ANALYSIS OF DNA SEQUENCES:

Modeling

Sequence Dependent Features and Their Biological Roles

A **THESIS**

SUBMITTED TO THE

UNIVERSITY OF POONA

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN

CHEMISTRY

BY

Ms. RUPALI VITTHAL PARBHANE

M. Sc. (Biochemistry)

CHEMICAL ENGINEERING DIVISION

NATIONAL CHEMICAL LABORATORY

PUNE 411 008, INDIA

October 2000

To my family

Certificate

Certified that the work incorporated in the thesis entitled "**Analysis Of DNA Sequences: Modeling Sequence Dependent Features and Their Biological Roles**" submitted by **Ms. Rupali Vitthal Parbhane** was carried out by the candidate under my supervision. Such material as has been obtained from other sources has been duly acknowledged in the thesis.

(B. D. Kulkarni)

Head and Deputy Director

Chemical Engineering Division

Candidates declaration

I hereby declare that the thesis entitled "**Analysis Of DNA Sequences: Modeling Sequence Dependent Features and Their Biological Roles**" submitted for Ph.D. degree to the University of Poona has not been submitted by me for a degree to any other university.

(Rupali V. Parbhane)

Chemical Engineering Division

National Chemical Laboratory

Pune 411 008, INDIA

Acknowledgements

I would like to express my deepest gratitude to Dr. B. D. Kulkarni for giving me an opportunity to carry out research work under his tutelage. His silent recognition, accurate observation, and critical analysis of a problem have always strengthened my will power to achieve the goals with perfection. Without his versatile guidance and support, it is unlikely that the thesis would have been completed.

I breathe my sincere thanks to Dr. S. S. Tambe for having undying patience while correcting my manuscripts. His strive for excellence will always remain a source of inspiration for me. His few soothing words in tough times and positive approach during struggling phase are worth remembering for lifetime.

My sincere thanks to Prof. V. Nagaraja, MCBL, IISc, Bangalore for having provided me all the experimental facilities for verifying one of my theoretical models. The stay at IISc campus was smooth due to his caring nature and parental concern.

The scientists like Prof. E.N. Trifonov, Weizmann Institute of Science, Israel; Prof. H.R. Drew, CSIRO Division of Biomolecular Engineering, Sydney; Prof. P. De Santis, Universita di Roma “La Sapienza”, Rome; Dr. Anita Anselmi, Dr. Ivan Brukner, Dr. Michelle Mulder mean so much to me as teachers. My sincere gratitude to them for having spared so much of their time in answering my queries.

I am grateful to my senior, Dr. Murli Nair for various reasons. He set a path and direction to my research when I newly joined at NCL. Discussions and debates with him have helped in generating new ideas as well as solving problematic situations.

It was a pleasure working in the environment constituting of senior colleagues like Dr. Jayaraman, Dr. Ravi kumar, Dr. Yadav. Due to all of them our CE group environment was of total intellectual and functional freedom.

I am always blessed by wonderful teachers like Mrs. Tapaswi, Mrs. Gadgil, Ms. Kelkar, Mrs. Vaidya, Ms. Gore, Mr. Khasgiwale, Dr. Chitale, Dr. Pol, Dr. Kodgul, Dr. Nadkarni, Dr. Hegde, Dr. Barnabas, Dr. Balani and so on. All of them have contributed so much in nurturing my academic interest.

I am thankful to Dr. Medha Joshi, Dr. Nita Parekh, Dr. Devapriya Choudhary for providing me their unfailing support on and off field. Their suggestions and constructive criticism went a long way in focussing my work.

I should not fail to place on record the help given to me by the office staff of the Chemical Engineering Division and the facilities provided by NCL library, Bioinformatics Center, Pune and IISc, Bangalore. A word of thanks to Mr. Bhujang for his excellent draftsmanship and Mr. Poman and Mr. Kamble for their helping attitude.

I am thankful to the Council of Scientific and Industrial Research, New Delhi for awarding a research fellowship and the Director, National Chemical Laboratory for permitting me to submit this work in the form of the thesis.

I would like to thank my friends Sangita, Sunita, Shashwati, Beena, Moneesha, Devayani, Vrinda, Shilpa, Mangala, Ashwin, Ramdas, Sreekumar, Satya, Rahul, Imran, Nandi, Jitender, Yogesh, Jayaram, Ashish, Jignesh, Anand, Narendra, Prashant for contributing so much in their own ways.

The thesis would have remained an unfulfilled dream without support and love of my family - “Aai, Dada, Kaka, Anna, Abhay, Rahul, Pradnya and Mama”. Their faith in me has always boosted my moral during struggling phase. A special mention of my soul mate, Nandu who have shown concern towards submission of my thesis and allowed my stay back in India with smiling face. How can I forget the wonderful moments spend with Anish, Dolly and Sidhdhali! What a delight all of you have been!

With this little eulogy, I have tried to express my everlasting gratitude towards all the people who deserves the credit of escalating my career to the height where I stand today.

Rupali Vitthal Parbhane

CONTENTS

Chapter 1 General Introduction

Chapter 2 Analysis of DNA curvature using artificial neural networks

Chapter 3 Artificial neural network based modeling of DNA sequences: new strategies using DNA shape code

Abbreviations

A	Adenine
ANN	Artificial neural network
bp	base-pair
C	Cytosine
CR	Calliper randomization
DNA	Deoxyribonucleic acid
EBP	Error-back-propagation
EIIP	Electron ion interaction potential
G	Guanine
GA	Genetic Algorithm
GDR	Generalized delta rule
m-RNA	messenger RNA
Pu	Purine
Py	Pyrimidine
R_L	Retardation anomaly
RMSE	Root-mean-squared-error
RTL	Relative transcription level
RNA	Ribonucleic acid
RNAP	RNA polymerase
T	Thymine
U	Uracil

ABSTRACT

Various genome projects are producing large amounts of DNA data sequences that need automated analysis for their characterization. Interpretation of nucleotide sequences by in-computo experiments with a view to providing some insight into the loc ation, structure and function of particular gene is thus clearly very important. The consequent increase in the number of approaches, algorithms and software to solve the problem is self-evident.

There have been a number of approaches, both experimental and theoretical, that have been directed towards understanding genome organization and functions. Theoretical approaches include statistical analysis, spectral analysis, linguistic analysis, Monte-Carlo methods and molecular dynamics approaches. Further, there is also a growing need to develop faster and newer methods for understanding biological processes. It is of paramount importance to develop techniques to unscramble the words in the sequences and read the hidden message. The encryption of messages in biological sequences is complex. The aforementioned methods while are useful in understanding a variety of biological phenomena; there still remain certain processes, which can not be analyzed using these techniques. Biological systems being very complex, it is often difficult to identify individual components of the systems and establish the way in which they interact with each other. The inherent complexity of biological systems makes it very difficult to understand them as well as to model them phenomenologically.

Artificial Neural Networks (ANNs) provide a unique computing architecture whose potential has attracted interest from researchers across different disciplines. One can simply view a neural network as a large set of interconnections with variable strengths (weights), in which the learned information is stored [1]. Recent advances in neural network theory and technology have made them powerful tool that helps to identify complex processes in the presence of noisy or incomplete information, colinearity of data, and time delays. It can also be used on incomplete data without assumed models or postulated formulas. Further, several features of neural network have encouraged their application to the analysis of protein and nucleic acids sequences. Neural networks have several unique characteristics and advantages as tools for the molecular sequence analysis problem. A very important feature of these networks is their adaptive nature, where “learning by example” replaces conventional “programming” in solving problems. This feature makes such computational models very appealing in application domains where one has little or incomplete understanding of the problem to be solved, but where training data are readily available.

A neural network consists of a large number of simple processing elements called neurons. The arrangement of neurons into layers and the connection patterns within and between layers is called the network architecture. In feedforward (FF) nets, the signals flow from the input units to the output units, in a forward direction: the input units receive signals from the outside world; the output units present the response of the net. The perceptron is the simplest form of a neural network used for the classification of the special type of patterns characterized as linearly separable. A perceptron has only two layers- input and output layers. It computes a linear combination of the network inputs and applies the net input to produce the output using a threshold output function. Multilayer perceptrons (MLPs) are generalized perceptrons with one or more hidden layers. A three-layer FF neural network is an MLP with one hidden layer and two layers of adaptive weights. An MLP has several distinctive characteristics: (a) it uses neurons with a differentiable non-linear activation function. (b) It has one or more layers of hidden neurons, which enables the network to learn complex tasks by extracting progressively more meaningful features from the input patterns; and (c) it exhibits a high degree of connectivity.

The neural network learning algorithms may be supervised or unsupervised. The back-propagation algorithm is an example of the supervised training. Examples of unsupervised training include the Kohonen self-organizing maps and the adaptive resonance theory (ART). ANNs have been applied to several problems in nucleic acid sequence analysis, viz. gene identification, intron/exon discrimination, prediction and analysis of promoters, terminators, ribosome binding sites, phylogenetic classification etc.

In recent years, a class of robust algorithms - known as “Genetic Algorithms” (GAs) - have been used with great success in solving optimization problems involving very large search spaces [2]. GAs were originally developed as genetic engineering models mimicking the population evolution in natural systems. Given a functional form, genetic algorithm searches its solution space so as to maximize (or minimize) the prespecified objective function. A simple GA has the following components: (i) representation/encoding scheme, (ii) initialization, (iii) fitness evaluation, (iv) selection policy: a) roulette wheel selection, b) tournament selection, (v) genetic operators- crossover and mutation. The thesis attempts at modeling the various sequence dependent features of DNA and their biological roles using ANNs and GAs.

Chapter 1 of the thesis introduces the subject and reviews the earlier work. In chapter 2 of the thesis, ANNs have been utilized for the prediction of DNA curvature in terms of Retardation Anomaly. The ANN model has been developed and illustrated using the

example and data of Bolshoy et al. [3]. The model captures the role of phasing, increased helix flexibility, run of polyA tracts, and flanking base pair effects in determining the extent of DNA curvature.

Chapter 3 describes two new encoding strategies, namely, *wedge* and *twist* codes that are introduced to represent DNA sequences in ANN-based modeling of biological systems. Wedge and twist codes are devised based on the direction of deflection angle, wedge angle and twist angle [4]. These codes have been evaluated by performing various case studies. The proposed coding schemes have been compared rigorously and shown to outperform the existing coding strategies especially in situations wherein limited data are available for building the ANN models.

In chapter 4, a hybrid technique involving two artificial intelligence (AI) tools viz., ANN and GA has been proposed for performing modeling and optimization of complex biological systems. In this methodology, first an ANN approximates (models) the non-linear relationship(s) existing between its input and output example data sets. Next, the GA, which is a stochastic optimization technique, searches the input space of the ANN with a view to optimize the ANN output. The efficacy of this formalism has been tested by conducting a case study involving optimization of DNA curvature characterized in terms of the R_L value. Using the ANN-GA methodology, a number of sequences possessing high R_L values have been obtained and analyzed to verify the existence of features known to be responsible for the occurrence of curvature. The methodology is a general one and can be suitably employed for optimizing any other biological feature.

In chapter 5, using an ANN and GA based hybrid strategy the effects of multiple base substitutions with particular emphasis on those that can cause maximum gene expression of β-globin gene are studied. The study reveals that multiple base substitutions in the conserved as well as non-conserved regions can cause substantial enhancements in relative transcription level (RTL). We identify positions in the nucleotide sequences, which preferably should not be altered, as well as those positions where mutations can lead to increased RTL. The various trends observed are rationalized. The ANN-GA strategy can help in experimental planning and reducing the search space.

In chapter 6, we have compiled 125 mycobacterial promoter sequences. Mycobacterial promoters have been analyzed for various features like: i) TSS, ii) -35 and -10 regions, iii) σ factor, iv) spacer length, v) upstream region of -35 box, and vi) % G+C content. These features are compared to similar features known for *E. coli* promoters. Further, the study suggests a broad classification of these promoters into three main types viz., i) *E. coli* type, ii)

Mycobacterial (*Non-E. coli*) type, and iii) Extended –10 promoters. The results throw some light on the mycobacterial transcription machinery and structure of mycobacterial promoters, which is an important step to understand the low level of its transcription, and the possible mechanisms of regulation of gene expression.

