.

In contrast to tools previously reported in the literature, BacPP is not only capable of identification of bacterial promoters in background genome sequence, but is designed to provide pragmatic classification according to σ factor—a piece of knowledge of equal importance. We anticipate BacPP will endow researchers of diverse fields with a reliable bioinformatics tool for *in silico* generation of hypotheses beyond what has been previously possible.

Further improvements and implementation of BacPP within a user-friendly platform are under progress. BacPP tool box is not

available in the web yet. Since user friendly and publicly accessible web-servers represent the future direction for developing practically more useful models, simulated methods or predictor (Chen et al., 2009), we shall make efforts in our future work to provide a web-server for the method presented in this paper. This tool is under patent process in the U.S Patent and Trademark Office.

Acknowledgments

We are grateful to Universidade de Caxias do Sul (UCS) for financial support.

Funding: Universidade de Caxias do Sul (UCS)

References

- Andrews, R., Diederich, J., Tickle, A.B., 1995. A survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-Based Systems* 8 (6), 373–389.
- Aldridge, P., Gnerer, J., Karlinsey, J.E., Hughes, K.T., 2006. Transcriptional and translational control of the *Salmonella* *fljC* gene. *Journal of Bacteriology* 188 (12), 4487–4496.
- Askary, A., Masoudi-Nejad, A., Sharafi, R., Mizbani, A., Parizi, S.N., Purmasjedi, M., 2009. N4: a precise and highly sensitive promoter predictor using neural network fed by nearest neighbors. *Genes & Genetic Systems* 84 (6), 425–430.
- Borukov, S., Nudler, E., 2003. RNA polymerase holoenzyme: structure, function and biological implications. *Current Opinion in Microbiology* 6, 93–100.
- Burden, S., Lin, Y.-X., Zhang, R., 2005. Improving promoter prediction for the NNPP2.2 algorithm: a case study using *Escherichia coli* DNA sequences. *Bioinformatics* 21 (5), 601–607.
- Battistella, E., Cechin, A.L., 2004. The protein folding problem solved by a fuzzy inference system extracted from an artificial neural network. *Lecture Notes in Computer Science* 3315, 474–483.
- Beach, M.B., Osuna, R., 1998. Identification and characterization of the *fis* operon in enteric bacteria. *Journal of Bacteriology* 180, 5932–5946.
- Brunak, S., Engelbrecht, J., Knudsen, S., 1991. Prediction of human mRNA donor and acceptor sites from the DNA sequence. *Journal of Molecular Biology* 220, 49–65.
- Barrios, H., Valderrama, B., Morett, E., 1999. Compilation and analysis of σ^{54} -dependent promoter sequences. *Nucleic Acids Research* 27 (22), 4305–4313.
- Bland, C., Newsome, A.S., Markovets, A.A., 2010. Promoter prediction in *E. coli* based on SIDD profiles and artificial neural networks. *BMC Bioinformatics* 11 (Suppl. 6), S17 (32(5)).
- Chen, C., Chen, L., Zou, X., Cai, P., 2009. Prediction of protein secondary structure content by using the concept of Chou's pseudo amino acid composition and support vector machine. *Protein & Peptide Letters* 16, 27–31.
- Cotik, V., Zaliz, R.R., Zvir, I., 2005. A hybrid promoter analysis methodology for prokaryotic genomes. *Fuzzy Sets and Systems* 1, 83–102.
- Chou, K.C., 2011. Some remarks on protein attribute prediction and pseudo amino acid composition (50th anniversary year review). *Journal of Theoretical Biology* 273, 236–247.
- Chou, K.C., Shen, H.B., 2010. Plant-mPLOC: a top-down strategy to augment the power for predicting plant protein subcellular localization. *PLoS ONE* 5, e11335.
- Ching, G., Inouye, M., 1986. Expression of the *Proteus mirabilis* lipoprotein gene in *Escherichia coli*. *The American Society of Biological Chemists* 261, 4600–4606.
- Castellanos, M.L., Harrison, D.J., Smith, J.M., Labahn, S.K., Levy, K.M., Wing, H.J., 2000. VirB alleviates H-NS repression of the *icsP* promoter in *Shigella flexneri* from sites more than one kilobase upstream of the transcription start site. *Journal of Bacteriology* 191 (12), 4047–4050.
- Chou, K.C., Zhang, C.T., 1995. Review: prediction of protein structural classes. *Critical Reviews in Biochemistry and Molecular Biology* 30 (1995), 275–349.
- Chou, K.C., Shen, H.B., 2008. Cell-PLOC: a package of Web servers for predicting subcellular localization of proteins in various organisms. *Nature Protocols* 3, 153–162.
- Chou, K.C., Shen, H.B., 2007. Review: recent progresses in protein subcellular location prediction. *Analytical Biochemistry* 370, 1–16.
- Crooks, G.E., Hon, G., Chandonia, J.M., Brenner, S.E., 2004. WebLogo: a sequence logo generator. *Genome Research* 14, 1188–1190.
- Demeler, B., Zhou, G., 1991. Neural network optimization for *E. coli* promoter prediction. *Nucleic Acids Research* 19, 1593–1599.
- Gordon, J.J., Towsey, M.W., Hogan, J.M., Mathews, S.A., Timms, P., 2006. Improved prediction of bacterial transcription start sites. *Bioinformatics* 22 (2), 142–148.
- Gama-Castro, S., Jimenez-Jacinto, V., Peralta-Gil, M., Santos-Zavaleta, A., Peñaloza-Spinola, M.I., Contreras-Moreira, B., Segura-Salazar, J., Muñoz-Rascado, L., Martínez-Flores, I., Salgado, H., Bonavides-Martínez, C., Abreu-Goodger, C., Rodríguez-Penagos, C., Miranda-Ríos, J., Morett, E., Merino, E., Huerta, A.M., Treviño-Quintanilla, L., Collado-Vides, J., 2008. RegulonDB (version 6.0): gene regulation model of *Escherichia coli* K-12 beyond transcription, active (experimental) annotated promoters and Text press navigation. *Nucleic Acids Research* 36, D120–D124.
- Gordon, L., Chervonenkis, A., Gammerman, A.J., Shahmuradov, I.A., Solov'yev, V.V., 2003. Sequence alignment for recognition of promoter regions. *Bioinformatics* 19 (15), 1964–1971.
- Hu, B., Zhu, J., Shen, S.C., Yu, G., 2000. A promoter region binding protein and DNA gyrase regulate anaerobic transcription of *nifLA* in *Enterobacter cloacae*. *Journal of Bacteriology* 182 (14), 3920–3923.
- Ibanez-Ruiz, M., Robbe-Saule, V., Hermant, D., Labrude, S., Norel, F., 2000. Identification of RpoS (σ^S)-regulated genes in *Salmonella enteric* serovar typhimurium. *Journal of Bacteriology* 182 (20), 5749–5756.
- Janga, S.C., Collado-Vides, J., 2007. Structure and evolution of gene regulatory networks in microbial genomes. *Research Microbiology* 158, 787–794.
- Kanhere, A., Bansal, M., 2005. Structural properties of promoters: similarities and differences between prokaryotes and eukaryotes. *Nucleic Acids Research* 33 (10), 3165–3175.
- Kutsukake, K., Ohya, Y., Iin, T., 1990. Transcriptional analysis of the flagellar regulon of *Salmonella typhimurium*. *Journal of Bacteriology* 172 (2), 741–747.
- Kandaswamy, K.K., Chou, K.C., Martinetz, T., Moller, S., Suganthan, P.N., Sridharan, S., Pugalenti, G., 2011. AFP-Pred: a random forest approach for predicting antifreeze proteins from sequence-derived properties. *Journal of Theoretical Biology*, 27056–27062.
- Li, Q.-Z., Lin, H., 2006. The recognition and prediction of σ^{70} promoters in *Escherichia coli* K-12. *Journal of Theoretical Biology* 242, 135–141.
- Lewin, B., 2008. *Genes IX*. Jones & Bartlett Publishers, Sudbury.
- Lin, W.Z., Xiao, X., Chou, K.C., 2009. GPCR-GIA: a web-server for identifying G-protein coupled receptors and their families with grey incidence analysis. *Protein Engineering, Design and Selection* 22, 699–705.
- Lin, H., Li, Q.Z., 2011. Eukaryotic and prokaryotic promoter prediction using hybrid approach. *Theory of Bioscience: Springer-Verlag* 10.1007/s12064-010-0114-8.
- Mares, R., Urbanowski, M.L., Stauffer, G.V., 1992. Regulation of the *Salmonella typhimurium* *metA* Gene by the MetR Protein and Homocysteine. *Journal of Bacteriology* 17 (2), 390–397.
- Maxson, M.E., Darwin, A.J., 2006. Multiple promoters control expression of the *Yersinia enterocolitica* phage-shock-protein A (*pspA*) operon. *Microbiology* 152 (4), 1001–1010.
- Mohabatkar, H., 2010. Prediction of cyclin proteins using Chou's pseudo amino acid composition. *Protein & Peptide Letters* 17, 1207–1214.
- Nanni, L., Lumini, A., 2009. A further step toward an optimal ensemble of classifiers for peptide classification, a case study: HIV protease. *Protein & Peptide Letters* 16, 163–167.
- E.C. Oppon, 2000. Synergistic use of promoter prediction algorithms: a choice for a small training dataset? 238 f. Doctorate in Computational Science—South African National Bioinformatics Institute (SANBI).
- Polate, K., Günes, S., 2007. A novel approach to estimation of *E. coli* promoter gene sequences: combining feature selection and least square support vector machine (FS_LSSVN). *Applied Mathematics and Computation* 190, 1574–1582.
- Penfound, T., Foster, J.W., 1999. NAD-dependent DNA-binding activity of the bifunctional NadR regulator of *Salmonella typhimurium*. *Journal of Bacteriology* 181 (2), 648–655.
- Perez, J.C., Groisman, E.A., 2009. Transcription factor function and promoter architecture govern the evolution of bacterial regulons. *PNAS* 106 (11), 4319–4324.
- Rani, T.S., Bhavani, S.D., Bapi, R.S., 2007. Analysis of *E. coli* promoter recognition problem in dinucleotide feature space. *Bioinformatics* 23 (5), 582–588.
- Ramirez-Santos, J., Collado-Vides, J., García-Varela, M., Gómez-Eichelmann, M., 2001. Conserved regulatory elements of the promote sequence of the gene *rpoH* of enteric bacteria. *Nucleic Acids Research* 29 (2), 380–386.
- R Development Core Team, 2008. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
- Rangannan, V., Bansal, M., 2007. Identification and annotation of promoter regions in microbial genome sequences on the basis of DNA stability. *Journal of Biosciences* 32 (5), 851–862.
- Smith, H.Q., Somerville, R.L., 1997. The *tpl* promoter of *Citrobacter freundii* is activated by the TyrR protein. *Journal of Bacteriology* 179 (18), 5914–5921.
- Sulavik, M.C., Dazer, M., Miller, P.F., 1997. The *Salmonella typhimurium* *mar* locus: molecular and genetic analyses and assessment of its role in virulence. *Journal of Bacteriology* 179 (6), 1857–1866.
- Skovierova, H., Rowley, G., Rezuchova, B., Homerova, D., Lewis, C., Roberts, M., Kormanec, J., 2006. Identification of the σ^E regulon of *Salmonella enterica* serovar Typhimurium. *Microbiology* 152, 1347–1359.
- Song, W., Maiste, P.J., Naiman, D.Q., Ward, M.J., 2007. Sigma 28 promoter prediction in members of the Gammaproteobacteria. *Federation of European Microbiological Societies* 271, 222–229.
- Shultzaberger, R.K., Chen, Z., Lewis, K.A., Schneider, T.D., 2007. Anatomy of *Escherichia coli* σ^{70} promoters. *Nucleic Acids Research* 35 (3), 771–788.
- Tobe, T., Yoshikawa, M., Mizuno, T., Sasaki, C., 1993. Transcriptional control of the invasion regulatory gene *virB* of *Shigella flexneri*: activation by VirF and repression by H-NS. *Journal of Bacteriology* 175 (19), 6142–6149.
- Typas, A., Becker, G., Hengge, R., 2007. The molecular basis of selective promoter activation by the σ^S subunit of RNA polymerase. *Molecular Microbiology* 63, 1296–1306.
- Wösten, M.M.S.M., Groisman, E.A., 1999. Molecular characterization of the PmrA regulon. *The Journal of Biological Chemistry* 274 (38), 27185–27190.

- Wang, Y., deHaseth, P.L., 2003. Sigma 32-dependent promoter activity in vivo: sequence determinants of the groE promoter. *Journal of Bacteriology* 185 (19), 5080–5086.
- Xiao, X., Wang, P., Chou, K.C., 2008. Predicting protein structural classes with pseudo amino acid composition: an approach using geometric moments of cellular automaton image. *Journal of Theoretical Biology* 254, 691–696.
- Xiao, X., Wang, P., Chou, K.C., 2009. GPCR-CA: a cellular automaton image approach for predicting G-protein-coupled receptor functional classes. *Journal of Computational Chemistry* 30, 1414–1423.
- Xiao, X., Wang, P., Chou, K.C., 2011a. GPCR-2L: predicting G protein-coupled receptors and their types by hybridizing two different modes of pseudo amino acid compositions. *Molecular Biosystems* 7, 911–919.
- Xiao, X., Wang, P., Chou, K.C., 2011b. Quat-2L: a web-server for predicting protein quaternary structural attributes. *Molecular Diversity* 15 (1), 149–155.
- Xu, Y., Wang, X.-B., Ding, J., Wuc, L.-Y., Deng, N.-Y., 2010. Lysine acetylation sites prediction using an ensemble of support vector machine classifiers. *Journal of Theoretical Biology* 264, 130–135.
- Yang, J., Hart, E., Tauschek, M., Price, G.D., Hartland, E.L., Strugnelli, R.A., Robins-Browne, R.A., 2008. Bicarbonate-mediated transcriptional activation of divergent operons by the virulence regulatory protein, RegA, from *Citrobacter rodentium*. *Molecular Microbiology* 2, 314–327.
- Zakeri, P., Moshiri, B., Sadeghi, M., 2011. Prediction of protein submitochondria locations based on data fusion of various features of sequences. *Journal of Theoretical Biology* 269, 208–216.
- Zeng, Y.H., Guo, Y.Z., Xiao, R.Q., Yang, L., Yu, L.Z., Li, M.L., 2009. Using the augmented Chou's pseudo amino acid composition for predicting protein submitochondria locations based on auto covariance approach. *Journal of Theoretical Biology* 259, 366–372.

5.3 CAPÍTULO III - NEURAL NETWORKS APPLIED TO BACTERIAL PROMOTER PREDICTION BASED ON DNA STABILITY

Este capítulo apresenta o artigo “*Neural Networks applied to bacterial promoter prediction based on DNA stability*” que será submetido para publicação em revista científica relacionada à área de Computação Aplicada. Neste artigo é descrita a metodologia de simulação de RN utilizando os dados de estabilidade da sequência promotora de acordo com o fato σ que a reconhece. Além da predição, foi realizada a extração de regras e este conhecimento será incorporado à implementação da ferramenta BacPP, melhorando a predição para as outras bactérias Gram-negativas.

Neural Networks applied to bacterial promoter prediction based on DNA stability

Scheila de A vila e Silva¹, Franciele Forte¹, Ivaine T. S. Sartor¹, Ana Paula Longaray Delamare¹, Günther J. L. Gerhardt², Sergio Echeverrigaray¹

¹ Universidade de Caxias do Sul, Instituto de Biotecnologia
Rua Francisco Getúlio Vargas, 1130 - CEP 95070-560
Caxias do Sul - RS - Brasil

Phone: 55 54 3218 21 00 extension 2075
Fax number 55 54 3218 21 49

² Universidade de Caxias do Sul, Departamento de Física e Química
Rua Francisco Getúlio Vargas, 1130 - CEP 95070-560
Caxias do Sul - RS - Brasil

Phone: 55 54 3218 21 00 extension 2075
Fax number 55 54 3218 21 49

{Scheila de A vila e Silva sasilva6@uacs.br, Franciele Forte, Ivaine T. S. Sartor, Ana Paula L. Delamare aplongar@gmail.com,
Günther Gerhardt gunther_lew@yahoo.com.br, Sergio Echeverrigaray selaguna@yahoo.com}

1 Introduction

The first and key step in gene expression process is the promoter sequence recognition by RNA polymerase enzyme (RNAP). These sequences are elements located before the transcription start site (TSS) of open reading frames (ORF) and they can play a regulatory role [1,2]. The proper regulation of transcription is crucial for a single-cell prokaryote since its environment can change dramatically and instantly [1,2,3]. In bacteria, RNAP holoenzyme consists of five subunits (2 α , β , β' , ω) and an additional sigma (σ) subunit factor. A collection of different σ subunits is responsible for lead RNAP binding on specific promoter regions and consequent activation of genes in response to environmental changes. The σ factors are labeled according to their molecular weight (σ^{24} , σ^{28} , σ^{32} , σ^{38} , σ^{54} and σ^{70}). Each σ factor has been assigned to a global function role, for instance, σ^{32} and σ^{24} play a role in heat-shock response, σ^{28} is associated with the expression of flagellar genes during normal growth, σ^{54} is involved in nitrogen metabolism and σ^{70} is the factor responsible for the bulk housekeeping transcriptional activity [3,4].

The canonical prokaryotic promoter has two consensual hexamers located before the first nucleotide transcribed (TSS): one centered at approximately 35 nucleotides (called -35 region) and another centered at 10 nucleotides (called -10 region or Pribnow Box), except for σ^{54} . These hexamers have

different consensual sequences, according to the σ factor which recognizes them. For instance, σ^{70} , σ^{32} , σ^{28} and σ^{54} has as consensus, respectively: TTGACA and TATAAT separated by 16-18 bp; CCCTTGAA and CCCGATNT separated by 13-15pb; CTAAA and GCCGATAA separated by 15pb; CTGGNA and TTGCA separated by 6pb [1,4].

The correct classification of a given DNA sequence as promoter or non promoter is an attractive research area in Bioinformatics because it improves genome annotation and allows generating hypotheses in the context of the bacterial transcription initiation process and gene function [5]. *In silico* promoter prediction techniques are attractive because they require less time and they are not so demanding when compared with molecular techniques [6]. Promoter compilation and analysis have led to computer programs which predict promoter sequences on basis on their homology or consensus sequences. Unfortunately, the predictive power of these algorithms is reduced due the frequent variations in consensus sequences position and nucleotide composition [2]. Many currently promoter prediction methods, many of which are applied just to σ^{70} -dependent promoter sequence, decide if the sequences are promoter or not by using Position weight matrices [4,7], SVM [2,6,8] or NN [9,10,11,12]. Besides the prediction, another interesting procedure is rule extraction from NN trained because it gives an explanation of how each decision is made [13]. Despite the importance of this procedure, it is not carried out by the NN approaches previously described [9,11,12].

Considering the role of the promoter in the open complex formation [3], structural features, such as stability, bendability and curvature are expected to distinguish promoter sequences from other genomic sequences. It is known that the eukaryotic promoter and prokaryotic σ^{70} -dependent promoter sequences have lower stability, higher curvature and lesser bendability than gene sequences [14]. Consequently, DNA stability is considered a promising parameter for promoter recognition [14, 15]. The relative stability of a DNA duplex depends on the identity of the nearest-neighbor bases [16,17], and different bioinformatic approaches using DNA duplex stability have been described [14,18,19].

In this paper, *E. coli* promoter sequence duplex stability was applied as input data in NN simulations. The approach carried out can be briefly described as follows: the stability was calculated by applying the Nearest Neighbors for the prediction of free energy, and the values obtained were

filtered using moving average. This data set was used for NN learning, and differently from previous reports, the simulations were carried out separately for six σ -dependent promoter of *E. coli*: σ^{70} , σ^{28} , σ^{38} , σ^{24} , σ^{32} , and σ^{54} . After the classification analysis, rule extractions were carried out to finding significant knowledge learned by the best NN architecture simulated for each σ factor promoter set.

2 Methods

2.1 Data Set

The positive examples were obtained from RegulonDB database [20] in its version available on April, 2010. A total of 1035 promoter sequences belonging to different *E. coli* σ factors were employed. Therefore, for σ^{24} , σ^{28} , σ^{32} , σ^{38} , σ^{54} and σ^{70} were obtained 69, 20, 69, 99, 38 and 740 sequences, respectively. The negative examples were randomly chosen from *E. coli* non-promoter intergenic regions in the same number and length of positive examples.

In order to carry out NN simulation the following procedure [14,17] was used:

$$\Delta G^0 = \Delta G^0_{ini} + \Delta G^0_{sym} \sum_{i=1}^{n-1} \Delta G^0_{ij+1} \quad (1)$$

According to [14], the terms ΔG^0_{ini} and ΔG^0_{sym} were not considered. The stability scores of all observed di-, tri-, tetra-, or dodeca- nucleotides in sliding overlapping windows (one nucleotide step) were calculated and averaged in order to evaluate the mean scores of individual promoter and non-promoter sequences [15].

The data sets were smoothed with a moving average using the software Low121 [21]. Aiming at finding the optimal number of iterations in the moving average, semi empirical test were carried out using four smoothing values: 3, 5, 8 and 12 degree.

2.2 Neural Network Simulation

The NN architecture used was a multi layer perceptron (MLP) with three layers: an input layer, a hidden layer, and an output layer. For each σ dependent-promoter sequence set, different architectures with the cross combination of the following parameters were tested:

1. Number of neurons in input layer: 80, 79, 78, 76, and 70. These numbers were obtained according to the size of the window (2, 3, 4, 6 and 12) applied on formula 1.

2. Number of neurons in hidden layer: 1 to 8.

3. Number of neurons in output layer: 1

R environment was used to carry out NN simulations [22]. The approach made use of a backpropagation algorithm [11]. The k-fold cross-validation methodology was chosen because it provides statistically valid results. At each iteration one of the k subsets was used as the test-set and the others formed a training set. The average error across all k trials was then computed [2]. The k values were 10 for σ^{70} promoters; 2 for σ^{28} and σ^{54} promoters; 3 for σ^{24} , σ^{32} and σ^{38} promoters, according to the number of promoter sequences available from the data set.

2.3 Results analysis

In this study, a known sequence predicted correctly was considered as true positive (TP) if it was a promoter, or as true negative (TN) if it was a non-promoter sequence. Incorrect predictions were denoted as false positive (FP) or false negative (FN). A promoter classified as non-promoter sequence was considered FN, and a non-promoter sequence classified as promoter, as FP. The cut-off value which classifies a sequence as promoter or non-promoter was determined by ROC curve analysis (Receiver Operator Characteristic curve). The ROC curve is a plot of sensitivity against specificity for different cut-off points. ROC curve is widely used because it provides a visual as well as numerical summary of a predictor's behavior [23,24].

The analysis of the results used the performance measures: Accuracy (A), Specificity (S) and Sensitivity (SN), which were calculated according to the following formulae:

$$A = \frac{TP + TN}{TN + TP + FN + FP} \quad (2)$$

$$S = \frac{TN}{TN + FP} \quad (3)$$

$$SN = \frac{TP}{TP + FN} \quad (4)$$

Specificity is the proportion of negative test sequences which are correctly classified, sensitivity is

the proportion of positive test sequences which are correctly classified, and accuracy is the proportion of correctly classified sequences of the entire data set [12].

2.4 Rule extraction

NN have been used during the last few decades in a wide variety of applications [11]. When their application is a decision support, such as in classification or clustering problems, it is often desirable to understand how the NN determines their own decision. The rule-extraction attempts to address this problem by making explicit the relationships between the input and output data [13,25].

The rules were extracted based on the value of the hidden neurons by the FAGNIS technique (Fuzzy Automatically Generated Neural Inferred System). This methodology has been successfully used to extract rules from promoters prediction [10], and from other biological problems by using NNs [26].

Briefly, FAGNIS technique is based on the idea that the activation functions (f_A) work on a small input range due to peculiarities of data. FAGNIS substitutes the activation functions for a set of linear segments. The value of f_A can be approximated by a set of linear segments, using a relationship as demonstrated in Equation 5:

$$f_A(a_j) \sim \sum_i [F_i(a_j) * (p_i * a_j + q_i)] \quad (5)$$

Where $f_A(a_j)$ is the original non-linear function, a_j is the activation signal (weighted sum of the input vector), $F_i(a_j)$ is a function that links each value of a_j to a correspondent linear segment(s), and $p_i * a_j + q_i$ are linear segments. To improve the precision of the approximation, $F_i(a_j)$ must be a fuzzy number.

The results are conveniently presented by rule prototypes, which are defined as the average model of the input data set. The rule can be written as a linear equation: “If $x \approx$ prototype then $y =$ constant of linear equation + (coefficients of the linear equation)”. Here, x is an input example, y corresponds to the NN output and the coefficients of the linear equation are the nucleotides of the sequence. Further explanation and details of FAGNIS can be obtained in [26].

3. RESULTS AND DISCUSSION

In the NN simulation, among all architectures tested, those which best classified the input data set

for each σ -dependent promoter are presented in Table 1. A greater number of neurons in the hidden layer did not increase accuracy, specificity or sensitivity values.

Table 1. The best architecture trained for each NN trained with the promoters according to the σ factor which recognize the sequence.

σ factor	Number of neurons in input layer	Number of iterations on Low121	Number of neurons in hidden layer	Number of neurons in output layer
σ 24	80	5	3	1
σ 28	80	3	4	1
σ 32	80	12	3	1
σ 38	80	12	5	1
σ 54	80	3	5	1
σ 70	80	5	4	1

The ROC curve for the best NN simulated for each σ -dependent promoter sequences is presented in Figure 2. It is possible to perceive the same mathematical behavior for all data set used. The ROC curves obtained did not show a typical behavior [23,24]. Nevertheless, the performance measures (Table 2) obtained with the chosen cut-off values (0.6 for σ^{28} and 0.5 for the other σ -dependent promoter sequences) showed satisfactory values when compared with the NN prediction literature [9,10,19]. The cut-off values chosen by ROC curve observation were.