In chapter 7 of the thesis, a multilayered feed-forward ANN architecture has been used to predict the mycobacterial promoter sequences. The trained network has been used to determine the structurally/functionally important regions with the help of calliper randomization approach. Results obtained thereby indicate that the upstream region of –35 box, –35 region, spacer region, and -10 box are important for mycobacterial promoters.

Mycobacterial promoters have large variation in transcription mechanism. One of the important controlling factors in transcription initiation is DNA conformation of the promoter sequence. In chapter 8 of the thesis, we have analyzed our own compilation of mycobacterial promoters for DNA curvature distribution. This analysis has been performed using several di- and tri- nucleotide dependent models of DNA curvature. The results of curvature distribution are compared and contrasted with *E. coli* promoters.

KEY REFERENCES

1. Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1986) *Nature*, **323**, 533-536.
2. Goldberg, D.E. (1989) Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Reading, Mass.
3. Bolshoy, A., McNamara, P., Harrington, R.E. and Trifonov, E.N. (1991) *Proc. Natl Acad. Sci. USA*, **88**, 2312-2316.
4. Shpigelman, E., Trifonov, E., Bolshoy, A. (1993) *Comp. Appl. Biosci.*, **9**, 435-440.

CHAPTER

1

General Introduction

1.1 BACKGROUND

Various genomes are currently being intensively “spelled” (sequenced) and characterized by detecting in the sequences a few familiar features (protein-coding regions, transcription signals, *A*lu repeats), and deposited in sequence libraries, where they are further annotated and “shelved”. The most interesting part of the sequence processing - the “reading," depends on prior deep studies on the nature of the various codes carried by the sequences, of which we know only too little. The deciphering is not a simple task and bottlenecks exist in the development of our understanding of genome organization and functions.

While sequencing is progressing on at alarming pace, data analysis will certainly become a rate-limiting step. The succeeding phases of the project would then depend largely on interpreting nucleotide sequences by in computo experiments with a view to providing some insight into the location, structure and function of particular gene. It is needless to emphasize the importance of the problem and the consequent increase in the number of approaches, algorithms and software to solve the problem is self-evident. This will allow biological and medical researchers to focus their attention on promising and manageable subsets of the data.

There have been a number of approaches, both experimental and theoretical, that have been directed towards understanding genome organization and functions. Theoretical approaches include statistical analysis, spectral analysis, linguistic analysis, Monte-Carlo methods and molecular dynamics approaches. Further, there is also a growing need to develop faster and newer methods in understanding biological processes. It is of paramount importance to develop techniques to unscramble the words in the sequence and read the hidden message. The encryption of messages in biological sequences is complex. It is now being established that sequences no longer carry a single message (e.g., the triplet code which are instructions for protein synthesis) but, in fact, carry overlapping messages like the DNA shape code and the chromatin code. Other signals, which are responsible for vital cell activities like transcription, are also encoded in different regions. While the aforementioned methods are useful in understanding a variety of biological phenomena, there still remain certain processes, which can not be analyzed using these techniques.

Thus, biological systems being very complex, it is often difficult to identify individual components of the systems and establish the way in which they interact with each other. The inherent complexity of biological systems makes it very difficult to understand them as well as to model them phenomenologically. First principle models quantitatively articulate the cause-and-effect relationships. These models contain a number of system parameters, and take on the form of algebraic or differential equations. Given the numerical values of the parameters, the phenomenological models permit the calculation of system outputs for a given set of inputs. Thus, it is important to use an alternative approach that can be applied to systems about which only partial information is known.

1.2 ARTIFICIAL NEURAL NETWORKS

Artificial Neural Networks (ANNs) provide a unique computing architecture whose potential has attracted interest from researchers across different disciplines. The NN technique has its origin in efforts to produce a computer model of the information processing that takes place in the nervous system [1]. One can simply view a neural network as a large set of interconnections with variable strengths (weights), in which the learned information is stored.

ANNs appear to be one of the most suited alternative tools. ANNs are mathematical approximations of the biological synapses and were initially developed as models for understanding the brain mechanisms involved in perception. The abilities of the ANNs to perform nonlinear mapping and their powerful internal representation capability has led neural networks to be used as a tool for modeling rather than understanding the brain functions per se. Recent advances in neural network theory and technology have made them powerful tool that helps to identify complex processes in the presence of noisy or incomplete information, colinearity of data, and time delays. It can also be used on incomplete data without assumed models or postulated formulas. Further, several features of neural network have encouraged their application to the analysis of protein and nucleic acids sequences. ANNs can incorporate both positive and negative information, that is both sequences with the feature of interest and without the feature are used to impart knowledge to the network. They are also able to detect second- and higher- order

correlations in patterns, and thus, are more useful in determining complex correlations than the conventional methods based simply on the frequency of occurrence of residues at certain positions. An ANN based on the knowledge it acquires at the time of training makes its own internal representation of the system being modeled and then automatically determines which residues and which positions are important. Neural networks are thus ideally suited for parallel sequence processing and are increasingly applied to the study of biological macromolecules. They aim at mapping nucleic acid/protein sequences on to spatial structure/functionality.

1.2.1 Neural Network Characteristics

Neural networks have several unique characteristics and advantages as tools for the molecular sequence analysis problem. A very important feature of these networks is their adaptive nature, where "learning by example" replaces conventional "programming" in solving problems. This feature makes such computational models very appealing in application domains where one has little or incomplete understanding of the problem to be solved, but where training data are readily available. Owing to the large number of interconnections between their basic processing units, neural networks are error tolerant, and can deal with noisy data. Neural network architecture encodes information in a distributed fashion. This inherent parallelism makes it easy to optimize the network to deal with a large volume of data and to analyze numerous input parameters. Flexible encoding schemes can be used to combine heterogeneous sequence features for network input. Finally, a multilayer network is capable of capturing and discovering high-order correlations and relationships in input data.

1.2.2 Neural Network Architecture

A neural network consists of a large number of simple processing elements called neurons. The arrangement of neurons into layers and the connection patterns within and between layers is called the network architecture. In feedforward (FF) nets, the signals flow from the input units to the output units, in a forward direction: the input units receive signals from the outside world; the output units present the response of the net.

(1) Perceptrons

The perceptron [2] is the simplest form of a neural network used for the classification of the special type of patterns characterized as linearly separable. A perceptron has only two layers- input and output layers. It computes a linear combination of the network inputs and applies the net input to produce the output using a threshold output function. An elementary perceptron consists of a single output neuron with adjustable synaptic weights and a threshold. The threshold can be treated as a synaptic weight connected to a fixed input of value 1. Such a fixed input unit is called a bias unit. One can use the elementary perceptron to solve a pattern classification problem with only two classes. To perform classification with more than two classes requires the use of more output neurons.

The weights of the perceptron can be adapted on an iteration-by-iteration basis, using an error-correction rule known as the perceptron convergence theorem [3]. The theorem guarantees that if a solution exists, the perceptron learning rule will in a finite number of steps, converge to correct weights that produce correct output values for all training patterns. The convergence algorithm is non-parametric in the sense that it makes no assumptions concerning the form of the underlying distributions. It may thus be more robust than classical techniques.

(2) Multilayer Perceptron

Multilayer perceptrons (MLPs) are generalized perceptrons with one or more hidden layers. A three-layer FF neural network is an MLP with one hidden layer and two layers of adaptive weights. While simple perceptrons can perform classification only on linearly separable patterns, MLPs are general-purpose, flexible, non-linear models that, given enough hidden neurons and enough data, can approximate virtually any function to any desired degree of accuracy [4]. MLPs have been applied successfully to solve some difficult and diverse problems by training them in a supervised manner with a highly popular algorithm known as the error-back-propagation algorithm.

An MLP has several distinctive characteristics:

(1) It uses neurons with a differentiable non-linear activation function.

(2) It has one or more layers of hidden neurons, which enables the network to learn complex tasks by extracting progressively more meaningful features from the input patterns; and

(3) It exhibits a high degree of connectivity.

The presence of a distributed form of non-linearity and the high connectivity of the network make the theoretical analysis of an MLP difficult to undertake. The use of hidden neurons makes the learning process harder to visualize. The learning process is more difficult because the search has to be conducted in a much larger space of possible functions in order to decide how input features should be represented by the hidden neurons.

1.2.3 Neural Network Learning Paradigm

The neural network learning algorithms may be supervised or unsupervised. The supervised training is accomplished by presenting a sequence of training vectors; each with an associated target output vector. An essential ingredient of the supervised learning is the availability of an external teacher. The back-propagation algorithm is an example of the supervised training.

In unsupervised or self-organized learning there is no external teacher to oversee the learning process. The learning normally is driven by a similarity measure without specifying target vectors. The self-organizing net modifies the weights so that the most similar vectors are assigned to the same output (cluster) unit, which is represented by an examplar vector. Examples of unsupervised training include the Kohonen self-organizing maps [5] and the adaptive resonance theory (ART) [6].

A. Back propagation

The back-propagation (BP) learning rule is central to much current work on learning in NNs [7]. The generalized delta rule is simply a gradient-descent method to minimize the error signal [8]. The algorithm provides a conceptually efficient method for changing the weights in a feedforward network, with differentiable activation function units, to learn a training set of input-output examples. BP can be used with a variety of architectures. The elementary BP network is a multilayer perceptron.

The BP training involves three stages: the feedforward of the input training pattern; the calculation and back-propagation of the associated error, and the adjustment of the weights. In the feedforward phase, the weights remain unaltered throughout the network, and the function signals of the network are computed on a neuron-by-neuron basis. In the back-propagation phase, error signals are computed recursively for each neuron starting at the output layer, and passed backward through the network, layer by layer (hence, the name "back-propagation"), to derive the error of hidden units. Weights are then adjusted to decrease the difference between the network's output and the target output. Since learning here is supervised (i.e., target outputs are available), an error function may be defined to measure the degree of approximation for any setting of the network weights. After training, application of the note involves only the computations of the feedforward phase. Even if training is slow, a trained net can produce its output very rapidly.

Many enhancements and variations have been proposed for the BP algorithm. These are mostly heuristic modifications with goals of increased speed of convergence, avoidance of local minima, and/or improvement in the network's ability to generalize. A theoretical framework for studying BP was described by Le Cun [9], whose formalism is well suited to the description of many different variations of BP. In the context of NN, Bayesian methods offer a number of important features [10]. A Bayesian framework was formulated [11] to provide objective criteria for comparing solutions using alternative network architectures, parameter settings, and alternative learning and interpolation models. The relative importance of different inputs can also be determined using a Bayesian technique [12].

B. Kohonen's self-organizing map

The self-organizing map has the special property of effectively creating a spatially organized internal representation of various features of input signals and their abstractions. The feature map is a two-layered network that can organize a topological map of cluster units from a random starting point. The network combines an input layer with a competitive layer of processing units. During the self-organization process, the cluster unit, whose weight vector matches the input pattern most closely (typically based on minimum Euclidean distance), is chosen as the winner. The winning unit and its neighboring units (in

terms of the topology of the cluster units) update their weights. After training is complete, pattern relationships and groups are observed from the competitive layer. This yields the graphical organization of pattern relationships. These maps result from an information compression that retains only the most relevant common features of the set of input signals.

C. Counter Propagation

The counter-propagation (CP) network [13] is an example in which layers from supervised and unsupervised learning paradigms are combined to construct a new type of network. A CP net is closely related to the nearest-neighbor classifier. Nearest-neighbor classifiers require a unit for every learned example in a training set. They are impractical as on-line classifiers because of the large number of computations required in classifying a new input. Thus, one needs to have a compact presentation of training data and use far fewer than one unit for every training sample. The CP approximates its training input vector pairs by adaptively constructing a look-up table. In this manner, a large number of training data points can be compressed to a more manageable number of look-up table entries. The accuracy of the approximation is determined by the number of entries in the look-up table, which equals the number of units in the cluster layer of the net.

The forward-only CP network has three layers: an input layer; a Kohonen clustering layer; and a Grossberg conditioning layer. As a pattern classifier, a CP network uses the Kohonen layer to determine winning units for the input patterns, and uses the Grossberg layer to map these winners into classes. The Kohonen layer is an LVQ (learning vector quantizer) network [14], which performs nearest-neighbor classification. The clusters may be formed based on either the dot (inner) product or the Euclidean distance. In the Grossberg layer, the weights from the cluster units to the output units are adapted to produce the desired response. Counter propagation is considered a faster alternative to BP, although questions remain about his performance.