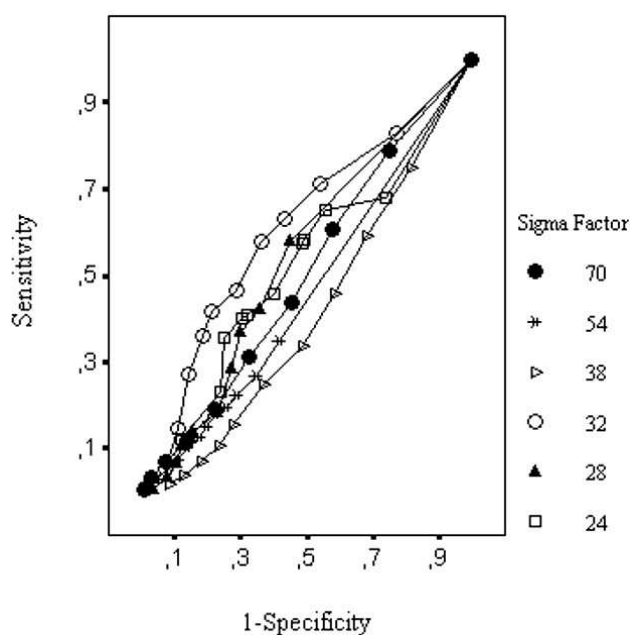


Figure 1. The ROC curve for the best architecture for each NN trained with the promoters according to the σ factor which recognize the sequence.

The best NN architecture for each σ -dependent promoter achieved the lower accuracy value of 56.52% for σ^{24} and the highest value of 80% for σ^{28} (Table 2). The specificity and sensitivity values

varied from 59.74% (σ^{24}) to 77.64% (σ^{54}), and 53.3% (σ^{24}) to 83% (σ^{28}), respectively. As an efficient classification tool, it is expected to recognize promoter as well non-promoters, the similar specificity and sensitivity values obtained for each σ -dependent promoter sequences are indicative of the consistency of the NN learning process. A comparison of the performance values obtained with NN simulations using stability and orthogonal input data (Table 2) showed that stability simulations were more efficient to predict σ^{28} , σ^{38} and σ^{54} dependent promoters. The σ^{70} -dependent promoters exhibited almost the same accuracy, specificity and sensitivity values for with both input data. Conversely, the performance values for σ^{24} and σ^{32} were lower than those obtained with orthogonal simulations. The low values obtained for by σ^{24} and σ^{32} cannot be explained by the NN prototypes (Figure 1) or motifs content, since there is no relation between the consensual regions or stability profiles presented by these sequences [4]. In general, the results obtained point out the importance of the σ factor separation in prediction approaches, since the overall accuracy for both stability (71%) and orthogonal data (72%) were lower than those obtained for individual σ -dependent promoter sequences.

Table 2. The performance measures for the best NN architecture presented in Table 1. In bold the best results when compared with the orthogonal simulation presented by [10].

σ factor	Accuracy (%)		Specificity (%)		Sensitivity (%)	
	Stability codification	Orthogonal codification	Stability codification	Orthogonal codification	Stability codification	Orthogonal codification
σ^{24}	56.5	71.6	59.7	69.1	53.3	73.9
σ^{28}	80.2	70.2	83.0	66.6	77.4	73.8
σ^{32}	64.1	72.4	52.5	71.4	75.5	73.4
σ^{38}	70.82	67.7	75.19	68.5	71.04	66.8
σ^{54}	78.82	73.5	80.0	73.2	77.64	73.8
σ^{70}	76.8	77.0	73.6	76.7	80.2	77.2

The literature presents many efforts to solve the bacterial promoter prediction problem. We carried out a comparison of the results achieved with previous approaches using DNA stability information. To our satisfaction, the NN results are comparable to related papers published thus far in the context of σ^{70} -dependent promoter sequences, once these papers are applied just for these sequences.

Ranganann and Bansal [14] developed their own method for promoter prediction by using DNA duplex stability and obtained modest accuracy of 52.2% with sensitivity of 99%. Promoters stability values were also used by [18, 19] as NN input data. Askary *et al.* [18] used an orthogonal codification for stability values on their predictions of σ^{70} -dependent promoters, and achieve 94% for sensitivity

and specificity. Using the nearest neighbors stability values for all the promoters from the RegulonDB, [19] obtained accuracy, specificity and sensitivity of approximately 70%.

As the performance parameter obtained by NN indicated good precision, the NN prototypes were extracted by the FAGNIS algorithm, in order to understand the most important features used during the learning process. This procedure was carried out for each σ -dependent sequence, and the results are shown in Figure 2. Rules were extracted for both promoter and random sequences. Except for σ^{70} , only one rule was extracted for each σ -dependent promoters.

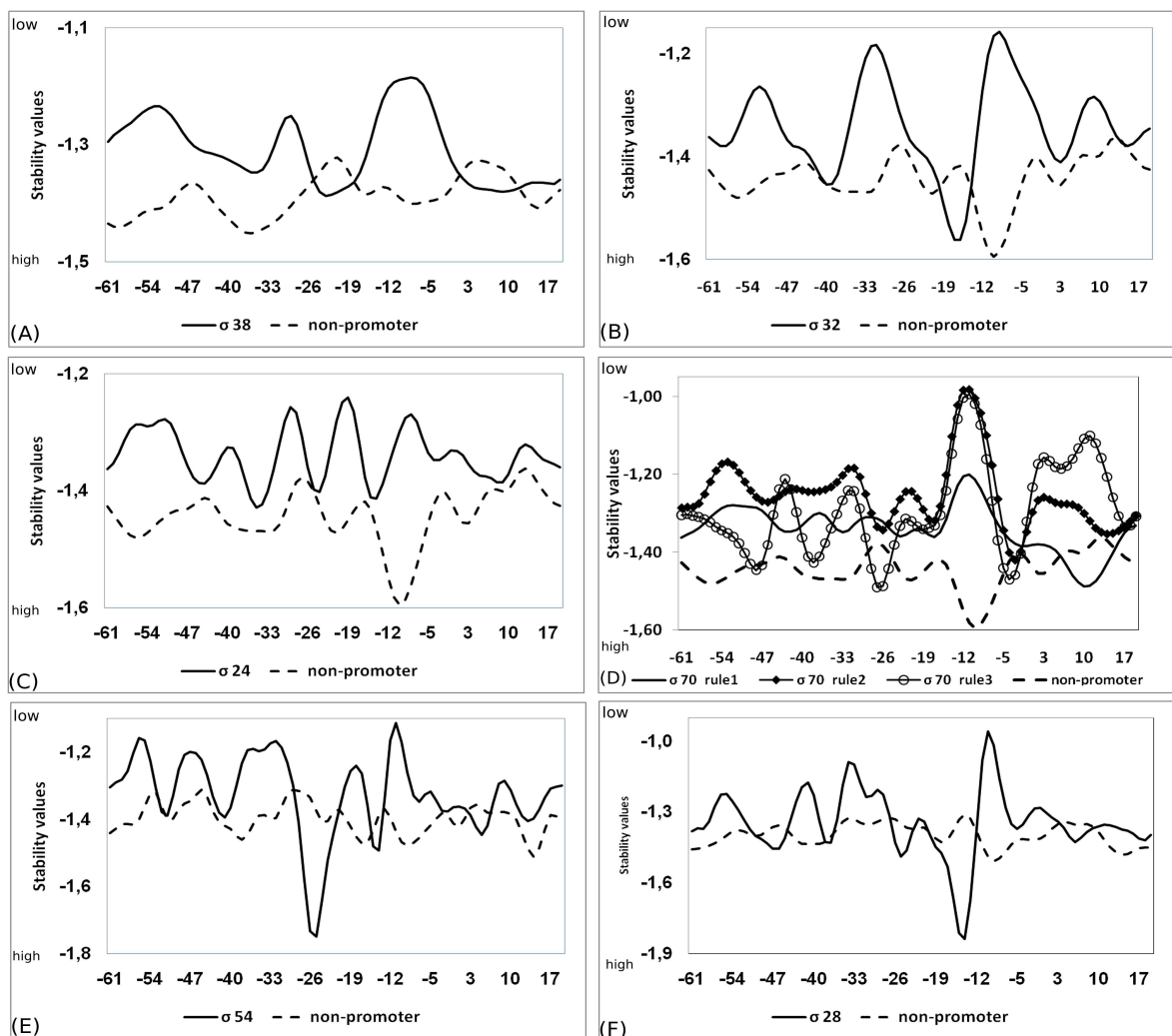


Figure 2. The prototypes extracted for each NN trained with the promoters according to the σ factor which recognize the sequence.

Considering the stability profiles presented by the rules extracted from NN (Figure 2), it is possible to perceive, some similarities and differences in the promoter sequences. In general, a clear difference was detected between promoter and non-promoter prototypes. Moreover, each σ -dependent promoter

gave a particular profile. For instance, some sequences presented several regions with high stability variation across the sequence (σ^{24} , σ^{28} σ^{54} and rule 1 of σ^{70}), whereas others (σ^{32} , σ^{38} and rules 2 and 3 of σ^{70}) showed highlighted lower stability in specific regions along the sequence. Despite these differences, most prototypes exhibited lower stability values near the conserved motifs (-25 and -10 region), as previously reported by [14]. The low stability around -10 region is expected, since this is the DNA melting region where transcription bubble is formed [1,3]. In order to extract information from the prototypes, principal components analysis (PCA) was carried out (Figure 3). A jointly analysis of Table 3, and Figures 2 and 3, was used to explain the stability features of the prototypes.

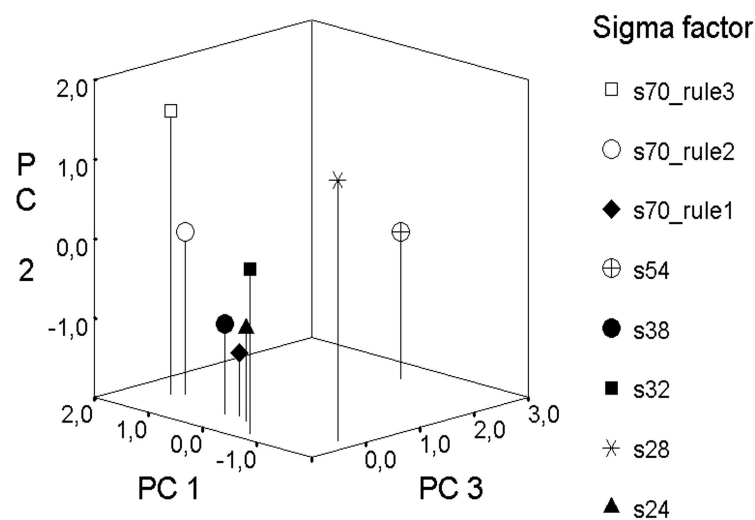


Figure 3. The PCA analysis for the promoter sequences belonging for the prototypes.

The first PCA component (which explained 38.6% of the variance) allowed to separate σ^{28} from the other sequences, and grouped the sequences belonging to rule 2 and 3 of σ^{70} . This separation was associated to high and low stability values in the -12 to -16 region of σ^{28} and σ^{70} (rules 2 and 3), respectively. The peculiar stability properties of the region that precedes the -10 consensual motif may be implicated in promoters recognition by σ^{28} and σ^{70} factors.

The second component (22.5% of the variance) separated the σ^{28} and σ^{70} (rule 3) prototypes from the others. These sequences were characterized by high stability in the -41 to -44 region, intermediate values for the -31 to -35 and +9 to +12 regions, and low stability in the -8 to -11 region, the consensual motif [4]. The third PCA component (18.2% of the variance) separated σ^{54} prototype from the others, and was associated to high stability (-1.55) of σ^{54} promoters on the -23 to -27 region (Table 3). This

finding is in accordance to biological background since, this high stability region matches one of the consensual motifs (defined as -24) of σ^{54} promoters [27].

Table 3. The average stability values for the most important regions according to the PCA analysis.

σ Factor	Component 1			Component 2			Component 3
	Stability of -12 to -16	Stability of -46 to -51	Stability of -41 to -44	Stability of -31 to -35	Stability of -8 to -11	Stability of +9 to +12	Stability of - 23 to -27
σ^{24}	-1.36	-1.32	-1.36	-1.37	-1.29	-1.35	-1.34
σ^{28}	-1.51	-1.41	-1.25	-1.21	-1.11	-1.36	-1.34
σ^{32}	-1.41	-1.32	-1.41	-1.24	-1.18	-1.31	-1.34
σ^{38}	-1.26	-1.27	-1.31	-1.31	-1.18	-1.37	-1.34
σ^{54}	-1.26	-1.29	-1.33	-1.19	-1.22	-1.33	-1.55
σ^{70} rule 1	-1.27	-1.29	-1.34	-1.32	-1.23	-1.47	-1.3
σ^{70} rule 2	-1.13	-1.23	-1.24	-1.2	-1.04	-1.31	-1.34
σ^{70} rule 3	-1.15	-1.37	-1.26	-1.29	-1.07	-1.12	-1.4

4. Conclusions

In this paper, we have presented an approach for prediction and classification of *E. coli* promoters based on NN trained with the stability of the sequence. By separating the promoter sequences according the σ factor which recognize them, we have demonstrated that the stability values can be used for prediction and classification of promoters. The boundaries classification obtained were (σ^{24} , σ^{28} , σ^{32} , σ^{38} , σ^{54} and σ^{70} accuracies of 56.5%, 80%, 61.9%, 70.8%, 78.8% and 76.5%, respectively). In contrast to the most of tools previously reported in the literature, this paper is not only applied to identification of σ^{70} -dependent promoters, but it is carried out to promoter classification according to its related σ factor. The satisfactory performance values obtained in the present work, indicates that the information generated by the NN learning (rules extracted from NN trained) can be used to increase accuracy, specificity and sensitivity of promoter prediction programs based on sequence data, as BacPP tool [10].

References

- [1] I. Hook-Barnard, X. B. Johnson, D. M. Hinton, *Escherichia coli* RNA polymerase enzyme of σ^{70} - dependent promoter requiring a -35 DNA element and an extended -10 TGn motif, *Journal of Bacteriology* 188 (2006) 8352-8359.
- [2] K. Polate, S. Günes, A novel approach to estimation of *E. coli* promoter gene sequences: Combining feature selection and least square support vector machine (FS_LSSVN), *Applied Mathematics and Computation* 190 (2007) 1574-1582.
- [3] J. L. Huffmann, R. G. Brennan, Prokaryotic transcription regulators: more than just the helix-turn-helix motif, *Current Opinion in Structural Biology* 12 (2002) 98-106.
- [4] E. Potvin, F. Sanschagrin, R. C. Levesque, Sigma factors in *Pseudomonas aeruginosa* Fed. of Eur. Microb. Soc. 32 (2008) 38-55.
- [5] P-E Jacques, S. Rodrigue, L. Gaudreau, J. Goulet, R. Brzezinski, Detection of prokaryotic promoters from the genomic

- distribution of hexanucleotides pairs, *BMC Bioinformatics* 7:423 (2006).
- [6] M. Towsey, P. Timms, J. Hogan, S. A. Mathews, The cross-species prediction of bacterial promoters using a support vector machine, *Computational Biology and Chemistry* 32 (2008) 359-366.
- [7] Q-Z LI, H. Lin, The recognition and prediction of σ^{70} promoters in *Escherichia coli* K-12, *Journal of Theoretical Biology* 242 (2006) 135–141.
- [8] L. Gordon, A. Chervonenkis, A. J. Gammernan, I. A. Shahmuradov, V. V. Solovyev, Sequence alignment for recognition of promoter regions, *Bioinformatics* 19 (15) (2003) 1964-1971.
- [9] S. Burden, Y.-X Lin, R. Zhang, Improving promoter prediction for the NNPP2.2 algorithm: a case study using *Escherichia coli* DNA sequences, *Bioinformatics* 21 (5) (2005) 601-607.
- [10] S. de Avila e Silva, S. Echeverrigaray, G. J. L. Gerhardt, BacPP: Bacterial promoter prediction—A tool for accurate sigma-factor specific assignment in enterobacteria, *Journal of Theoretical Biology* 287 (2011) 92-99.
- [11] R. N. Kalate, S. S. Tambe, B. D. Kulkarni, Artificial neural networks for prediction of mycobacterial promoter sequences, *Computational Biology and Chemistry* 27 (2003) 555-564.
- [12] T.S. Rani, S.D. Bhavani, R.S. Bapi, Analysis of *E. coli* promoter recognition problem in dinucleotide feature space, *Bioinformatics* 23(5) (2007) 582-588.
- [13] K. Odajima, Y. Hayashi, G. Tianxia, R. Setiono, Greedy rule generation from discrete data and its use in neural network rule extraction, *Neural Networks* 21 (2008) 1020-1028.
- [14] V. Rangannan, M. Bansal, Identification and annotation of promoter regions in microbial genome sequences on the basis of DNA stability, *Journal of Biosciences* 32(5) (2007) 851-862.
- [15] P. Akan, P. Deloukas, DNA sequence and structural properties as predictors of human and mouse promoters, *Gene* 410 (1) (2008) 165-176.
- [16] K. J. Breslauer, R. Frank, H. Blocker, L. A. Marky, Predicting DNA duplex stability from the base sequence, *Proc. Natl. Acad. Sci.* 83 (1986) 3746-3750.
- [17] J. SantaLucia, D. Hicks, The thermodynamics of DNA structural motifs, *Annu. Rev. Biophys. Biomol. Struct.* 33 (2004) 415-440.
- [18] A. Askary, A. Masoudi-Nejad, R. Sharafi, A. Mizbani, S. N. Parizi, M. Purmasjedi, N4: A precise and highly sensitive promoter predictor using neural network fed by nearest neighbors, *Genes & Genetic Systems* 84 (6) (2009) 425-430
- [19] S. de Avila e Silva, G. J. L. Gerhardt, S. Echeverrigaray, Rules extraction from neural networks applied to the prediction and recognition of prokaryotic promoters, *Genetics and Molecular Biology* 34:2 (2011) 353-360.
- [20] S. Gama-Castro, V. Jimenez-Jacinto, M. Peralta-Gil, A. Santos-Zavaleta, M. I. Peñaloza-Spinola, B. Contreras-Moreira, J. Segura-Salazar, L. Muñiz-Rascado, I. Martinez-Flores, H. Salgado, C. Bonavides-Martinez, C. Abreu-Goodger, C. Rodríguez-Penagos, J. Miranda-Ríos, E. Morett, E. Merino, A. M. Huerta, L. Treviño-Quintanilla, J. Collado-Vides, RegulonDB (version 6.0): gene regulation model of *Escherichia coli* K-12 beyond transcription, active (experimental) annotated promoters and Text press navigation, *Nucleic Acids Research* 36 (2008) D120-D124.
- [21] R. Hegger, H. Kantz, T. Schreiber, Practical implementation of nonlinear time series methods: The TISEAN package, *CHAOS* 9 (1999) 413-435
- [22] R Development Core Team, R: A language and environment for statistical computing, R Foundation for Statistical Computing (2008) Vienna, Austria. URL: <http://www.R-project.org>.
- [23] R.J. Irwin, T. C. Irwin TC, A principled approach to setting optimal diagnostic thresholds: where ROC and indifference curves meet, *Eur. J. Intern. Med.* 22(3) (2011) 230-234.
- [24] P. Sonego, A. Kocsor, S. Pongor, ROC analysis: applications to the classification of biological sequences and 3D structures, *Briefings in Bioinformatics* 9 (3) (2008) 198-209
- [25] R. Andrews, J. Diederich, A. B. Tickle, A survey and critique of techniques for extracting rules from trained artificial neural networks, *Knowledge-Based Systems* 8(6) (1995) 373-389.
- [26] E. Battistella, A. L. Cechin, The Protein Folding Problem Solved by a Fuzzy Inference System Extracted from an Artificial Neural Network, *Lecture Notes in Computer Science* 3315 (2004) 474-483.
- [27] H. Barrios, B. Valderrama, E. Morett, Compilation and analysis of σ^{54} -dependent promoter sequences, *Nucleic Acids Research* 27 (22) (1999) 4305-4313.

5.4 CAPÍTULO IV – BANCO DE DADOS INTERGENICDB

Este capítulo apresenta o banco dados modelado pelo aluno de graduação do curso de Bacharelado em Ciência da Computação, Aurione Molin, sob a orientação da professora Dr^a. Helena Graziottin Ribeiro e co-orientação de Scheila de Avila e Silva. Este banco de dados teve sua interface implementada pela aluna de graduação do curso de Bacharelado em Ciência da Computação Vanessa Davanzo, sob a orientação do professor Msc. Daniel Luis Notari e co-orientação de Scheila de Avila e Silva. Os professores atuam como pesquisadores no Núcleo de Pesquisa em Bioinformática da Universidade de Caxias do Sul e estão vinculados ao Centro de Computação e Tecnologia da Informação.

1 BANCO DE DADOS: “INTERGENICDB”

O banco de dados "Intergenicdb" foi criado para auxiliar no estudo de sequências promotoras de DNA de organismos procaríotos. Para que isso fosse possível, fez-se necessária a criação de um banco de dados para o armazenamento das regiões intergênicas identificadas como promotoras. Atualmente não há um banco de dados disponível na comunidade científica que atenda aos requisitos solicitados pelos pesquisadores do núcleo de pesquisa de Bioinformática da Universidade de Caxias do Sul.

Os requisitos do banco de dados foram coletados por MOLIN (2009). Os dados armazenados no banco foram retirados de Banco de Dados de Biologia Molecular (BDBM) e artigos científicos. Os BDBMs utilizados no projeto são o CRM, o NCBI e o RegulonDB. As informações extraídas foram armazenadas em um arquivo de texto e, posteriormente carregados no banco de dados "promotoresdb".

A partir das necessidades apresentadas foi feita a modelagem do protótipo do banco de dados. Os requisitos levantados foram os seguintes (MOLIN, 2009):

- Cada região promotora está ligada a um organismo, que possui uma classificação biológica de acordo com o NCBI
- Cada gene possui um nome, um símbolo, um número de posição (conforme NCBI) que indica o seu de início e um para seu fim, uma função e uma quantidade percentual de CG.

- Cada região intergênica possui um número para a posição inicial e final na sequência, um tamanho, sua sequência de nt.
- A região intergênica de teste é uma fonte de dados usada para predição que define se a sequência é ou não candidata a ser promotora.
- Um método preditor é um algoritmo usado na predição das chances de uma região ser ou não promotora, e possui um nome, um percentual de uma cadeia analisada e uma descrição de suas características de funcionamento.
- Uma publicação é um documento relacionado à área de estudo de promotores, que possui um autor, um título, uma data de publicação, um instituto ao qual está vinculado, uma cidade e um país, um endereço web e um espaço para informações adicionais.

1.1 Modelo Conceitual e Esquema Lógico do Banco de Dados

A Ilustração 1 mostra o modelo conceitual criado para representar os requisitos em entidades, atributos e relacionamentos.

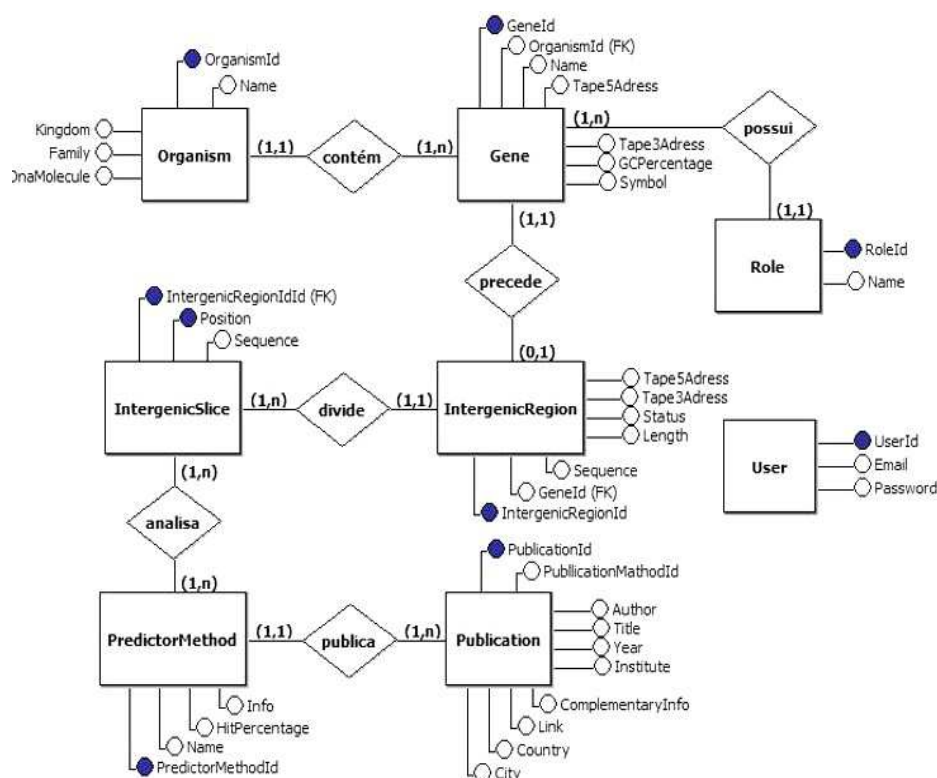


Ilustração 1: Modelo Conceitual do “intergenicDB”

A partir do modelo conceitual, foi criado o modelo lógico (Ilustração 2). As tabelas e seus atributos foram construídos da seguinte forma:

- Tabela *Organism*
 - *OrganismId* é um número inteiro e é a chave primária da tabela
 - *Kingdom* é uma *string* de tamanho 32, com valor não nulo
 - *Family* é uma *string* de tamanho 64, com valor não nulo
 - *Name* é uma *string* de tamanho 100, com valor não nulo
 - *DnaMolecule* é uma *string* de tamanho 100, com valor não nulo