1.2.4 Applications for DNA/RNA sequence analysis

ANNs have been applied to several problems in nucleic acid sequence analysis, viz. gene identification, intron/exon discrimination, prediction and analysis of promoters,

terminators, ribosome binding sites, phylogenetic classification etc. The brief summary of all such applications is listed in Table I.

Neural Network architectures:

2L,FF = two-layer, feedforward network (i.e., perceptron)

3L or 4L, FF = three-or-four layer, feedforward network (i.e., multi-layer perceptron)

Neural Network Learning Algorithms:

BP = back-propagation

Delta = Delta rule

QP = Quick-propagation

RP = Rprop

ART = Adaptive resonance theory

CP = Counter-propagation

Input sequence encoding methods:

BINn = binary-numbered direct encoding of residue identity, where n is the number of input units representing each residue

REALn = real-numbered direct encoding of residue features, where n is the number of units representing each residue

FEATn = indirect encoding of sequence residue frequency

FREQ = indirect encoding of sequence residue frequency

SVD = singular value decomposition

Output sequence encoding methods

n(CODEs) where n is the number of output units.

CODEs are: Y = Yes (positive); N = No (negative); I = Intron, E = Exon; O = Other (counter-example); RTL = relative transcription level.

Table I: Applications for DNA/RNA sequence analysis

Application	ANN Architecture	Input/Output Encoding	Ref.
INTRON/EXON (I/E) DISCRIMINATION AND GENE IDENTIFICATION			
Coding region recognition	4L,FF,BP	FEAT7/1(Y,N)	[15]
Coding region recognition	3L,FF,BP	FEAT13/1(Y,N)	[16]
I/E feature weighting	2L, FF, Delta	FEAT6/1(Inequality)	[17]
I/E feature weighting	2,3L, FF, Delta, BP	FEAT6/1(Inequality)	[18]
Splicing donor/acceptor site prediction	3L, FF, BP	BIN4/1(Y,N)	[19]
Splicing donor/acceptor site prediction	3L, FF, BP	BIN4/1(Y,N)	[20]
Splicing donor/acceptor site prediction	3,4L, FF, BP	BIN4/1(Y,N)	[21]
I/E discrimination	2L, FF, BP	BIN4,FREQ/1(Y,N)	[22]
I/E compositional constraints	3L,FF,BP	BIN4/3(I,E,O)	[23]
Parallel implementation for I/E discrimination	3L,FF,BP,QP,RP	BIN4/1(I,E)	[24]
PREDICTION & ANALYSIS OF RIBOSOME-BINDING SITES, PROMOTERS AND OTHER SITES			
Ribosome-binding site prediction	Perceptron	BIN4/1(Y,N)	[25]
Ribosome-binding site prediction	3L,FF,BP	BIN4/1(Y,N)	[26]
Ribosome-binding site prediction	3L,FF,BP	BIN4/1(Y,N)	[27]
E. coli promoter prediction	2×3L,FF,BP	BIN2/1(Y,N)	[28]
E. coli promoter prediction	Perceptron	?	[29]
E. coli promoter prediction	3L,FF,BP	BIN4/1(Y,N)	[30]
E. coli promoter prediction	3L,FF,BP	BIN2,BIN4/1(Y,N)	[31]
E. coli promoter prediction	3L,FF,BP	BIN4/1(Y,N)	[32-33]
E. coli promoter prediction	3L,FF,BP	BIN4 +3 + FREQ/1(Y,N)	[34]
E. coli promoter prediction	2×3L,FF,BP	BIN4/1(Y,N)	[35]
Transcription start site and feature detection	3L,FF,BP	BIN4/1(Y,N)	[36]
Eukaryotic promoter prediction	3L,FF,BP	BIN4/1(Y,N)	[37]
RNA polymerase II binding site prediction	4L,FF,BP	FEAT13/1(Y,N)	[38]
Prediction of transcriptional terminator	3L,FF,BP	BIN4, REAL1/1(Y,N)	[39]
Prediction of transcription control signal	3L,FF,BP	BIN4/1(RTL)	[40]
DNA/RNA SEQUENCE ANALYSIS, PHYLOGENETIC CLASSIFICATION AND CODE MAPPING			
Clustering and functional region identification	2L,Kohonen	REAL1/Map(30)	[41]
Clustering and functional region identification	2L,Kohonen	REAL1/Map	[42]
Phylogenetic classification	2L,ART	BIN4/19(Class)	[43]
Ribosomal RNA classification	2×3L,FF,BP,CP	FREQ,SVD/220,15 (Class)	[44]
Transfer RNA gene recognition	3L,FF,BP	BIN4/10(Class)	[45]
Genetic code mapping	3L,FF,BP	BIN4/20(Class)	[46]

As a technique for computational analysis, neural network technology is very well suited for the analysis of molecular sequence data. The perceptron learning algorithm developed by Rosenblatt [2] was adapted to sequence pattern analysis by Stormo et al., [25] in an attempt to distinguish ribosomal binding sites from non-binding sites. The conceptual basis of the back-propagation learning algorithm was first presented by Werbos [47]. Later, Rumelhart and his colleagues introduced the broad potential of the NN approach and presented the back-propagation algorithm to a wider readership [1, 48]. Back-propagation soon became the most popular NN paradigm. It has been successfully used to perform a variety of input-output mapping tasks for recognition, generalization, and classification [49], including many molecular sequence analysis problems. As the field continues to develop, researchers have broadened the choices of NN architectures and learning paradigms to solve a wider range of problems.

1.3 GENETIC ALGORITHMS

Genetic Algorithms (GAs) are stochastic methods, which enforce the survival of the fittest paradigm of evolution along with the genetic propagation of characteristics. A simple GA has the following components:

1. Representation/Encoding scheme
2. Initialization
3. Fitness Evaluation
4. Selection Policy
5. Genetic Operators

1.3.1 Representation

Most problems in GA literature use the binary encoding scheme where each locus of the string is drawn from a binary alphabet of zero or one.

1.3.2 Initialization

Initialization refers to the generation of the initial population of solutions as well as the choice of some parameters of the population, such as its size. The preferred

characteristics of an initial population are diversity and reasonable levels of fitness values. However, in practice, depending upon the application, generating an initial population varies from random generation to careful choosing of candidates based on the user's experience. Sometimes choosing few distinct and diverse solutions and assigning copies based on their fitness values could provide a good starting population. Optimal choice of the population size tends to depend upon the nature of the domain, the representation, the evaluation scheme and the genetic operators used. In this algorithm, the population is continuously augmented by the newly created products of recombination. However, the algorithm has a measure of the age or lifetime of an individual beyond which individual 'dies' or is removed from the population. This lifetime, instead of the selection probability, is set proportional to the fitness of the individual. This means that fitter individuals live longer than the rest and the population is controlled by the death rate of individuals.

1.3.3 Fitness Evaluation

Once a population of candidate solutions has been created, they need to be evaluated to determine their fitness in the environment. For an optimization problem, the environment is the objective function. Depending on how low (for minimization problems) or how high (for maximization problems) the objective function value for an individual is, its fitness should have a proportionally high value. In some problems one does not have a single objective but several to be optimized simultaneously as well as constraints to be satisfied by the solutions. One way of handling multiple objectives is to define a new objective function that is a weighted sum of all the objectives. Here, the choice of the weights can reflect the relative importance of optimizing the different objectives. To handle constraints in genetic algorithms, the objective function is usually augmented with a penalty term that weights in the feasibility of the solution.

Fitness Scaling:

Scaling the solutions within the population ensures that individuals with fitness equal to the average of the population contribute one expected offspring to the next generation [50]. Also, later during the run, scaling overcomes lack of differentiation between average and the best members of the population. The most widely used scaling method is linear scaling.

1.3.4 Selection Scheme

The selection scheme has to make sure that the fitter individuals in the population are allotted more opportunities to reproduce and recombine to produce offspring. To this end, two different selection schemes are normally used.

Roulette Wheel Selection

In this scheme, once the fitness evaluation is completed, the population is sorted in ascending order of fitness and a running sum of the fitness is calculated for each member starting from the first one in the sorted list. The first member of the sorted list (beginning with the member with the lowest fitness) whose cumulative fitness is greater than the random number, is selected. The Roulette Wheel Selection procedure can be thought of as a dynamic selection scheme with a variable probability of selection across generations.

Tournament Selection

In this scheme, a specified number, called the tournament size, of members are chosen from the parent population and enter a competition. The winner is decided based on the best fitness and allowed to enter the reproductive phase. This process is repeated sufficiently, along with recombination and mutation, to produce the offspring population. This method slightly offsets the effects of a few large fitness solutions in the population by biasing the selection scheme towards above average solutions in general [50]. As opposed to the roulette wheel selection procedure, this is a static selection scheme where the probability of selection of a candidate remains fairly constant across generations.

1.3.5 Genetic Operators

Genetic operators provide the means by which the genetic components or the building blocks of the current population (the parents) are altered to produce the next population (the offspring). Genetic operations typically fall under two categories: i) crossover and ii) mutation.

Crossover

Chromosomal crossover refers to the random recombination of parts of two chromosomes (the parents) to produce two new chromosomes (the offspring). This is a

large-scale operator in the sense that it significantly perturbs the genotype of the parents. From an optimization viewpoint, the recombination operator tends to improve the combinatorial diversity by using the building blocks present in the population.

Mutation

To be effective, the GA needs an influx of characteristics extraneous to the population. This is provided by the mutation operator. For a simple GA using binary encoding, mutation is normally applied after crossover and with a low probability (around 1%). This is because, with high probabilities, mutation tends continually to destroy the good features (schema) brought forth by recombination and selection.

1.3.6 Applications of Genetic Algorithms

Conformational analysis involves the search for the structure or conformation that gives the global minimum in total potential energy or minimum deviation from a set of constraints derived from experiments. A conformation is normally characterized by a set of bonds and torsion angles that are constrained to satisfy these structural and molecular constraints. The earliest application of GAs to this problem was by Lucasius et al. [51-52] using the DENISE program to generate plausible DNA fragments to fit constraints obtained from NMR. This approach actually uses a two-tier GA to optimize first the components and then the entire structure. A similar problem in protein folding has been solved using GAs by Dandekar and Argos [53]. Here, the protein is modeled as fragments of amino acids each of which can assume different conformations from a predefined set and the idea is to locate the best combination, so as to minimize a defined fitness function. This function is a sum of several terms relating to the secondary and tertiary structure of the protein. In related work, hybrid GA based methods in conformational analysis have been instrumented in the elucidation of the structure of C_{60} (buckminsterfullerene) [54]. A new modeling technique for arriving at the three dimensional (3-D) structure of an RNA stem-loop has been developed based on a conformational search by a genetic algorithm and following refinement by energy minimization [55].

Comparison of the secondary structure of the 5' non-coding region of poliovirus 3 RNA derived from the genetic algorithm with the model of Skinner et al. [56] demonstrates many of the confirmed structural elements. The GA generates a population of all possible

stems, then mixes, combines and recombines these stems in multiple iterations on a massively parallel computer, ultimately selecting a most fit structure based on its energy [57]. The secondary structure of the region containing the determinants of neurovirulence was better predicted using the genetic algorithm, whereas the dynamic programming algorithm [58] required phylogenetic comparative sequence analysis to arrive at the correct conclusion.

1.4 GENESIS AND SCOPE OF THE PRESENT WORK

Biological systems are complex in nature and several known and unknown factors govern their functioning. It is difficult most of the times to interpret underlying relationship(s) between several experimental conditions and corresponding system output(s). Phenomenological modeling of such systems is also difficult due to the inherent complexity of biological systems and inadequate information about them. Thus, it is important to develop and use alternate methods that can be applied to systems with inadequate information. Artificial Intelligence (AI) tools viz. ANN and GA can uncover the underlying relationship(s) of such biological systems.

Detailed understanding of the biosystems require carrying out experiments that are often costly and time consuming. Most of the experiments are also difficult to perform. Due to multilevel interactions, a small change in input parameter of the system may result in changes in large number of features of system. Thus, to have a predictive model that captures the cause and effect relationship is certainly a difficult task. AI tools like ANN and GA can help in building up predictive models and use qualitative and quantitative information about the system. Thus, such modeling can help us in having better understanding of intricate biosystems. Therefore, the primary objective of this thesis is: i) to built up quantitative predictive relationship between inputs and outputs of biosystems wherever possible, and ii) in instances where such predictive quantitative relationship can not be built due to gross inadequacy of input-output data. It is hoped that they would at least provide qualitative guidelines for narrowing the choice of experiments to be performed.

It is with this view that in chapter 2, we develop an ANN model to establish a correlation between a nucleotide sequence of DNA and its effective curvature,

characterized in terms of retardation anomaly (R_L) value. In chapter 3 for ANN – based modeling of DNA sequences, two new input coding strategies namely, the *wedge* and the *twist* code have been suggested. The performance of the proposed strategies has been tested by performing various case studies. Chapter 4, presents a hybrid strategy involving an ANN and a GA for the optimization of a biologically important feature or property. This strategy is general and is illustrated using an example of optimization of DNA curvature. Chapter 5 illustrates a hybrid non-linear strategy involving an ANN and GA for optimization of transcription efficiency in eukaryotic systems using β-globin gene as a case example. The study helps to obtain an insight into the structural aspects of β-globin gene leading to high transcription efficiency.

Chapter 6 of the thesis provides a compilation of different mycobacterial promoters and analysis of their DNA sequences for various features. In chapter 7, an ANN model is developed for classifying mycobacterial promoter sequences from non-promoter sequences. Calliper randomization approach has been suggested for determining structurally and functionally important regions within the mycobacterial promoter sequences. Chapter 8 presents theoretical analysis of DNA curvature for mycobacterial promoters using several di- and trinucleotide dependent models of DNA curvature.

In essence, the thesis aims at building predictive relationships using AI tools for complex biological systems with a view to model and analyze DNA sequences for their properties and biological roles. This continues to be a poorly understood area and it is hoped that the approach adopted in the thesis takes a step forward in resolving the issues.

1.5 REFERENCES

1. Rumelhart, D.E. and McClelland, J.L. (1986) Parallel Distributed Processing, MIT Press, Cambridge, MA.
2. Rosenblatt, F. (1962) Principles of Neurodynamics, Spartan Books, Washington, DC.
3. Minsky, M. and Papert, S. (1969) Perceptrons, MIT Press, Cambridge, MA.
4. White, H. (1992) Artificial Neural Networks: Approximation and Learning Theory, Blackwell, Oxford.
5. Kohonen, T. (1989) Self-organization and Associative Memory, 3rd edn. Springer. Berlin and Heidelberg.
6. Carpenter, G.A. and Grossberg, S. (1988) *Computer*, **21**, 77-88.
7. Chauvin, Y. and Rumelhart, D.E. (1995) Backpropagation: Theory, Architectures and Applications, Lawrence Erlbaum Associates, Hillsdale, NJ.
8. Baldi, P. (1995) *IEEE Transactions on Neural Networks*, **6**, 182-195.
9. Le Cun, Y. (1988) In *Proceedings of the 1988 Connectionist Models Summer School*, eds D. Touretzky, G. Hinton and T. Sejnowski, pp. 21-28. Morgan Kaufmann, San Mateo, CA.
10. Bishop, C. (1995) Neural Networks for Pattern Recognition. Oxford University Press, New York.
11. MacKay, D.J.C (1992) *Neural Computation*, **4**, 448-472.
12. MacKay, D.J.C. (1994) In *Models of Neural Networks III*, eds E. Dormany, J.L. van Hemmen and K. Schulten, Springer, New York.
13. Hecht-Nielsen, R. (1987) *Applied Optics*, **26**, 4979-4984.
14. Kohonen, T. (1988) *Neural Networks*, **11**, 303.
15. Uberbacher, E.C. and Mural, R.J. (1991) *Proc. Natl. Acad. Sci. USA*, **88**, 11261-11265.
16. Uberbacher, E.C., Xu, Y. and Mural, R.J. (1996) *Methods Enzymol.*, **266**, 259-281.
17. Snyder, E.E. and Stormo, G.D. (1993) *Nucl. Acids Res.*, **21**, 607-613.
18. Snyder, E.E. and Stormo, G.D. (1995) *J. Mol. Biol.*, **248**, 1-18.

19. Brunak, S., Engelbrecht, J. and Knudsen, S. (1990) *Nucl. Acids Res.*, **18**, 4797-4801.
20. Brunak, S., Engelbrecht, J. and Knudsen, S. (1991) *J. Mol. Biol.*, **220**, 49-65.
21. Ogura, H., Agata, H., Xie, M., Odaka, T. and Furutani, H. (1997) *Comput. Bio. Med.*, **27**, 67-75.
22. Farber, R., Lapedes, A. and Sirotkin, K. (1992) *J. Mol. Biol.*, **226**, 471-479.
23. Granjeon, E. and Tarroux, P. (1995) *Comp. Applic. Biosci.*, **11**, 29-37.
24. Reczko, M., Hatzigeorgiou, A., Mache, N., Zell, A. and Suhai, S. (1995) *Comp. Applic. Biosci.*, **11**, 309-315.
25. Stormo, G.D., Schneider, T.D. and Gold, L. (1982) *Nucl. Acids Res.*, **10**, 2997-3011.
26. Bisant, D. and Maizel, J. (1995) *Nucl. Acids Res.*, **23**, 1632-1639.
27. Nair, T.M. (1997) *J. Biomol. Struct. & Dyn.*, **15**, 611-617.
28. Lukashin, A., Anshelevich, V., Amirkiyan, B., Gragerov, A. and Frank-Kamenetskii, M. (1989). *J. Biomol. Struct. Dyn.*, **6**, 1123-1133.
29. Nakata, K., Kanehisa, M. and Maizel, J. (1988) *Comput. Applic. Biosci.*, **4**, 367-371.
30. Abremski, K., Sirotkin, K. and Lapedes, A. (1993) *Mathematical Modelling and Scientific Computing*, **2**, 636-641.
31. Demeler, B. and Zhou, G.W. (1991) *Nucl. Acids Res.*, **19**, 1593-1599.
32. O'Neill, M.C. (1991) *Nucl. Acids Res.*, **19**, 313-318.
33. O'Neill, M.C. (1992) *Nucl. Acids Res.*, **20**, 3471-3477.
34. Horton, P.B. and Kanehisa, M. (1992) *Nucl. Acids Res.*, 20, 4331-4338.
35. Mahadevan, I. and Ghosh, I. (1994) *Nucl. Acids Res.*, **22**, 2158-2165.
36. Pedersen, A. G. and Engelbrecht, J. (1995) In Proceedings of the *Third International Conference on Intelligent Systems for Molecular Biology*, eds C. Rawling, D. Clark, R. Altman, T. Hunter, T. Lengaver and S. Wodak, pp. 292-299. AAAI Press, Menlo Park, CA.
37. Larsen, N.I., Engelbrecht, J. and Brunak, S. (1995) *Nucl. Acids Res.*, **23**, 1223-1230.

38. Matis, S., Xu, Y., Shah, M., Guan, X. and Einstein, J.R. et al.. (1996) *Comp. & Chem.*, **20**, 135-140.

39. Nair, T. M., Tambe, S.S. and Kulkarni, B.D. (1994) *FEBS Lett.*, **346**, 273-277.

40. Nair, T. M., Tambe, S.S. and Kulkarni, B.D. (1995) *Comp. Applic. Biosci.*, **11**, 293-300.

41. Arrigo, P., Giuliano, F., Scalia, F., Rapallo, A. and Damiani, G. (1991) *Comp. Applic. Biosci.*, **7**, 353-357.

42. Giuliano, F., Arrigo, P., Scalia, F., Cardo, P.P. and Daminani, G. (1993) *Comp. Applic. Biosci.*, **9**, 687-693.

43. Leblanc, C., Katholi, C.R., Unnasch, T.R. and Hruska, S. I. (1994) In *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, eds R. Altman, D. Brutlag, P. Karp, R. Lathrop and D. Searls, pp. 253-260. AAAI Press, Menlo Park, CA.

44. Wu, C.H. and Shivakumar, S. (1994) *Nucl. Acids Res.*, **22**, 4291-4299.

45. Sun, J., Song, W.Y., Zhu, L.H. and Chen, R.S. (1995) *J. Comput. Biol.*, **2**, 409-416.

46. Tolstrup, N., Engelbrecht, T. J. and Brunak, S. (1994) *J. Mol. Biol.*, **243**, 816-820.

47. Werbos, P.J. (1974) Beyond regression. New tools for prediction and analysis in the behavioral sciences. Thesis, Harvard University.

48. Rumelhart, D.E., Hinton, G.E. and Williams, R. J. (1986) *Nature*, **323**, 533-536.

49. Dayhoff, J. (1990) Neural Network Architectures: An Introduction. Van Nostrand Reinhold, New York.

50. Goldberg, D.E. (1989) Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading, MA, 412.

51. Lucasius, C.B., Blommers, M.J., Buydens, L.M.C. and Kateman, G. (1991) Handbook of Genetic Algorithms. ed. L. Davis, Van Nostrand Reinhold, New York, 251-281.

52. Blommers, M.J., Lucasius, C.B., Buydens, L.M. and Kateman, G. (1992) *Biopolymers*, **32**, 45-52.

53. Dandekar, T. and Argos, P. (1994) *J. Mol. Biol.*, **236**, 844-861.

54. Deaven, D.M. and Ho, K.M. (1995) *Phys. Rev. Lett.*, **75**, 288-291.

55. Ogata, H; Akiyama, Y. and Kanehisa, M. (1995) *Nucl. Acids Res.*, **23**, 419-426.

56. Skinner, M.A., Racanaiello, V.R., Dunn, G., Cooper, J., Minor, P.D. and Almond, J.W. (1989) *J. Mol. Biol.*, **207**, 379-392.

57. Shapiro, B.A. and Navetta, J. (1994) *J. Supercomput.*, **8**, 195-201.

58. Zuker, M. (1989) *Science*, **244**, 48-52.

CHAPTER

Analysis of DNA curvature using artificial neural networks

In this chapter, we develop an Artificial Neural Network (ANN) for the prediction of DNA curvature in terms of Retardation Anomaly. An ANN capturing the role of phasing, increased helix flexibility, run of polyA tracts, and flanking base pair effects in determining the extent of DNA curvature has been developed. The network predictions validate the experimentally known results and also explain how the base pairs other than ApA affect the curvature. The results suggest that ANN can be used as a model-free tool for studying the DNA curvature.

2.1 INTRODUCTION

The concept of sequence-dependent DNA structure was proposed more than a decade ago [1-3]. It is important in packaging, recombination, and transcription. In polyacrylamide gel DNA molecules are believed to migrate by a head-on reptation mechanism [4-5]. Relative electrophoretic mobility of most curved DNA fragments monotonously decreases with the fragment length. This is usually characterized as an increasing ratio of apparent to actual DNA length (known as "R_L factor") with increase in the fragment length. The R_L- factor is a measure of electrophoretic anomaly of the curved DNA and reflects the additional friction of the DNA in the gel due to curvature [6]. For most curved DNA fragments, therefore, longer the length, greater is the frictional drag and the R_L-factor is a monotonously increasing function of the fragment length [7]. The principal sequence feature responsible for intrinsic DNA curvature is generally assumed to be the runs of adenines. However, according to the wedge model of DNA curvature, each dinucleotide step is associated with a characteristic deflection of the local helix axis [8]. It is to be noted, however, that the first principle models for predicting the curvature are themselves being debated for their generality [9]. The objective of this chapter is to utilize artificial neural networks (ANN) for establishing a correlation between a nucleotide sequence of DNA and its effective curvature wherein the curvature is characterized in terms of the R_L value. Additionally, a detailed study of the effect of base substitutions as well as effect of different factors on the DNA bend has been conducted.