- Tabela *Gene*
 - *GeneId* é um número inteiro e é a chave primária da tabela
 - *OrganismId* é um número inteiro e é a chave estrangeira que estabelece o relacionamento com a tabela *Organism*
 - *Name* é uma *string* de tamanho 100, com valor não nulo
 - *Symbol* é uma *string* de tamanho 45, com valor não nulo
 - *MainRole* é uma *string* de tamanho 45, com valor não nulo
 - *Tape5Adress* é um número inteiro, com valor não nulo
 - *Tape3Adress* é um número inteiro, com valor não nulo
 - *GCPpercentage* é um número do tipo *float*, com valor não nulo

- Tabela *IntergenicRegion*
 - *GeneId* é um número inteiro e é a chave primária da tabela, estabelece o relacionamento com a tabelam *Gene*
 - *Tape5Adress* é um número inteiro, com valor não nulo
 - *Tape3Adress* é um número inteiro, com valor não nulo
 - *Sequence* é uma *string* de tamanho 7000, com valor não nulo
 - *Length* é um número inteiro, com valor não nulo
 - *Status* é um *char*, com valor não nulo

- Tabela *IntergenicSlice*
 - *IntergenicSliceId* é um número inteiro e é a chave primária da tabela
 - *GeneId* é um número inteiro e é a chave estrangeira que estabelece o relacionamento com a tabela *IntergenicRegion*
 - *Sequence* é uma *string* de tamanho 60 (mínimo), com valor não nulo

- Tabela *PredictorMethod*
 - *PredictorMethodId* é um número inteiro e é a chave primária da tabela
 - *Name* é uma *string* de tamanho 100, com valor não nulo
 - *HitPercentage* é um número do tipo *float*, com valor não nulo
 - *Info* é uma *string* de tamanho 500, com valor não nulo

- Tabela *PredictorMethodor_IntergenicSlice*
 - *PredictorMethodId* é um número inteiro e é faz parte da chave primária composta que estabelece o relacionamento entre as tabelas *IntergenicSlice* e

PredictorMethod

○ *IntergenicSliceId* é um número inteiro e é faz parte da chave primária composta que estabelece o relacionamento entre as tabelas *IntergenicSlice* e *PredictorMethod*

- Tabela *Publication*

- *PublicationId* é um número inteiro e é a chave primária da tabela
- *PredictorMethodId* é um número inteiro e é a chave estrangeira que estabelece o relacionamento com a tabela *PredictorMethod*
- *Author* é uma *string* de tamanho 100, com valor não nulo
- *Title* é uma *string* de tamanho 200, com valor não nulo
- *Year* é do tipo *data*, com valor não nulo
- *Institute* é uma *string* de tamanho 300, com valor não nulo
- *City* é uma *string* de tamanho 50, com valor não nulo
- *Country* é uma *string* de tamanho 45, com valor não nulo
- *Link* é uma *string* de tamanho 255, com valor não nulo
- *ComplementaryInfo* é uma *string* de tamanho 500, com valor não nulo

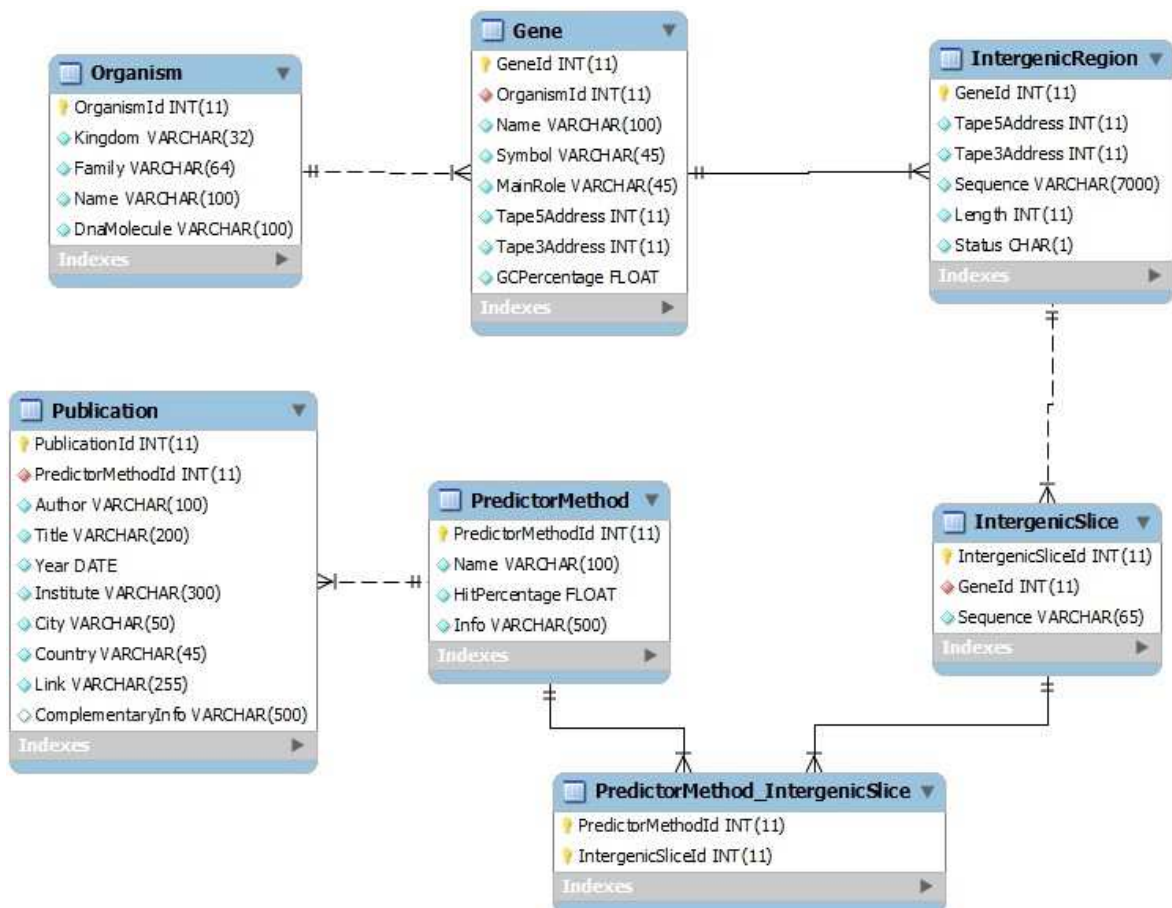


Ilustração 2: Modelo Lógico do "IntergenicDB"

2 MODELAGEM DO PORTAL “INTERGENICDB”

1.2 Requisitos do Sistema

O sistema “IntergenicDB” surgiu da necessidade de uma ferramenta que interagisse com o banco de dados “intergenicDB”. O objetivo do sistema é realizar buscas eficientes no banco de dados como também permitir que o usuário faça *download* e *upload* de arquivos de forma simples e intuitiva. Após a coleta de requisitos, foi possível a criação de um diagrama de caso de uso, que é mostrado na Ilustração 3.

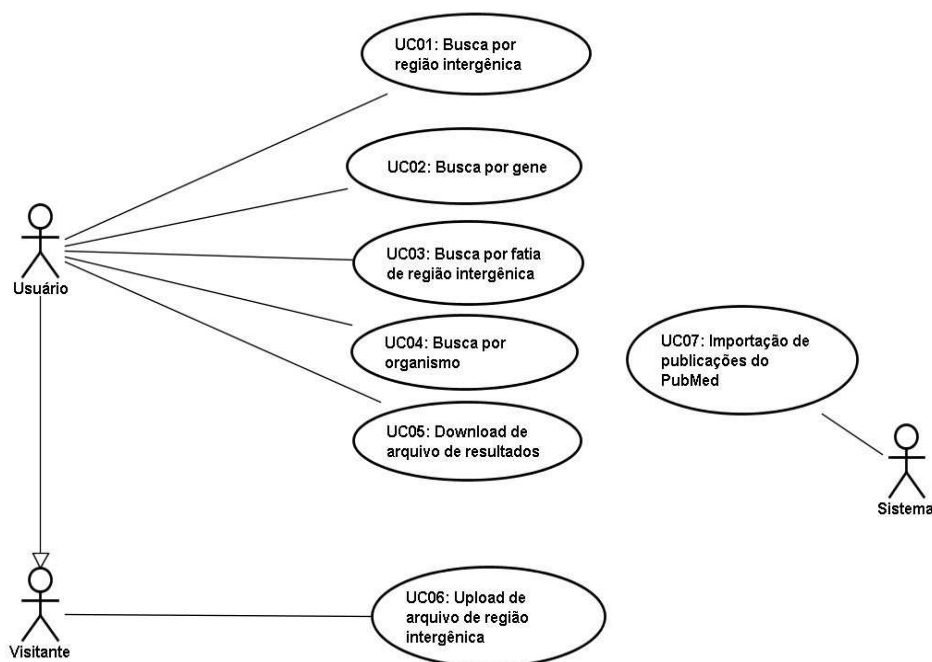


Ilustração 3: Diagrama de Caso de Uso do “IntergenicDB”

O diagrama de caso de uso mostra três atores: o usuário, o visitante e o sistema. O ator usuário poderá realizar as seguintes operações:

- *Buscar por região intergênica*: o usuário poderá efetuar buscas no sistema por região intergênica, sendo possível buscar por tamanho, faixa de tamanho e posição na fita;
- *Buscar por gene*: o usuário poderá efetuar buscas no sistema por gene, sendo possível buscar por nome, símbolo, função principal, e porcentagem de GC;

- *Buscar por organismo*: o usuário poderá efetuar buscas no sistema por organismo, sendo possível buscar por nome, reino e família;
- *Download de arquivo do resultado*: o usuário poderá efetuar *downloads* dos resultados de suas buscas em arquivos no formato TXT, XML e HTML.

O próximo ator é o visitante, que também é um usuário, mas que possui permissão para fazer o *upload* de arquivos. Essa diferenciação foi feita para que apenas usuários cadastrados possam executar essa ação. Por fim, o ator sistema poderá fazer a importação de publicações do portal PubMed para o “IntergenicDB”.

1.3 Protótipo de Interface

A partir dos requisitos e do diagrama de caso de uso, foi montado o protótipo de interface. Sua navegação é feita por abas, sendo elas “Home”, “Busca”, “Ajuda” e “Sobre”. Se o usuário for cadastrado para executar *uploads*, uma quinta aba é adicionada, a aba “Uploads”. A Ilustração 4 mostra o protótipo de tela na aba “Busca”, que mostra as opções conforme os requisitos de busca (Davanzo, 2010).

Ilustração 4: Protótipo de interface do “IntergenicDB”

1.4 Modelagem Conceitual do Projeto de Interface

O “IntergenicDB” foi modelado de acordo com padrões de projeto largamente utilizados, como o *Model-View-Controller* (MVC), *Abstract Factory* e *Data Gateway* (Ilustração 5). O *Model-View-Controller* é um o padrão utilizado o desenvolvimento de interfaces há pelo menos 30 anos, sendo escolhido para o desenvolvimento das páginas do “IntergenicDB (Davanzo, 2010). O MVC surgiu no final dos anos 1970, em um framework desenvolvido por Trygve Reenskaug para a plataforma *SmallTalk*, mas sofreu diversas adaptações ao longo do tempo (FOWLER *et al.*, 2002). No entanto, sua principal característica segue inalterada: o MVC consiste em três tipos de objetos. O *Model* (Modelo) é o objeto da aplicação, a *View* (Visualização) é a representação do *Model* na tela e o *Controller* (Controlador) define a forma em que a interface reage aos estímulos do usuário (GAMMA *et al.*, 1995).

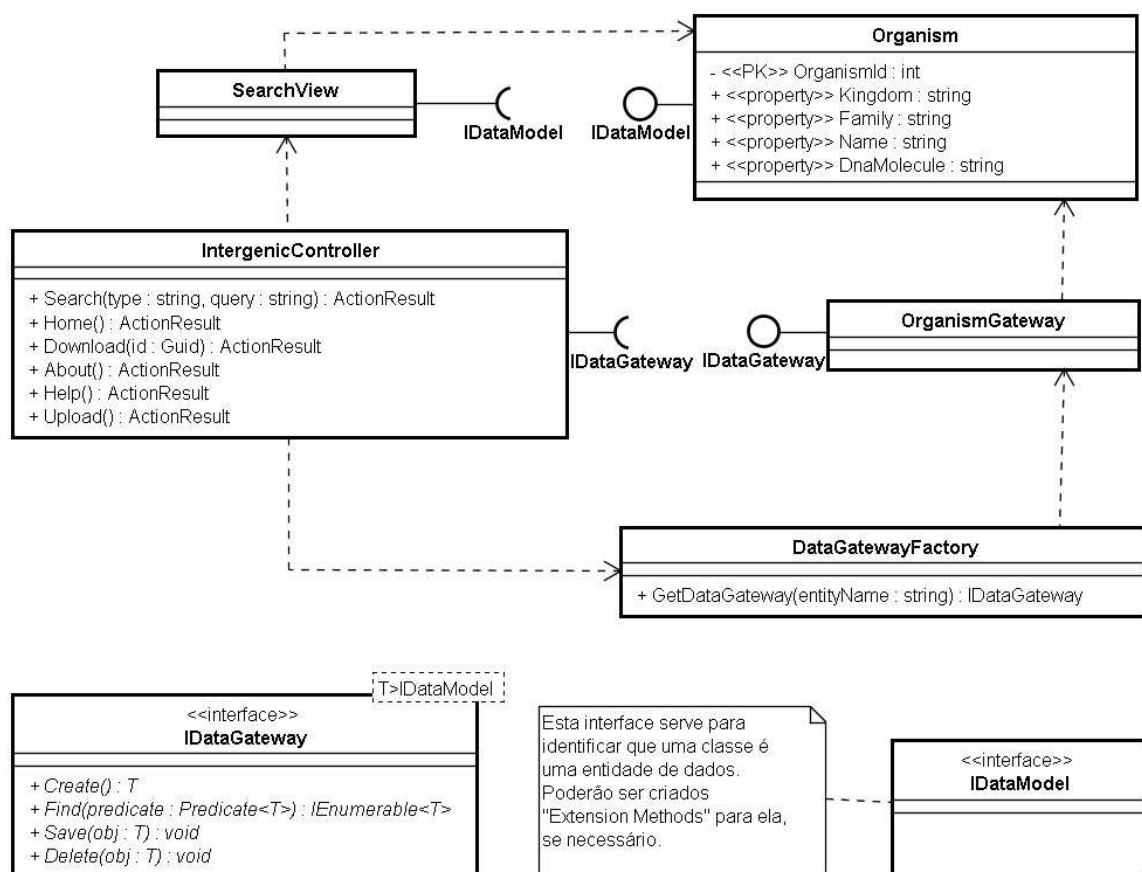


Ilustração 5: Síntese dos padrões de projeto utilizados

No “IntergenicDB”, o *Model* foi concebido usando dois outros padrões de projeto: o *Abstract Factory* e o *Data Gateway*. O primeiro provém uma interface para

a criação de famílias de objetos sem especificar suas classes concretas (GAMMA *et al.*, 1995), permitindo o desacoplamento do *Controller* do modelo de dados. Para tanto, a *Factory* recebe um parâmetro que identifica o tipo de operação a ser executada e retorna um objeto que respeita uma interface conhecida pelo *Controller*: *IDataGateway*.

A interface *IDataGateway* implementa o padrão de projeto *Data Gateway*: define as operações de criação, leitura, salvamento e remoção de registros na base de dados e centraliza as rotinas de acesso a dados em seu interior, permitindo a fácil localização, substituição e manutenção das rotinas de acesso a dados (FOWLER *et al.*, 2002). O uso de padrões não é importante apenas para resolver problemas conhecidos, mas também para definir claramente as atribuições de cada um dos objetos e criando grupos funcionais facilmente distinguíveis.

1.5 Arquitetura do Sistema

A arquitetura do sistema necessita de uma máquina servidora com um servidor *Web* com a linguagens ASP.NET e o SGBD (Sistema de Gerência de Banco de Dados) MySQL. O servidor receberá as requisições de serviços, as encaminhará para o *Controller*, que consultará o *Model* (modelo de dados) e retornará a página em que o usuário irá interagir com as informações armazenadas no banco de dados, a *View*.

1.6 Componentes do Sistema

O sistema “IntergenicDB” foi dividido em uma página inicial e cinco grupos de navegação. Os grupos são:

- *Busca*: nesta página o usuário poderá fazer buscas no portal. A página “Busca” contém todas as possibilidades de busca que o sistema oferece. Através desta página o usuário chegará à página de resultados, onde ele terá a opção de *download* em formato TXT e XML;
- *Ajuda*: nesta página o usuário poderá obter informações sobre o funcionamento do sistema e como utilizá-lo;
- *Sobre*: nesta página o usuário poderá obter informações sobre o projeto em que esse sistema está inserido, a instituição de ensino e etc;

- *Cadastro*: nesta página o usuário poderá se cadastrar no *site* para poder posteriormente fazer *upload* de arquivos;
- *Logon*: nesta página o usuário poderá efetuar o *logon* no sistema caso ele forneça usuário ou senha errados na página inicial. Depois de feito o *logon*, o usuário terá acesso à página de *upload*.

A página inicial do “IntergenicDB” teve um *menu* do tipo abas, que permitirá o acesso aos grupos citados acima. Além disso, os campos de usuário e senha estarão no topo da página, permitindo que o usuário faça o *logon* no sistema. A Ilustração 6 mostra o modelo de navegação do *site*.

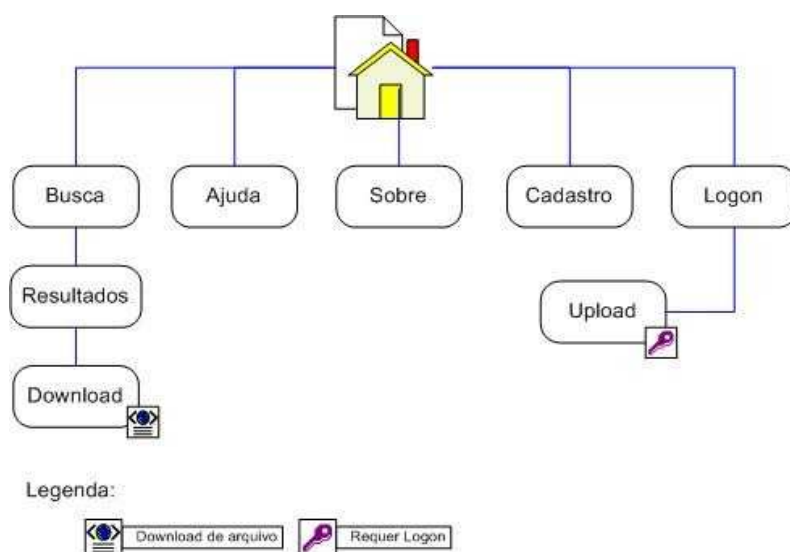


Ilustração 6: Modelo de navegação do “IntergenicDB”

1.7 Implementação

A tecnologia escolhida para a implementação do sistema foi o ASP.NET, utilizando a linguagem C#, pois esta combinação de linguagem e plataforma possibilita ganho de produtividade. A plataforma provê ferramentas de desenvolvimento poderosas, como o *Visual Studio 2010* (interface de desenvolvimento) e o LINQ (*Language Integrated Query*) que integra as rotinas de pesquisa e manipulação dos dados do SGBD a partir de estruturas da própria linguagem.

O sistema é compatível tanto com o .NET *framework* desenvolvido pela Microsoft, quanto com o *framework* de código aberto Mono, desenvolvido pela Novell

e compatível com Linux e Mac. No primeiro cenário, a aplicação foi executada sobre o IIS (*Internet Information Services*), o servidor *Web* distribuído com o *Windows Server*. No segundo cenário, o sistema foi executado sobre o servidor Apache, utilizando os módulos do Mono para a execução das páginas .NET em ambiente Posix (Linux, BSD, Unix e Mac OS X).

O SGBD utilizado no sistema foi o MySQL, pois é um banco de dados de código aberto, possui mecanismos de consulta de alto desempenho, capacidade de inserção de dados de forma rápida e suporte para funções *Web* especializadas (MOLIM, 2009). O MySQL possui uma versão gratuita que atende aos requisitos do sistema “IntergenicDB”, além de poder ser executado tanto na plataforma Windows quanto nas plataformas Posix.

A interface da aplicação foi desenvolvida utilizando HTML 5 e CSS 3, padrões internacionalmente aceitos e regulados pelo W3C (*World Wide Web Consortium*) e jQueryUI que provêm um conjunto amplo de ferramentas para o desenvolvimento de código JavaScript (executado no navegador do cliente), a fim de proporcionar uma experiência de navegação mais rica para o usuário.

1.8 Referências bibliográficas

Fowler, M.; Rice, D.; Foemmel, M.; Hieatt, E.; Mee, R.; Stafford, R. *Patterns of Enterprise Application Architecture*. Addison Wesley, 2002.

Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison Wesley, 1995.

Molin, A. F. Prototipagem de um Banco de Dados de Promotores como Base para um Portal de Serviços. Centro de Computação e Tecnologia da Informação, Universidade de Caxias do Sul: Caxias do Sul: 2009. Bacharelado em Ciência da Computação - Trabalho de Conclusão de Curso.

Davanzo, V. F. Desenvolvimento de consultas para um banco de dados de regiões promotoras. Centro de Computação e Tecnologia da Informação, Universidade de Caxias do Sul: Caxias do Sul: 2010. Bacharelado em Ciência da Computação - Trabalho de Conclusão de Curso.

6 CONSIDERAÇÕES FINAIS

A metodologia de RNs possibilitou (por meio da extração de regras a partir das arquiteturas treinadas) o desenvolvimento de uma nova ferramenta, chamada BacPP. Diferentemente dos trabalhos previamente descritos na literatura, esta ferramenta fornece a classificação de uma determinada sequência como promotora e a probabilidade de ser reconhecida pelos diferentes fatores σ de *E. coli*. Ou seja, esta ferramenta não se restringe apenas aos promotores reconhecidos pelo fator σ^{70} .

Os resultados mostrados pelo BacPP utilizando as sequências promotoras de *E. coli* foi melhor que a predição realizada pelas RNs que originaram os protótipos implementados na ferramenta. Além disso, quando comparados os resultados do BacPP com a literatura, os resultados são satisfatórios e competitivos.

Além da abordagem utilizando a codificação ortogonal, foram realizadas simulações nas quais os parâmetros de entrada para a RN foram os valores de estabilidade da sequência. Considerando os seis grupos de promotores, os resultados obtidos foram melhores do que aqueles apresentados pelas simulações utilizando codificação ortogonal. Os valores de sensibilidade foram melhores para cinco grupos (apenas as sequências reconhecidas pelo σ^{24} não mostraram melhora) e, a especificidade e acurácia foram melhores para três grupos (sequências reconhecidas pelos fatores σ^{28} , σ^{38} , σ^{54}). Estes resultados mostram o potencial de utilização das propriedades físicas da sequência (estabilidade, curvatura, maleabilidade) como parâmetro de classificação. Além disso, percebeu-se que as abordagens empregadas podem ser utilizadas em combinação e melhorar a predição realizada pelo BacPP.

As partir da análise das regras extraídas das RNs treinadas foi possível verificar que, em geral, muitos protótipos obtidos em ambas simulações (codificação

ortogonal e com valores de estabilidade) apresentam informação compatível com o conhecimento biológico já estabelecido para os motivos -35 e -10. No entanto, percebeu-se que não é possível agrupar todos os promotores em um mesmo conjunto de treinamento porque os perfis de estabilidade e a composição de nt apresentaram grande variação conforme o fator σ que reconhece as sequências.

Procurando ampliar a informação para outras bactérias Gram-negativas, modelou-se e implementou-se um banco de dados com informações sobre a estrutura e/ou composição das sequências intergênicas, bem como os valores de predição obtidos com o BacPP. Este banco, procura diminuir a falta de dados de promotores para outras bactérias Gram-negativas, além de *E. coli*.

7 REFERÊNCIAS BIBLIOGRÁFICAS

ANDREWS, R; DIEDERICH, J; TICKLE, A. B. A survey and critique of techniques for extracting rules from trained artificial neural networks. **Knowledge-Based Systems** 8: 373-389, 1995.

ASKARY, A., MASOUDI-NEJAD, A., SHARAFI, R., MIZBANI, A., PARIZI, S. N., PURMASJEDI, M. N4: A precise and highly sensitive promoter predictor using neural network fed by nearest neighbors. **Genes & Genetic Systems** 84: 425-430, 2009.

BALDI, P.; BRUNAK, S. **Bioinformatics: the machine learning approach**. 2. ed. Cambridge: MIT, 2001. 351 p.

BARRERA, J.; CESAR-Jr, R. M.; FERREIRA, J. E., GUBITOSO, M. E. An environment for knowledge discovery in biology. **Computers in Biology and Medicine** 34: 427-447, 2004.

BORUKHOV, S.; NUDLER, E. RNA polymerase: the vehicle of transcription. **Trends in Microbiology** 16: 126-134, 2008.

BURDEN, S.; LIN, Y.-X.; ZHANG, R. Improving promoter prediction for the NNPP2.2 algorithm: a case study using Escherichia coli DNA sequences. **Bioinformatics** 21: 601-607, 2005.

BURGESS, R. R.; ANTHONY, L. How sigma docks to RNA polymerase and what sigma does. **Current Opinion in Microbiology** 4: 126-131, 2001.

CECHIN, A. L.. **The extraction of Fuzzy Rules from Neural Networks**. 1998. 149 f. Tese (Doutorado em Informática) – Aachen: Shaker, [1998].

CLOETE, I.; ZURADA, J. M. **Knowledge-Based neurocomputing**. Cambridge: MIT, 2000. 486 p.

CROOKS, G.E.; HON, G.; CHANDONIA, J.M.; BRENNER, S.E. WebLogo: A sequence logo generator, **Genome Research** 14:1188-1190, 2004.