ANNs are massively connected parallel structures containing processing elements called *neurons*. The neurons communicate via a set of interconnections with variable strengths (weights). The phenomenal abilities of ANNs; to perform nonlinear mapping from input to output space, and classification, has led them to be used as a powerful tool for modeling rather than understanding the brain functions per se. In order that a network learns the input-output mapping, or classification, it needs to be trained with the help of available examples. Training procedure involves adjustment of the connection weights until the network learns the mapping/classification. ANNs trained with the error-back-propagation (EBP) algorithm [10-

11] represent the most widely used network paradigm. An EBP network is a multilayered feedforward structure that undergoes supervised learning; i.e., for training, it requires an example data set comprising pairs of input and the corresponding desired output patterns (vectors). Once adequately trained, the network can make predictions corresponding to the new input data. In biological sciences, the EBP networks have been successfully used for promoter recognition, terminator recognition, non-coding regions of DNA, capturing transcription control signals, phylogenetic analysis, etc. (see review [12]).

2.2 SYSTEM AND METHODS

The simulation programs were written in FORTRAN-77 and compiled using the Microsoft FORTRAN 5.0 compiler for the IBM PC and compatibles.

2.2.1 Data

The EBP network was trained using the experimental data by Bolshoy et al. [8] comprising the R_L values of circular and curved, and straight synthetic fragments extrapolated to 90 base pair length (columns 1-3 of Table I). These data were chosen since they are self-consistent wherein all the experiments are carried out under 'standard' gel conditions [13]. The data set comprising a total of 54 sequences and their corresponding R_L values was divided into training (40 patterns) and test (14 patterns) sets. The test set is used to evaluate the generalization capability of the EBP network in predicting the R_L values corresponding to the set of fragments not used during training.

2.2.2 Data Representation

Two possible ways to code nucleotide sequences, namely, CODE-2 and CODE-4 have been generally used for data representation. In these strategies, each nucleotide is represented by a unique two (CODE-2) or a four (CODE-4) digit binary string. Consequently,

as many (two or 4) input neurons are required to code a single nucleotide. Nair et al. [14] devised a novel coding strategy known as *Electron Ion Interaction Potential* (EIIP) code wherein each nucleotide is represented by its EIIP value; thus a single input neuron is sufficient for the nucleotide representation. In an event when the available data is limited, it is preferable to use EIIP coding since it results in smaller (as compared to CODE-2 and CODE-4 strategies) network. In EIIP strategy, the four nucleotides are coded as: A,0.1260; T,0.1335; G,0.0806; and C,0.1340. In the present study, these values have been used to represent the DNA sequences.

As can be seen from Table I, the nucleotide sequences are of different sizes, i.e., 10, 21, 31 and 42 base pair long. For training, the input vectors (coded fragments) need to be of the same size. Thus, the shorter fragments were uniformly padded with 0.01 until each fragment is 42 base pair long. The resulting data can be viewed as a matrix of size (54×42). Each column of this matrix was normalized, so that upon normalization, each matrix element lies between 0.05 and 0.95. It is to be noted that the DNA sequences considered in this study are of two types, namely, circular (sequence nos. 1-3) and linear (sequence nos. 4-54). In order to differentiate them, two additional inputs were considered at the end positions of each input vector. Accordingly, the circular and linear fragments were coded as (0.05, 0.90) and (0.90, 0.05), respectively. The experimental R_L values were also normalized so as to lie between 0.05 and 0.95 and taken as the target output for the network training.

2.2.3 Neural Network Simulation

The neural network simulations were performed on a 486 AT equipped with a math coprocessor. The network consists of three layers viz. *input, hidden* and *output* (Figure 2-1). The neurons in the input layer are simple distribution nodes, which pass their input as the output. The number of neurons (44) in the input layer is equal to the dimensionality of the input vector and the number of output layer neurons (1) is same as the dimensionality of the output vector. However, there is no easy way to assign the number of neurons in the hidden layer

responsible for the nonlinear representational ability of the EBP networks and, thus, the number is fixed heuristically. In this study, logistic sigmoid transfer function is used at the hidden and output nodes to represent the non-linearity. The network training is an error minimization procedure involving adjustment of the network weights until the error (the difference between the desired and network-predicted outputs) with respect to the test set is minimized. For weight updation the generalized delta rule with the momentum term [10] has been used. The error function, namely, the *root mean squared error* (RMSE), is defined as:

$$RMSE = \sqrt{\frac{\sum_{p=1}^{P} E_p^2}{p \times n}} = \sqrt{\frac{\sum_{p=1}^{P} \sum_{i=1}^{n} \left(t_{pi} - o_{pi}\right)^2}{p \times n}} \qquad (1)$$

where index p ranges over the number of input patterns (P); i ranges over the number (n) of output units; E_p represents the error on pattern p and, t_{pi} and o_{pi} are the target and actual output values of the ith output unit when pth pattern is presented to the network. The detailed algorithmic steps for EBP network training can be found in several references [15-16] and briefly summarized in Appendix I. In order to get optimal network weights three parameters, namely, momentum coefficient, learning rate and number of neurons in the hidden layer need to be heuristically optimized. These were found to be 0.1, 0.15 and 1, respectively. The RMSE profiles corresponding to the training and the test sets for the optimized network are shown in Figure 2-2. The weights after 4690 training iterations correspond to the minimum RMSE with respect to the test set and, hence, were taken as optimal.

2.3 RESULTS AND DISCUSSION

Table I shows a comparison of the network predicted and experimental retardation anomaly (R_L) values for the sequences in training and test sets (also see Figure 2-3). The correlation coefficient for the network predicted and experimental R_L values has been found to be 0.954 and suggests that the EBP network has satisfactorily captured the relationship between a DNA sequence and its R_L value. In addition, the trained EBPN is capable of predicting correctly low as well as high R_L values (refer Figure 2-3). Some misfit in the higher R_L values may be because of a limited sequence ensemble available for training the network.

The optimized net was subsequently used to evaluate the effect of single base substitutions on the R_L values. Towards this end, each nucleotide from a sequence was substituted with the remaining three on one-at-a-time basis and the network was used to predict the R_L value of the resulting sequence. The substitutions, which caused significant changes in the RL values, are listed in Table II. For conciseness, the results for sequence nos. 8, 10, 14, 16, 17, 31, 52 only are listed. In Figure 2-4, the graphical representation of the results for sequence nos. 8, 52, 17 and 16 is provided. These sequences are representative of the base pair lengths 10, 21, 31 and 42, respectively. The network predicted R_L values indicate that the mutations related to curvature are of three types: (i) the ones resulting in significant change in curvature with possible explanation from previous studies, (ii) those causing slight or no change in curvature, and (iii) those with no apparent explanation from the previous studies.

It is noticed that A tract of length 3 to 6 base pairs causes significant bending (see sequence 10, 14, 17). The R_L values obtained by mutating polyA tract validate the observation made by Milton and Gesteland [17] that each adenine residue in the A tract does not contribute equally. It can be seen from the plots of all possible single base substitutions in the polyA tract for sequence 8, 16, and 17 that substitutions with either T, G or C cause dissimilar effects. For instance, in the plot for sequence 17 in Figure 2-4, the R_L values resulting from the substitutions in the polyA tracts by T, G or C (positions 5 to 9 and 14 to 17) exhibit varying sensitivities towards single base substitutions. Mutations in the non-AA fragments indicate that some mutations alter the DNA bend even when they do not lead to formation of polyA tract. For example, if sequence 52 is mutated by G at position 14, the R_L changes from 1.05 to 1.16, and if position 19 is mutated by G, the resultant R_L is 0.79. This can be interpreted as base steps other than ApA are involved in sequence directed DNA bends.

The Guanine residues in a nearest neighbor contact with the A tracts are known to modify the bend [18]. This observation has been confirmed for all the sequences containing the A tracts. For instance, when sequence 31 (R_L = 1.14) is mutated at 17th position by G, i.e., next to the A tract, the R_L obtained is 1.12.

It can also be noted from Table II that if mutation of G by A results in a significant change in R_L then similar effect (increase or decrease) is observed if G is replaced by T and, sometimes, by C. This has been verified as follows. The R_L for sequence 16 is 1.06; if position 27 is replaced by A or T or C, the resultant R_L values are 0.92, 0.90 and 0.89, respectively. As can be seen, these are consistently lower than 1.06. This is a new observation and not reported in the earlier studies.

With the help of the trained network it is possible to study the effect of different factors that influence the DNA curvature. For this purpose, sequences listed in Table III have been considered and their R_L values were estimated. The role of phasing has been evaluated by examining a set of sequences described as $(A_5N_k)_n$; k = 4, 5, 6, 10. Each one of these sequences contains the A_5 tract flanked by C at 3' and 5', with a total of k bases intervening in the G+C - rich segment between the A_5 tracts. The series $(A_5N_5)_n$ has 10 - bp phasing that nearly matches the expected helix screw of about 10.3 bp per turn which is the average of 10.5 for B-DNA and 10.1 for poly(dA).poly(dT) in solution. It can be verified from Table III that the series $(A_5N_5)_n$ possesses largest R_L as compared to $(A_5N_6)_n$ and $(A_5N_4)_n$. Thus it can be inferred that the bending elements must be repeated in phase with the helix screw to add coherently.

To differentiate between the bending due to increased flexibility and systematic bending wherein the direction of the helix axis is altered in a definite way, the series $(A_5N_{10})_n$ may be examined. The R_L value of 0.932 for the series suggests normal gel mobility due to the formation of a zigzag structure wherein systematic bends are nearly exactly out of phase.

The importance of continuous run of A residues in determining the extent of curvature was investigated by interrupting the A_5 tract with another nucleotide N (referred to as IAN in Table III) at the central base. It is noticed that substitution by either T or C does not affect the R_L value. However, substitution by G causes decrease in the curvature (R_L value changes from 1.091 to 1.089). To check whether Guanines also contribute to the curvature, sequence (G_5N_5) was examined. The network predicted R_L for G_5N_5 (= 0.989) indicates normal gel mobility and suggests that in this particular case the purines A and G are not equivalent.

To examine the role of phasing of 5' and 3' junctions in influencing a bend, sequences A_{5-8} and A_{8-5} have been considered. It has been found that A_{8-5} is more anomalous (R_L = 1.18) and, hence, more strongly bent than A_{5-8} (R_L = 0.92). The greater anomaly in A_{8-5} implies greater bending at the 3' than at 5' junction.

The role of flanking base pairs was investigated by studying the retardation behavior of FCT and FGG sequences. The greatest degree of bending is witnessed when the 5' - flanking base is C, and the 3' - base is T (R_L = 1.09). However, if G is present at 3' and 5' ends, the effect is less pronounced (R_L = 0.975). These findings are well supported by the experimental studies by Koo et al. [13].

To summarize, in this chapter, an error-back-propagation neural network has been employed for predicting the retardation anomalies of DNA sequences. The trained network is able to evaluate the role of phasing, increased helix flexibility, run of polyA tracts, and flanking base pair effects in determining the curvature. It can also be used to examine the additive effect of multiple base substitutions. The results of this study indicate that ANNs can be successfully used as the feature detectors to study the bending characteristics of DNA sequences. In view of the excellent performance of the ANNs in capturing the local and global features, it is possible to use them as a model-free technique for the purpose of curvature predictions thus, avoiding sidetracks in designing costly experiments.

2.4 APPENDIX I: IMPLEMENTATION OF EBP ALGORITHM

The detailed numerical steps for training a two-layer EBP network having a bias neuron each in its input and hidden layers are given below. The numerical procedure assumes the pattern-mode of weight-updation and the logistic sigmoid nonlinearity at the hidden and output nodes.

Step 1. Initialize the hidden and output layer connection weights to small random values (say between –1 and +1).

Step 2. Apply the k^{th} input pattern $x_k = (x_{k0}, x_{k1}, ..., x_{kn})$ from the training set containing p patterns to the input layer nodes.

Step 3. Compute the weighted-sum of inputs (activation level) for the individual neurons in the hidden layer according to

$$net_{kj}^{h} = \sum_{i=0}^{n} w_{ji}^{h} \, x_i^k \quad ; j = 1, m$$

where net_{kj}^{h} denotes the weighted-sum for the hidden layer node j when k^{th} input pattern is applied, w_{ji}^{h} represents the connection weight between the input neuron i and hidden layer neuron j, n refers to the number of input units, and m is the number of hidden nodes.