- COTIK, V.; ZALIZ, R. R.; ZWIR, I. A hybrid promoter analysis methodology for prokaryotic genomes. **Fuzzy Sets and Systems** 152: 83-102, 2005.
- DE ROBERTIS, E. D. P. **Bases da biologia celular e molecular**. 2. ed. Rio de Janeiro: Guanabara Koogan, 1993. 307 p.
- DEMELEER, B.; ZHOU, G. Neural network optimization for *E. coli* promoter prediction. **Nucleic Acids Research** 19: 1593-1599, 1991.
- GAMA-CASTRO, S.; JIMENEZ-JACINTO, V.; PERALTA-GIL, M.; SANTOS-ZAVALETA, A.; PEÑALOZA-SPINOLA, M. I.; CONTRERAS-MOREIRA, B.; SEGURA-SALAZAR, J.; MUÑIZ-RASCADO, L.; MARTINEZ-FLORES, I.; SALGADO, H.; BONAVIDES-MARTINEZ, C.; ABREU-GOODGER, C.; RODRÍGUEZ-PENAGOS, C.; MIRANDA-RÍOS, J.; MORETT, E.; MERINO, E.; HUERTA, A. M.; TREVIÑO-QUINTANILLA, L.; COLLADO-VIDES, J. RegulonDB (version 6.0): gene regulation model of Escherichia coli K-12 beyond transcription, active (experimental) annotated promoters and Textpresso navigation. **Nucleic Acids Research** 36: D120-D124, 2008.
- GOÑI, J.R.; PEREZ, A.; TORRENTS, D.; OROZCO, M. Determining promoter location based on DNA structure first-principles calculations. **Genome Biology** 8: R 263, 2007.
- GORDON, L.; CHERVONENKIS, A. Y.; GAMMERMAN, A. J.; SHAHMURADOV, I. A.; SOLOVYEV, V. V. Sequence alignment for recognition of promoter regions. **Bioinformatics** 19: 1964-1971, 2003.
- GORDON, J.J.; TOWSEY, M.; HOGAN, J.; MATHEWS, S. A. TIMMS, P.; Improved prediction of bacterial transcription start sites. **Bioinformatics** 22:142-148, 2006.
- HAYKIN, S. **Neural networks: a comprehensive foundation**. 2. ed. New Jersey: Prentice-Hall, 1999. 842 p.
- HEGGER R., KANTZ H., SCHREIBER T. Practical implementation of nonlinear time series methods: The TISEAN package, **Chaos** 2: 413 – 435, 1999.
- HERTZ, G. Z.; STORMO, G. D. Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. **Bioinformatics** 15: 563-577, 1999.
- HOOK-BARNARD, I.; JOHNSON, X. B.; HINTON, D. M. Escherichia coli RNA Polymerase Recognition of a σ 70 – Dependent Promoter Requiring a -35 DNA Element and an Extended -10 TGn Motif. **Journal of Bacteriology** 188: 8352-8359, 2006.
- HORNIK, K. Multilayer feedforward networks are universal approximators. **Neural Networks** 2: 359-366, 1989.
- HOWARD, D.; BENSON, K. Evolutionary computation method for pattern recognition of cis-acting sites. **BioSystems** 72: 19-27, 2003.

- HUANG, S. H.; XING, H. Extract intelligible and concise fuzzy rules from neural networks. **Fuzzy Sets and Systems** 132: 233-243, 2005.
- JACQUES,P-E., RODRIGUES, S., GAUDREAU, L., GOULET, J., BRZEZINSKI, R. Detection of prokaryotic promoters from the genomic distribution of hexanucleotides pairs. **BMC Bioinformatics** 7:423 , 2006.
- JÁUREGUI, R; ABREU-GOODGER, C.; MORENO-HAGELSIEB, G., COLLADO-VIDES, J.; MERINO, E. Conservation of DNA curvature signals in regulatory regions of prokaryotic genes. **Nucleic Acids Research** 31: 6770-6777, 2003.
- KALATE, R. N.; TAMBE, S. S.; KULKARNI, B. D. Artificial neural networks for prediction of mycobacterial promoter sequences. **Computational Biology and Chemistry** 27: 555-564, 2003.
- KANHERE, A.; BANSAL, M. Structural properties of promoters: similarities and differences between prokaryotes and eukaryotes. **Nucleic Acids Research** 33:3165-3175, 2005a.
- KANHERE, A.; BANSAL, M. A novel method for prokaryotic promoter prediction based on DNA stability. **BMC Bioinformatics** 6:1, 2005b
- KIRYU, H.; OSHIMA, T.; KIYOSHI, A. Extracting relations between promoter sequences and their strengths from microarray data. **Bioinformatics** 21: 1062-1068, 2005.
- KOZOBAY-AVRAHAM,L.; HOSID, S.; VOLKOVICH, Z.; BOLSHOY, A. Prokaryote Clustering based on DNA curvature distributions. **Discrete Applied Mathematics** 11:2378-2387, 2008.
- LEHNINGER, A. L.; COX, M. M.; NELSON, D. L. **Principles of biochemistry**. 3. ed. New York: Worth, 2000. 1152 p.
- LEWIN, B. **Genes IX**. Porto Alegre: Artes Médicas, 2008. 955 p.
- LI, Q-Z., LIN, H. The recognition and prediction of $\sigma 70$ promoters in *Escherichia coli* K-12, **Journal of Theoretical Biology** 242: 135–141, 2006.
- MADIGAN, M. T. **Microbiologia de Brock**. 12.ed. Porto Alegre: Artmed, 2010. 1128 p.
- MAHADEVAN, I.; GHOSH, I. Analysis of *E. coli* promoter structures using neural networks. **Neural networks** 12: 717-725, 1999.
- MOUNT, D. W. **Bioinformatics : sequence and genome analysis**. New York: Cold Spring Harbor Laboratory, 2000. 564 p.
- MÜNCH, R., HILLER, K., GROTE, A., SCHEER, M., KLEIN, J., SCHOBERT, M.; JAHN D. Virtual Footprint and PRODORIC: an integrative framework for regulon

prediction in prokaryotes. **Bioinformatics** 21: 4187-4189, 2005.

MURAKAMI, K. S.; MASUDA, S.; CAMPBELL, E. A.; MUZZIN, O.; DARST, S. A. Structural Basis of Transcription Initiation: An RNA polymerase holoenzyme-DNA complex. **Science** 296: 1285-1290, 2002.

NARYSHKIN, N.; REVYAKIN A.; KIM, Y. MEKLER, V. EBRIGHT, R. H. Structural organization of the RNA polymerase-promoter open complex. **Cell** 101: 601-611, 2000.

ODAJIMA, K.; HAYASHI, Y.; TIANXIA, G.; SETIONO, R. Greedy rule generation from discrete data and its use in neural network rule extraction. **Neural Networks** 21: 1020-1020, 2008.

O'NEILL, M. C. Training back-propagation neural networks to define and detect DNA-binding sites. **Nucleic Acids Research** 19: 313-318, 1991.

PANDEY, S.P.; KRISHNAMACHARI, A. Computational analysis of RNA Pol-II promoters sequences. **Biosystems** 83: 38-50, 2006.

PEDERSEN, A. G.; ENGELBRECHT, J. Investigations of Escherichia coli promoter sequences with artificial neural networks: New signals discovered upstream of the transcriptional start point. In: **Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology (ISMB-95)** 3: 292-299, 1995.

PETERSON, J.D.; UMayAM, L.A; DICKINSON, T.M.; HICKEY, E.K.; WHITE, O. The Comprehensive Microbial Resource. **Nucleic Acids Research** 29:123-125. 2001.

POLAT, K.; GÜNES, S. A novel approach to estimation of *E. coli* promoter gene sequences: Combining feature selection and least square support vector machine (FS_LSSVM). **Applied Mathematics and Computation** 190:1574–1582, 2007.

Python Software Foundation, 2009. URL <http://www.python.org>.

R Development Core Team. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria, 2005. URL <http://www.R-project.org>.

RAMPRAKASH, J.; SCWARZ, F. P. Identification and annotation of promoters regions in microbial genome sequences on the basis of DNA stability. **Journal of Biosciences** 32: 851-862, 2007.

RANGANNAN, V.; BANSAL, M. Energetic contributions to the initiation of transcription in *E. coli*. **Biophysical Chemistry** 138: 91-98, 2008.

RANI, T.S., BHAVANI, S.D., BAPI, R.S. Analysis of *E. coli* promoter recognition problem in dinucleotide feature space. **Bioinformatics** 23: 582-588, 2007.

RUMELHART, D. E; HINTON, G. E.; WILLIAMS, R. J Learning representations by

back-propagating errors. **Nature** 323: 533-536, 1986.

RUSSELL, S. J.; NORVIG, P. **Artificial intelligence: a modern approach**. 2.ed. New Jersey: Prentice Hall International, c2003. 1080 p.

SIVARAMAN, K.; SESHASAYEE, A. S. N.; SWAMINATHAN, K.; MUTHUKUMARAN, G.; PENNATHUR, G. Promoter addresses: revelations from oligonucleotide profiling applied to the *Escherichia coli* genome. **Theoretical Biology and Medical Modelling** 2: 20, 2005.

SHULTZABERGER, R. K.; CHEN Z., LEWIS, K. A.; SCHNEIDER, T. D. Anatomy of *Escherichia coli* σ^{70} promoters. **Nucleic Acids Research** 35: 771-788 2007.

SONG, W.; MAISTE, P. J.; NAIMAN, D. Q.; WARD, M. J. Sigma 28 promoter prediction in members of the Gammaproteobacteria. **FEMS Microbiology Letters** 271: 222-229, 2007.

SPSS. SPSS Inc. Disponível em www.spss.com. Último acesso em 08 de maio de 2006.

THIYAGARAJAN, S.; RAJAN, S.S.; GAUTHAM, N. Effect of DNA structural flexibility on promoter strength – molecular dynamics studies of *E.coli* promoter sequences. **Biochemical and Biophysical Research Communications** 341: 557-566, 2006.

TOLOKHONOV, I.; LANDICK, R. The Role of the Lid Element in Transcription by *E. coli* RNA Polymerase. **Journal of Molecular Biologic** 361: 644-658, 2006.

TOWSEY, M.; TIMMS, P.; HOGAN, J.; MATHEWS, S. A. The cross-species prediction of bacterial promoters using a support vector machine. **Computational biology and Chemistry** 32: 359-366, 2008.

WHEELER, D. L.; BARRET, T.; BENSON, D. A.; BRYANT, S. H.; CANESE, K.; CHETVERNIN, V.; CHURCH, D. M.; DICUCCIO, M.; EDGAR, R.; FEDERHEN, S.; GEER, L. Y.; HELMBERG, W.; KAPUSTIN, Y.; KENTON, D. L.; KHOVAYKO, O.; LIPMAN, D. J.; MADDEN, T. L.; MAGLOTT, J. O.; PRUITT, K. D.; SCHULER, G. D.; SCHRIML, L. M.; SEQUEIRA, E.; SHERRY, S. T.; SIROTKIN, K.; SOUVOROV, A.; STARCHENKO, G.; SUZEK, T. O.; TATUSOV, R.; TATUSOVA, T. A.; WAGNER, L.; YASCHENKO, E. Database resources of the National Center for Biotechnology Information. **Nucleic Acids Research** 36: D13-D21, 2008.

WU, C. H. Artificial neural networks for molecular sequence analysis. **Computers & Chemistry** 21: 237-256, 1997.

WU, C. H.; MCLARTY, J. W. **Neural networks and genome informatics**. New York: Elsevier, 2000. 205 p.

APÊNDICE 1 - PATENTE INTERNACIONAL DA FERRAMENTA BACPP

Este apêndice apresenta a documentação do depósito de patente realizada no *United States Patent and Trademark Office*. Como a descrição de como a ferramenta foi implementada é feita no capítulo II, aqui são apresentados os formulários e o código-fonte.

1. Documentação referente ao depósito de patente

Doc Code: OATH

Document Description: Oath or declaration filed

PTO/SB/01 (10-08)

Approved for use through 06/30/2010. OMB 0651-0032

U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

DECLARATION FOR UTILITY OR DESIGN PATENT APPLICATION (37 CFR 1.63) <input type="checkbox"/> Declaration Submitted With Initial Filing OR <input checked="" type="checkbox"/> Declaration Submitted after Initial Filing (surcharge (37 CFR 1.16 (f)) required)	Attorney Docket Number	049954.002
	First Named Inventor	de Avila e Silva
	<i>COMPLETE IF KNOWN</i>	
	Application Number	12/855366
	Filing Date	12 August 2010
	Art Unit	TBA
	Examiner Name	TBA

I hereby declare that: (1) Each inventor's residence, mailing address, and citizenship are as stated below next to their name; and (2) I believe the inventor(s) named below to be the original and first inventor(s) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

METHOD, SYSTEM AND APPARATUS TO PREDICT AND/OR RECOGNIZE AND/OR CLASSIFY BIOLOGICAL SEQUENCES

(Title of the Invention)

the application of which

is attached hereto

OR

was filed on (MM/DD/YYYY) as United States Application Number or PCT International

Application Number and was amended on (MM/DD/YYYY) (if applicable).

I hereby state that I have reviewed and understand the contents of the above identified application, including the claims, as amended by any amendment specifically referred to above.

I acknowledge the duty to disclose information which is material to patentability as defined in 37 CFR 1.56, including for continuation-in-part applications, material information which became available between the filing date of the prior application and the national or PCT international filing date of the continuation-in-part application.

Authorization To Permit Access To Application by Participating Offices

If checked, the undersigned hereby grants the USPTO authority to provide the European Patent Office (EPO), the Japan Patent Office (JPO), the Korean Intellectual Property Office (KIPO), and any other intellectual property offices in which a foreign application claiming priority to the above-identified application is filed access to the above-identified patent application. See 37 CFR 1.14(c) and (h). This box should not be checked if the applicant does not wish the EPO, JPO, KIPO, or other intellectual property office in which a foreign application claiming priority to the above-identified application is filed to have access to the application.

In accordance with 37 CFR 1.14(h)(3), access will be provided to a copy of the application-as-filed with respect to: 1) the above-identified application, 2) any foreign application to which the above-identified application claims priority under 35 USC 119(a)-(d) if a copy of the foreign application that satisfies the certified copy requirement of 37 CFR 1.55 has been filed in the above-identified US application, and 3) any U.S. application from which benefit is sought in the above-identified application.

In accordance with 37 CFR 1.14(c), access may be provided to information concerning the date of filing the Authorization to Permit Access to Application by Participating Offices.

This collection of information is required by 35 U.S.C. 115 and 37 CFR 1.63. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 21 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: **Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

DECLARATION — Utility or Design Patent Application

Claim of Foreign Priority Benefits

I hereby claim foreign priority benefits under 35 U.S.C. 119(a)-(d) or (f), or 365(b) of any foreign application(s) for patent, inventor's or plant breeder's rights certificate(s), or 365(a) of any PCT international application which designated at least one country other than the United States of America, listed below and have also identified below, by checking the box, any foreign application for patent, inventor's or plant breeder's rights certificate(s), or any PCT international application having a filing date before that of the application on which priority is claimed.

Prior Foreign Application Number(s)	Country	Foreign Filing Date (MM/DD/YYYY)	Priority Not Claimed	Certified Copy Attached?	
				YES	NO
			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Additional foreign application numbers are listed on a supplemental priority data sheet PTO/SB/02B attached hereto.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

DECLARATION — Utility or Design Patent Application

Direct all correspondence to:	<input checked="" type="checkbox"/> The address associated with Customer Number:	25,461	OR	<input type="checkbox"/> Correspondence address below
Name Laurence P. Colton				
Address				
City		State		ZIP
Country	Telephone 404-815-3681		Email lcolton@sgrlaw.com	
WARNING:				
<p>Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to the USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available. Petitioner/applicant is advised that documents which form the record of a patent application (such as the PTO/SB/01) are placed into the Privacy Act system of records DEPARTMENT OF COMMERCE, COMMERCE-PAT-7, System name: <i>Patent Application Files</i>. Documents not retained in an application file (such as the PTO-2038) are placed into the Privacy Act system of COMMERCE/PAT-TM-10, System name: <i>Deposit Accounts and Electronic Funds Transfer Profiles</i>.</p> <p>I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. 1001 and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.</p>				
NAME OF SOLE OR FIRST INVENTOR:		<input type="checkbox"/> A petition has been filed for this unsigned inventor		
Given Name (first and middle [if any]) Scheila			Family Name or Surname de Avila e Silva	
Inventor's Signature				Date
Residence: City Caxias do Sul	State	Country BR	Citizenship BR	
Mailing Address Rua Visconde de Pelotas, 2205/22, Pio X				
City Caxias do Sul	State	Zip 95020-500	Country BR	
<input checked="" type="checkbox"/> Additional inventors or a legal representative are being named on the _____ supplemental sheet(s) PTO/SB/02A or 02LR attached hereto.				

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

DECLARATION**ADDITIONAL INVENTOR(S)
Supplemental Sheet**

Page 4 of 4

Name of Additional Joint Inventor, if any		<input type="checkbox"/> A petition has been filed for this unsigned inventor	
Given Name (first and middle [if any])		Family Name or Surname	
Sergio Echeverrigaray		Laguna	
Inventor's Signature		Date	
Residence: City Paysandu	State	Country BR	Citizenship BR
Rodrigues Alves, 1763 Lourdes			
Mailing Address			
City Paysandu	State	ZIP 95076-670	Country BR
Name of Additional Joint Inventor, if any		<input type="checkbox"/> A petition has been filed for this unsigned inventor	
Given Name (first and middle [if any])		Family Name or Surname	
Gunther Johannes Lewezuk		Gerhardt	
Inventor's Signature		Date	
Residence: City Porto Alegre	State	Country BR	Citizenship BR
Travessa Ferreira de Abreu, 22/05 Santana			
Mailing Address			
City Porto Alegre	State	Zip 90040-260	Country BR
Name of Additional Joint Inventor, if any		<input type="checkbox"/> A petition has been filed for this unsigned inventor	
Given Name (first and middle [if any])		Family Name or Surname	
Inventor's Signature		Date	
Residence: City	State	Country	Citizenship
Mailing Address			
City	State	Zip	Country

This collection of information is required by 35 U.S.C. 115 and 37 CFR 1.63. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 21 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Acknowledgement Receipt

The USPTO has received your submission at **16:22:58** Eastern Time on **12-AUG-2010** .

\$ **527** fee paid by e-Filer via *RAM* with Confirmation Number: 2609.

eFiled Application Information

EFS ID	8208940
Application Number	12855366
Confirmation Number	9740
Title	METHOD, SYSTEM AND APPARATUS TO PREDICT AND/OR RECOGNIZE AND/OR CLASSIFY BIOLOGICAL SEQUENCES
First Named Inventor	Scheila de Avila e Silva
Customer Number or Correspondence Address	25461
Filed By	Laurence Peter Colton
Attorney Docket Number	049954.002
Filing Date	
Receipt Date	12-AUG-2010
Application Type	Utility under 35 USC 111 (a)

Application Details

Submitted Files	Page Count	Document Description	File Size	Warnings
pta-ar-fucs-049954-002-ads.pdf	4	Application Data Sheet	237909 bytes	⚠ WARNINGS
This is not an USPTO supplied ADS fillable form				
pta-ar-fucs-049954-002-1.pdf	27		1324120 bytes	◆ PASS
		Document Description	Page Start	Page End
		Specification	1	19
		Claims	20	23
		Abstract	24	24
		Drawings-other than black and white line drawings	25	27
fee-info.pdf	2	Fee Worksheet (PTO-875)	36854 bytes	◆ PASS

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

If you need help:

- *Call the Patent Electronic Business Center at (866) 217-9197 (toll free) or e-mail EBC@uspto.gov for specific questions about Patent e-Filing.*
- *Send general questions about USPTO programs to the [USPTO Contact Center \(UCC\)](#).*
- *If you experience technical difficulties or problems with this application, please report them via e-mail to [Electronic Business Support](#) or call 1 800-786-9199.*

Electronic Patent Application Fee Transmittal

Application Number:				
Filing Date:				
Title of Invention:	METHOD, SYSTEM AND APPARATUS TO PREDICT AND/OR RECOGNIZE AND/OR CLASSIFY BIOLOGICAL SEQUENCES			
First Named Inventor/Applicant Name:	Schella de Avila e Silva			
Filer:	Laurence Peter Colton			
Attorney Docket Number:	049954.002			
Filed as Small Entity				
Utility under 35 USC 111(a) Filing Fees				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
Utility filing Fee (Electronic filing)	4011	1	82	82
Utility Search Fee	2111	1	270	270
Utility Examination Fee	2311	1	110	110
Pages:				
Claims:				
Miscellaneous-Filing:				
Late filing fee for oath or declaration	2051	1	65	65
Petition:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Miscellaneous:				
Total in USD (\$)				527

2. Código-fonte do BacPP

```
### Esta parte do programa usa os arquivos com as ponderacoes gerados anteriormente #####  
## para dizer se uma sequencia eh ou nao promotora#####
```

```
##### Funcao que conta quantas vezes um valor de ponderacao aparece #####  
def contador(cont):
```

```
    if cont in dicSeq:  
        dicSeq[cont]=dicSeq[cont]+1  
    else:  
        dicSeq[cont]=1
```

```
#####
```

```
##### Funcao que ordena o dicionario da funcao anterior #####
```

```
lista=[]  
def ordena():
```

```
    lista=dicSeq.items()  
    lista.sort()
```

```
    return lista
```

```
#####
```

```
##### Funcao que determina o score de uma sequencia #####
```

```
def soma (lin):  
    linha_res=0
```

```
    for i in range(0,quant_seq2-1):
```

```
        linha_prob=dados_ent2[i].split(delimiter)  
        if ((linha[i]=='t') or (linha[i]=='T')):  
            linha_res=linha_res+float(linha_prob[0])  
            contador(linha_prob[0])
```

```
        if ((linha[i]=='a') or (linha[i]=='A')):  
            linha_res=linha_res+float(linha_prob[1])  
            contador(linha_prob[1])
```

```
        if ((linha[i]=='g') or (linha[i]=='G')):  
            linha res=linha res+float(linha_prob[2])
```

```
        if ((linha[i]=='c') or (linha[i]=='C')):  
            linha_res=linha_res+float(linha_prob[3].split('\n')[0])  
            contador(linha_prob[3])
```

```
    return linha_res
```

```
#####
```

```

#####
delimiter=' '
delimiter2='\t'

print ("\n")
print ("\n")
print ("#####")
print ("#####")
print ("###")
print ("###          Seja bem-vindo ao programa BacPP          ###")
print ("###      Este programa foi desenvolvido pelo grupo de Bioinformatica      ###")
print ("### da Universidade de Caxias do Sul e classifica uma determinada      ###")
print ("### sequencia como promotora ou nao de um gene e fornece a probabilidade ###")
print ("### da sequencia ser reconhecida por um determinado fator sigma      ###")
print ("###")
print ("#####")
print ("#####")
print ("\n")
print ("\n")

arq_ent=raw_input("Digite o nome do arquivo de entrada:")## le o arquivo com as sequencias a
#serem analisadas

ent=open(arq_ent,'r') ## le o arquivo com as sequencias a serem analisadas
dados_ent= ent.readlines()
dados_ent = [line3.strip() for line3 in dados_ent]
#####

#####
ent3 = [24,28,32,38,54,70]
aleat = [-1,5,-2,-1,2,2]
minimo = [4,48,7,2,41,5]
maximo = [39.105.42.20.81.221]
#####

probabi=[]

out=open('Result_prevePROB_'+ arq_ent+'_regrasNewINTERGENIC44444.txt','w')# gera o arquivo de saida
saida = str(out)
#####

#####
for sigma in range (0,6): # laco que le os valores de sigma e substitui no arquivo
    z= str(ent3[sigma])

    ent2=open('new_preve_prom'+z+'_regrasNew'+z+'.txt','r')#le o arquivo com a regra ponderada
# para todos os sigmas

    resultado=[]

    dados_ent2=ent2.readlines()

```

```

##### CALCULA O VALOR DA SEQUENCIA #####
for linha in dados_ent:
    quant_seq2=int(len(linha)-1)
    a=[]
    linha_res2=0
    dicSeq={}

    elem=ordena()
    resultado.append(soma(linha))

##### CALCULA A PROBABILIDADE DA SEQUENCIA #####

probabi_aux=[]
for linha2 in resultado:
    if ((linha2 <= aleat[sigma]) or (linha2 < minimo[sigma])):
        probabilidade = 0
    if (linha2>=minimo[sigma]):
        if (linha2>maximo[sigma]):
            probabilidade =99
        else:
            probabilidade = (linha2*100)/maximo[sigma]
    probabi_aux.append(int(probabilidade))
probabi.append(probabi_aux)

# Listas de probabilidades para cada sigma -----
m=probabi[0]#probabilidades do sigma 24
n=probabi[1]#probabilidades do sigma 28
o=probabi[2]#probabilidades do sigma 32
p=probabi[3]#probabilidades do sigma 38
q=probabi[4]#probabilidades do sigma 54
r=probabi[5]#probabilidades do sigma 70

out.write("Inicio"+delimiter2+"Fim"+delimiter2+ "Sequencia analisada"+delimiter2)
out.write("S24"+delimiter2+"S28"+delimiter2+"S32"+delimiter2+"S38"+delimiter2+"S54"+delimiter2+"S70"+'\n')
for i in range(0,len(m)):
    out.write(str(dados_ent[i])+delimiter2)
    out.write(str(m[i])+delimiter2+str(n[i])+delimiter2+str(o[i])+delimiter2+str(p[i])+delimiter2+str(q[i]))
    out.write(delimiter2+str(r[i])+'\n')

print ("Simulacao terminada")
print ("Procure o arquivo de saida")

ent.close()
ent2.close()
out.close()

#####

```