Step 4. Transform the weighted-sum using the logistic sigmoid transfer function to get the outputs of the hidden layer nodes according to:

$$y_{kj}^{h} = \frac{1}{1 + \exp\left(-net_{kj}^{h}\right)} \quad ; j = 1, m$$

Step 5: Compute the weighted-sum of inputs (net activation) for the individual nodes in the output-layer as

$$net_{kl}^{o} = \sum_{j=0}^{m} w_{ij}^{o} y_{kj}^{h} \quad ; l = 1, s$$

where w_{lj}^{o} is the connection weight between node *l* in the output layer and node *j* in the hidden layer. As before, $\hat{y}_{ko}^{h} = 1$

Step 6: Transform the net activations of the output layer units using the logistic sigmoid function to get the respective output as

$$\hat{y}_{kl}^{o} = \frac{1}{1+\exp\left(-net_{kl}^{o}\right)} \quad ; l = 1, s$$

Step 7: Compute the scaled-error for the output-layer units as

$$d_{kl}^{o} = \left(y_{kl} - \hat{y}_{kl}^{o}\right)\hat{y}_{kl}^{o}\left(1-\hat{y}_{kl}^{o}\right) \quad ; l = 1, s$$

where y_{kl} refers to the desired output neuron l when the input vector x_k, is applied to the input nodes.

Step 8: Compute the scaled-error for neurons in the hidden layer according to

$$d_{kj}^{h} = \hat{y}_{kj}^{h}\left(1-\hat{y}_{kj}^{h}\right)\sum_{l=1}^{s} d_{kl}^{o}\, w_{lj}^{o} \; ; j = 0, m$$

Step 9: Update the weights between the output and hidden layer nodes as

$$w_{lj}^{o}(t+1) = w_{lj}^{o}(t) + \eta\, d_{kl}^{o}\hat{y}_{kj}^{h} + \alpha\left[w_{lj}^{o}(t) - w_{lj}^{o}(t-1)\right] \; ; j = 0, m; l = 1, s$$

where the training iteration number is represented by *t*, and η and α denote the learning coefficient ($0 < \eta < 1$) and the momentum parameter ($0 \le \alpha < 1$), respectively.

Step 10: Update the hidden-layer weights as given below, and implement steps (2-10) with another input pattern.

$$w_{ji}^{h}(t+1) = w_{ji}^{h}(t) + \eta\, d_{kj}^{h} x_{ki} + \alpha\left[w_{ji}^{h}(t) - w_{ji}^{h}(\text{t -1})\right] \; ; i = 0, n \; ; j = 1, m$$

In this procedure, steps (2-6) correspond to the forward pass and steps (7-10), to the reverse pass. The procedure (barring step 1) is repeated for all the input patterns in the training set until the network satisfies a prescribed convergence criterion based on a suitable measure of error.

2.5 REFERENCES

1 Trifonov, E.N. (1985) *CRC Critical Reviews in Biochemistry*, **19**, 89-106.

2 Trifonov, E.N. and Ulanovsky, L.E. (1987) In Wells,R.D. and Harvey,S.C.(eds), *Unusual DNA structures*. Springer-Verlag, Berlin, 173-187.

3 Trifonov, E.N. (1991) *Trends Biol.Sci.*, **16**, 467-470.

4 Fisher, M.P. and Dingman, C.W. (1971) *Biochemistry*, **10**, 1895-1899.

5 Lumpkin, O.J. and Zimm, B.H. (1982) *Biopolymers*, **21**, 2315-2316.

6 Marini, J.C., Levene, S.D., Crothers, D.M. and Englund, P.T. (1982) *Proc. Natl. Acad. Sci. USA*, **19**, 7664-7668.

7 McNamara, P.T., Bolshoy, A., Trifonov, E.N. and Harrington, R.E. (1990) *J. Biomol. Str. Dyn.*, **8**, 529-538.

8 Bolshoy, A., McNamara, P., Harrington, R.E. and Trifonov, E.N. (1991) *Proc. Natl. Acad. Sci. USA*, **88**, 2312-2316.

9 Dlakic, M. and Harrington, R.E. (1996) *Proc. Natl. Acad. Sci. USA*, **93**, 3847-3852.

10 Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1986) *Nature*, **323**, 533-536.

11 Rumelhart, D.E. and McClelland, J.L. (1986) *Parallel and Distributed Processing : Explorations in the Microstructure of Cognition*. MIT Press, Cambridge, MA.

12 Nair, T.M. (1996) In *Elements of Artificial Neural Networks With Selected Applications in Chemical Engineering and Chemical and Biological Sciences* by Tambe, S.S., Kulkarni, B.D. and Deshpande, P.B., Simulation and Advanced Controls, Inc., Louisville.

13 Koo, H.-S., Wu, H.-M., and Crothers, D.M. (1986) *Nature*, **320**, 501-506.

14 Nair, T.M., Tambe, S.S. and Kulkarni, B.D. (1994) *FEBS Lett.*, **346**, 273-277.

15 Simpson, P. (1990) *Artificial Neural Systems: Foundations, Paradigms, Applications and Implementations*, Pergamon Press., New York.

16 Freeman, J.A. and Skapura, D.M. (1992) Neural Networks Algorithms, Applications, and Programming Techniques. Addison-Wesley Publishing Company, Inc.

17 Milton, D.L., Casper, M.L. and Gesteland, R.F. (1990) *J. Mol. Biol.*, **213**, 135-140.

18 Milton, D.L. and Gesteland, R.F. (1988) *Nucl. Acids. Res.*, **16**, 3937-3949.

Table I: Network predicted R_L in comparison with Experimental R_L of various sequence units

Sequence No.	Unit	Experimental	Network Predicted
	Circles		
01	TCTCTAAAAAATATATAAAAA	0.59 (0.06)	0.54
02	TCAAATTGGGGGAAAGATCCC	0.51 (0.05)	0.55*
03	GGGCAAAAAACGGCAAAAAAC	0.52 (0.05)	0.56
	AA-containing and control fragments		
04	CTTTTAAAAG	1.01 (0.03)	1.02
05	GTTTTAAAAC	1.01 (0.03)	1.01
06	GGGTCGACCC	1.00 (0.02)	1.05*
07	GGCAACAACG	1.01 (0.02)	1.09*
08	GGCAAGAACG	1.04 (0.04)	1.09*
09	GGCAATAACG	1.06 (0.04)	1.09
10	GGCCAAACCG	1.14 (0.06)	1.09
11	GGGCAAAAAACGGCAAAAAAC	1.43 (0.03)	1.20
12	GGCTGGGCAAAAAACGGGCAAAAAACGGCAAAAAACGGCTCC	1.26 (0.03)	1.16*
13	GGCTGGGCAAAAAACGGCAAAAAACGGCTCC	1.19 (0.03)	1.18
14	GGCTGGGCAAAAAACGGCTCC	1.14 (0.03)	1.17
15	GGCAGGGTCGGGCAAAAAACGGCTGGATCCC	1.07 (0.03)	1.03*
16	GGCAGGGCGGTCGACGGGCAAAAAACGGCGTCGGGCGGATCC	1.06 (0.03)	1.06
17	GGGCAAAAACGCCAAAATTTTGCCGCGGGCC	1.11 (0.03)	1.12*
18	GGGCAAAAACGGGCGGCCAAAATTTTGCCGC	1.01 (0.02)	1.00
19	AAAAAAATTTTTTTTTTAAA	1.00 (0.02)	0.97*
20	AAAAAAAAAAAAAAAAAAAA	0.98 (0.03)	1.02
21	TCTCCTTCTTGGTTCTCTTCTC	1.00 (0.02)	1.00
22	CCCCCGGGGG	1.05 (0.06)	1.04
23	GACAGGACTC	1.01 (0.03)	1.00
24	CCATCGATGG	0.98 (0.03)	0.98
25	CGGGATCCCG	1.00 (0.02)	0.99

26	GCGGGTAGTTTTTTCCTACAC	1.13 (0.02)	1.12
27	GCGCGATTTTTACGAAAAAAA	1.25 (0.02)	1.18
28	GGCTGGGCAAAAAACGGCTCC	1.14 (0.02)	1.17
29	ACCTGGGCAAAAAACGGCTCC	1.14 (0.02)	1.14
30	GGCTCACCAAAAAACGGCTCC	1.12 (0.02)	1.18*

Table I continued ...

Sequence No.	Unit	Experimental	Network Predicted
31	TCACTTATATAAAAAATATAT	1.13 (0.02)	1.14
32	TCGCTTATATAAAAAATATAT	1.13 (0.02)	1.13
33	GCCCCTAAAAAGCCCCTTTTA	1.12 (0.02)	1.14
34	GTGGGACAAAGTGCCCACAAA	1.06 (0.02)	1.06
35	CTGTGAAAAAACACACTTTTT	1.13 (0.02)	1.13*
36	AAAAACACACAAAAAACACAC	1.29 (0.02)	1.14
37	TTTTAAAAAC	0.99 (0.04)	0.98
38	GGCCTTTTTAAAAACCGGGCC	1.03 (0.03)	1.03
39	GGCCTTTTTAAAAAAACCGCC	1.07 (0.03)	1.06
40	GGCCTTTTTAAAAAAAAACCC	1.15 (0.03)	1.18
41	GGCCTTTTTTTAAAAAAACCC	1.21 (0.03)	1.18
42	CGGAGCCGTTTTTTGCCCAGC	1.15 (0.03)	1.13
43	CCGGCCAAAAAAAACGCGCGC	1.09 (0.03)	1.07*
44	CCGGCCAAAAAAAAAACGCGC	1.04 (0.03)	1.04
45	CCGGCCAAAAAAAAAAAACGC	1.01 (0.03)	1.02
46	CCGCCAAAAAAAAAAAAAACG	1.05 (0.03)	1.03
47	CCGCAAAAAAAAAAAAAAAC	1.07 (0.03)	1.12
	Non-AA fragments		
48	CATGTCACCGACGCATCACCG	1.07 (0.02)	1.09*
49	TCCCCAGACGTCCCCAGCACG	1.02 (0.02)	1.00
50	GCGAGAGGGTACGGACATCTC	1.10 (0.02)	1.21
51	TGTGAGAGGGGCATGAGATCA	1.11 (0.02)	1.10
52	TACGGATCTCGCATGACTCTC	1.06 (0.02)	1.05
53	CGGAGCTATCCGGAGCCTATC	1.07 (0.02)	1.20

54	GGAGAGCTCACACGACTAGTC	1.03 (0.02)	1.11*

Table II: Simulated R_L values for effective single base substitution.

Seq. no.	Effective mutation	Retardation anomaly	Seq. no.	Effective mutation	Retardation anomaly
8	GA*2	1.05	17	GA*11	1.16
	GT*2	1.05		GA*27	1.02
	AG*4	1.03		GT*11	1.16
	AG*8	1.13		GT*27	1.00
	GC*2	1.04		AG*14	1.18
10	GA*2	1.05		CG*30	0.90
	GT*2	1.05		GC*11	1.16
	CG*4	1.03		GC*27	1.00
	CG*8	1.13	31	AG*14	1.19
	GC*2	1.05		TG*19	0.95
14	TG*19	1.04	52	GA*11	1.11
16	GA*27	0.92		CA*19	1.02
	GA*30	1.17		GT*11	1.12
	GT*27	0.90		GT*15	1.02
	GT*30	1.18		TG*14	1.16
	AG*14	1.15		CG*19	0.79
	CG*19	0.80		GC*11	1.12
	GC*27	0.89		GC*15	1.02
	GC*30	1.18			[i]

Table III: Simulated R_L values for specific sequence patterns.

Name	Sequence (5' to 3')	Network Predicted R_L
A_5N_4	CAAAAACGG	1.049
A_5N_5	GGCAAAAACG	1.091
A_5N_6	GGCCAAAAACG	1.065
A_5N_{10}	CCGGCAAAAACGGGC	0.932
IAC	GGCAACAACG	1.091
IAG	GGCAAGAACG	1.089
IAT	GGCAATAACG	1.091
G_5N_5	TCGTGGGGGC	0.989
$A_{5\text{-}8}$	CCAAAAACGGGCAAAAAAAA	0.915
$A_{8\text{-}5}$	CCAAAAAAAACGGGCAAAAA	1.181
FCT	GGCAAAAATG	1.090
FGG	CCGAAAAAGG	0.975

GA*20 means that the guanine is replaced by adenine at position 20.

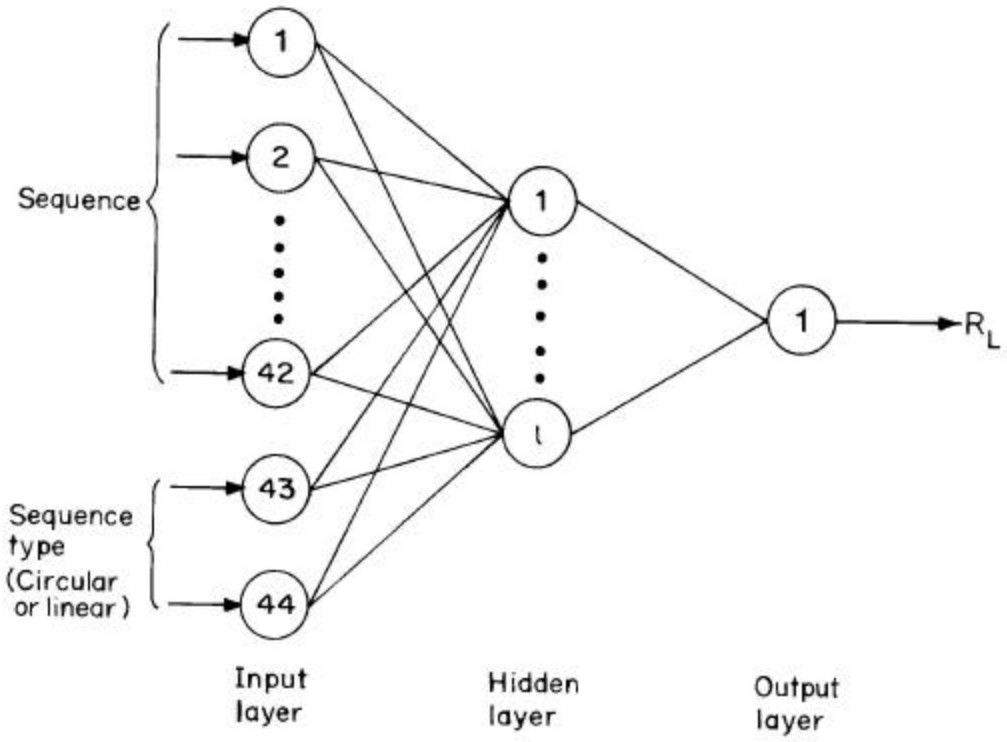


Figure 2-1: Network Architecture used in the simulation: 44 neurons in the input layer, one hidden layer consisting of one neuron, and one neuron in the output layer. The trained network approximates y = f(x), where x and y represent the input (DNA sequence) and the output (R_L value).

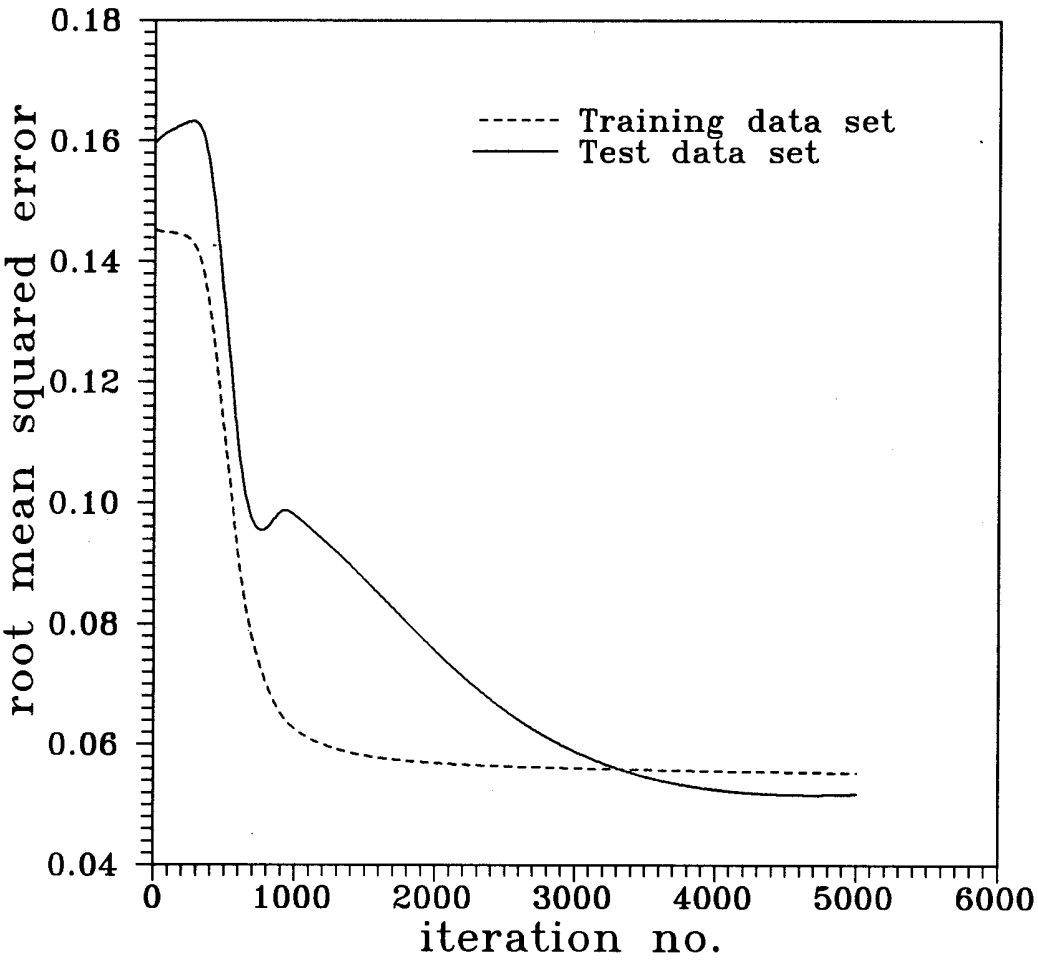


Figure 2-2: RMSE profiles corresponding to the training and test data sets.

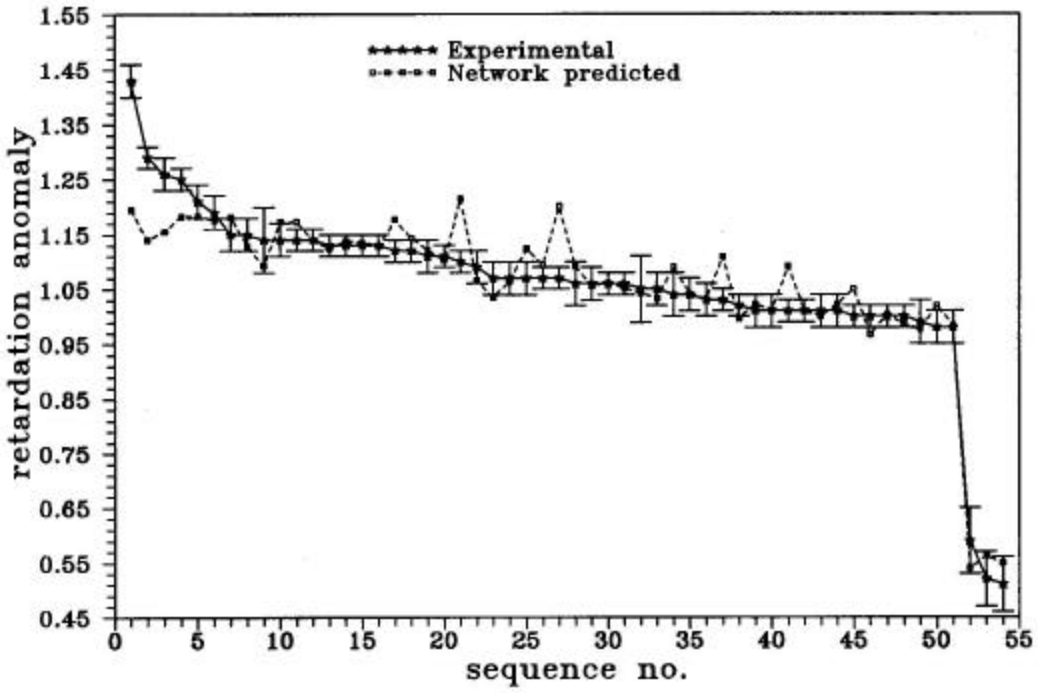


Figure 2-3: Graphical comparison of experimental and network predicted retardation anomalies. Sequence numbers are arranged in descending order of experimental R_L values. Experimental R_L shown with their error bars.

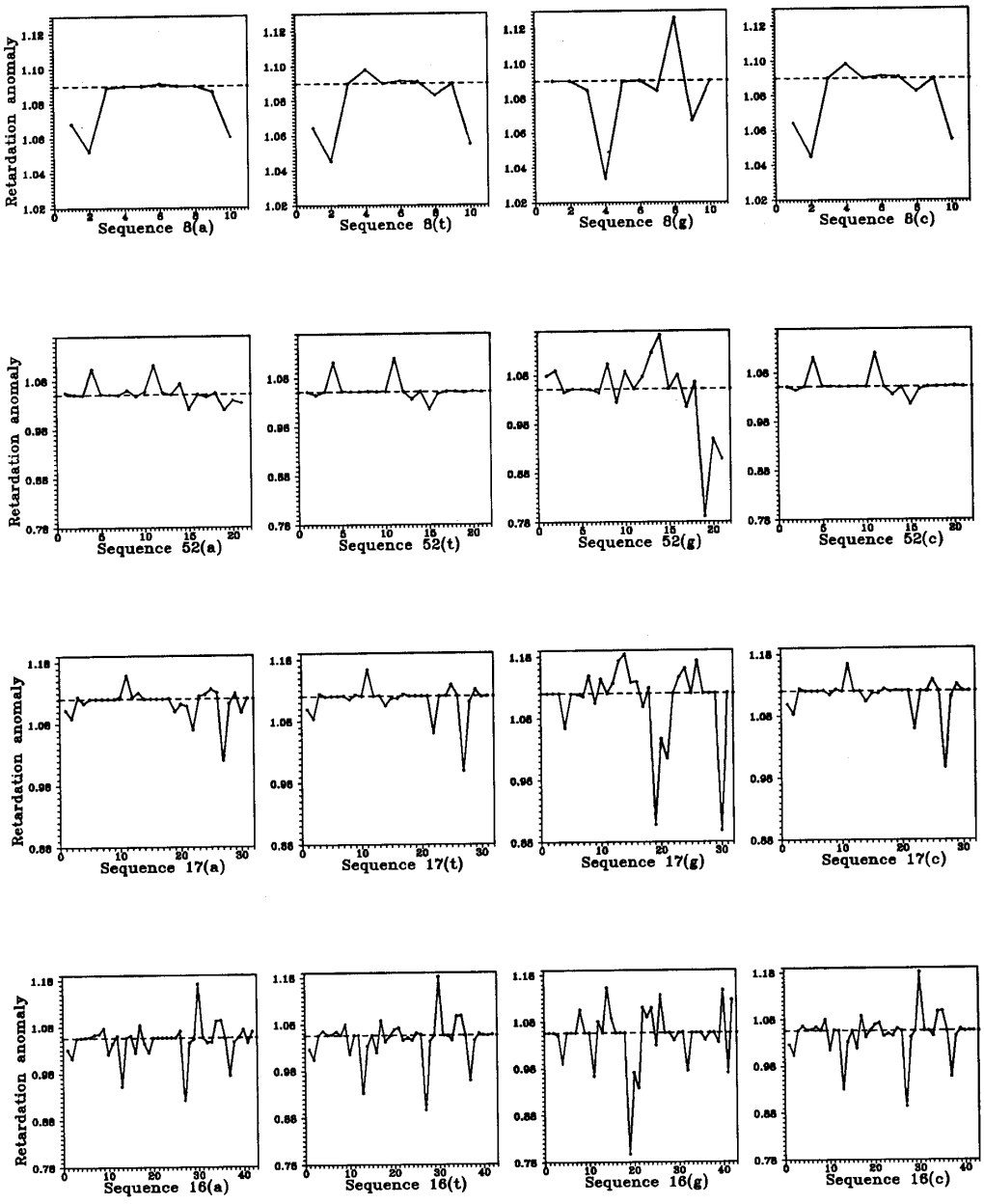


Figure 2-4: Graphical representation of the R_L values for all possible single base substitutions for sequence numbers. 8, 52, 17 and 16 having base pairs of length 10, 21, 31 and 42, respectively.

CHAPTER

ANN modeling of DNA sequences: new strategies using DNA shape code

Two new encoding strategies, namely, *wedge* and *twist* codes, which are based on the DNA helical parameters, are introduced to represent DNA sequences in artificial neural network (ANN)-based modeling of biological systems. The performance of the new coding strategies has been evaluated by conducting three case studies involving mapping (modeling) and classification applications of ANNs. The proposed coding schemes have been compared rigorously and shown to outperform the existing coding strategies especially in situations wherein limited data are available for building the ANN models .

3.1 INTRODUCTION

In the last decade, artificial neural networks (ANNs) have been extensively used in the analysis of nucleic acid sequences (see review [1]); the main reason being their ability of recognizing and classifying patterns not only from the quantitative data but also from the qualitative data such as DNA sequences. These ANN abilities have been used in various classification applications in biological sciences e.g., analysis of *E. coli* promoter structures [2], prokaryotic transcription terminator prediction [3], and identification of *E. coli* ribosome binding sites [4]. ANNs have also been used in mapping (modeling) applications, for instance, in the analysis of transcription control signals [5] and DNA curvature [6], where the objective was to identify the functional relationship(s) between a DNA sequence and its property.

Of all the different ANN architectures, the one with a multilayered feedforward structure and trained using the error-back-propagation (EBP) algorithm [7] represents the most widely used network paradigm. The EBP network (EBPN) mostly comprises three layers (input, hidden and output) of interconnected neurons (also termed as "processing elements" or "nodes") and learns the relationship between its inputs and outputs via a procedure called "network training". The peculiarity of the EBP algorithm is that it trains nonlinear multilayered networks wherein a nonlinear activation function is used for the computation of the outputs of the hidden and/or output nodes. The nonlinear neural networks are preferred over the linear ones for modeling high dimensional systems since the input-output relationships in such systems are often nonlinear. In a typical ANN-based mapping (classification) application, the network input-output is an appropriately coded DNA sequence and its property (class), respectively. The EBPN training involves minimization of an error function using a steepest descent strategy [7-9] such as the *generalized delta rule* (GDR) wherein the network output is compared with its desired (target) value and the difference (error) is used to iteratively modify the strengths (weights) of the interneuron connections. Network training, following convergence, produces weights that can be considered to be the parameters of the converged ANN model. These weights can then be used to make predictions corresponding to the new DNA sequences, which were not part of the data employed during the development of the network model.

In ANN-based DNA sequence studies, an individual nucleotide of a sequence is represented using three main coding strategies viz., CODE-2, CODE-4 and the EIIP. The first two of these mononucleotide-based coding schemes use binary representation and, therefore, possess purely empirical character. The EIIP-code [3] on the other hand uses a nucleotide-specific physical property, namely, the Electron Ion Interaction Potential (EIIP) for input coding and, therefore, has a sound theoretical basis. In CODE-2 and CODE-4 approaches, each nucleotide is represented by two (00=A, 01=T, 10=G, and 11=C) and four (0001=C, 0010=G, 0100=A, and 1000=T) binary digits, respectively, whereas in EIIP-code nucleotides are characterized by their unique EIIP values (0.1260=A, 0.1335=T, 0.0806=G, 0.1340=C). Thus, CODE-2, CODE-4 and EIIP strategies require two, four and one neurons, respectively, to represent a nucleotide. Since the data requirement to train an ANN increases as the number of neurons in the network increases, the CODE-4 strategy needs maximum number of data points as compared to the CODE-2 and EIIP strategies with the EIIP-code needing minimum number of data points. According to a thumb rule, the number of data points required for network training equals the number of network connection weights although reasonably satisfactory results have been obtained with lesser data points. This may be due to the intrinsic dimensionality of the system being much lower than its apparent dimensionality. More often than not, the available training data is insufficient and, hence, schemes requiring fewer neurons to code a nucleotide sequence are desirable. With this objective, we introduce here two coding strategies, namely, the *wedge* code and the *twist* code requiring just one value for the representation of a dinucleotide, in the ANN-based modeling of DNA sequences. The performance of the proposed strategies has been tested by conducting three case studies: (i) prediction of DNA curvature, (ii) prediction of the promoter strength of various promoters transcribed by *E. coli* RNA polymerase, and (iii) prokaryotic transcription terminator prediction. While the first two case studies are the mapping applications of ANNs, the third one involves an ANN-based classification.

3.1.1 Philosophy of wedge and twist codes

There exist several helical parameters describing the DNA structure [10] that are based on translation and rotation. In this study, we shall consider the parameters based on the wedge model, which are estimated from the experimental gel retardation data of Bolshoy et al. [11]. The DNA helical parameters characterizing the wedge (deflection) angle (σ), twist angle (Ω) and the direction of deflection angle (δ) are known as *DNA shape code*. These Eulerian angles are functions of the dinucleotides i.e., adjacent base pairs in a DNA molecule. The dinucleotides AA ($5^{'}$-AA-$3^{'}$ on one strand) and TT (on the opposing strand) together form two stacked A•T base pairs so that the wedge and twist angle values are equal for the AA and TT dinucleotides. Similarly, dinucleotide pairs AC & GT, AG & CT, CA & TG, CC & GG, and GA & TC have equal magnitudes for the wedge and twist angles. For a detailed discussion of the specific features of these angles, the reader is directed to Kabsch et al. [12], Bolshoy et al. [11], and Shpigelman et al. [13]. To have a unique dinucleotide-specific value for the wedge and twist codes, the sign of the direction of deflection angle can be ascribed to the values of the wedge and twist angles, since the direction angle δ changes its sign for the complementary dinucleotides. The wedge and twist code values obtained thereby are listed in Table I. Since these codes incorporate the structural and physical properties of dinucleotides, they have a sound theoretical basis and, therefore, can be employed to replace the arbitrary coding strategies such as the CODE-2 and CODE-4. As compared to the EIIP-code, which among the existing strategies requires least (one) number of input neurons to represent a nucleotide, the use of wedge and twist codes reduce the input space of an ANN by half thereby leading to a smaller network and, consequently, requiring a smaller data set for training the network. This chapter is organized as follows. First, procedural details of the ANN-based modeling along with the strategies for optimizing the network architecture and weights are outlined. Next, the results of three ANN-based case studies wherein the proposed codes have been utilized for the dinucleotide representation are presented. Specifically, the results obtained by using the wedge and twist codes are compared with those obtained using the CODE-4 and EIIP coding strategies. The CODE-2 scheme has not been considered for comparison since the CODE-4 strategy has been found to outperform the CODE-2 strategy [14]. The performance of the two new codes is also compared with a dinucleotide-based random

strategy wherein the 16 possible dinucleotide combinations are coded by equally spaced real numbers in [0,1] range as given by: 0.0625=AA, 0.125=AC, 0.1875=AG, 0.25=AT, 0.3125=CA, 0.375=CC, 0.4375=CG, 0.50=CT, 0.5625=GA, 0.625=GC, 0.6875=GG, 0.75=GT, 0.8125=TA, 0.875=TC, 0.9375=TG and 1.00 = TT. In all the case studies, the network training and simulation procedures for the random dinucleotide coding approach are same as that for the wedge and twist codes.

3.2 MATERIALS AND METHODS

The neural networks considered are three-layered feed-forward type trained using the EBP algorithm. The logistic sigmoid transfer function has been employed at the hidden and also at the output nodes of all the networks. In a situation where sufficient training data are available for network training, all the coding schemes are likely to perform equally well. The efficiency of the proposed codes, therefore, has been tested using limited training data (case studies I and II).

A generalized EBPN architecture for the mapping and classification applications of DNA sequences is shown in Figure 3-1. The computer code for training such an EBPN was written in FORTRAN-77 and compiled using the Microsoft FORTRAN compiler for the IBM PC and compatibles.

3.2.1 Neural Network Simulation

The neural network simulations were performed on a 486 (66MHz) PC. The error function used during the network training was RMSE (refer chapter 2, section 2.2.3).

Although the objective of network training is to minimize the RMSE with respect to the training set, it does not guarantee that the trained network possesses satisfactory generalization ability. Such an ability ensures that the network is capable of predicting accurately the outputs when new inputs, which do not belong to the training set, are presented to the network. Since the weights resulting in the minimum RMSE for a representative test set ensure satisfactory generalization performance, these are considered to be the optimal weights in practice.

In general, network training (more specifically the RMSEs with respect to the training and test sets) shows sensitivity towards the number of network hidden nodes (N_H) and the GDR parameters, namely, the momentum coefficient (α), and learning rate (η). To obtain the overall optimal weights resulting in the least RMSE for the test set, several independent training runs were performed by systematically varying the number of hidden nodes and the magnitudes of the GDR parameters (α and η). For each combination of the stated parameters, additionally, the effect of the random number generator seed was examined. This is necessary for studying the effect of the randomly initialized weights whose sequence depends on the seed value of the random number generator. By changing the seed value, a different sequence of random numbers is generated and, consequently, the starting point in the weight space of an ANN gets shifted. This helps in rigorous exploration of the nonlinear error surface possessed by the EBP networks.

3.2.2 Case Study I: Prediction of DNA curvature

According to the junction model, the principal sequence feature responsible for the intrinsic DNA curvature is generally assumed to be the runs of adenines. On the other hand, the wedge model of DNA curvature considers that each dinucleotide step is associated with a characteristic deflection of the local helix axis [11]. It may however be noted that the generality of such first principle models for predicting the curvature is still being debated [15]. Thus, a practical and simpler approach is to develop an empirical model correlating a nucleotide sequence of DNA and its effective curvature. The use of ANNs for developing such models has an advantage in that they can approximate nonlinear relationships even between qualitative and quantitative data. Accordingly, this case study aims at developing an ANN model for predicting the curvature of a DNA in terms of its retardation anomaly value, which is a measure of the electrophoretic anomaly of the curved DNA reflecting the additional friction of the DNA in the gel due to curvature [16]. The relative electrophoretic mobility of most curved DNA fragments monotonously decreases with the fragment length. This is usually characterized as the ratio of the apparent to actual DNA length, and the ratio termed as the "R_L factor" is found to increase with increasing fragment length. In an earlier study [6], an ANN-based prediction of the R_L factor using the EIIP-code was successfully

conducted and the results obtained thereby have been utilized here for comparison purposes.

The data (54 sequences) comprising circular and curved, and straight synthetic fragments and their experimental R_L values were taken from the study by Bolshoy et al. [11]. The choice of such data was based on the consideration that the data set pertains to the most exhaustive experimental gel retardation study of DNA sequences. The respective experiments were carried under standard gel conditions and hence the data is ideal for EBPN training. The sequences are of uneven length that varies between 10 and 42 base pairs. Each sequence forming the network input was coded separately using the dinucleotide-specific wedge, twist and random code values. Since a single wedge/twist/random code value describes a dinucleotide, a sequence say 21 base-pair long, can be coded using ten values. To complete the coding of the entire sequence, the 21^{st} nucleotide was paired with the first one and coded accordingly. All the sequences with odd lengths were analogously coded. For CODE-4 strategy, the sequences were coded using four digit binary numbers as described earlier. It is necessary for the network training that all the input patterns are of the same length. Since the nucleotide sequences are of variable length, the shorter ones (length smaller than 42 bp) represented using the wedge/twist/random codes were uniformly padded with a small dummy number (0.01) until each short sequence becomes 21 (=42/2) units long. For CODE-4, similar padding was applied till each fragment was 168 (=42×4) units long. This is an indirect way of informing the network that the sequence position valued 0.01 does not belong to either A, T, G or C. The resulting data can be viewed as a matrix of size (54×21) for the wedge/twist/random codes, and of size (54×168) for the CODE-4. Next, each column of the (54×21) matrix was normalized so that each column element upon normalization lies between 0.05 and 0.95. While performing normalization, the padded elements of a sequence were not processed. In order to differentiate between the circular and linear sequences, two additional inputs were considered at the end position of each coded sequence. Specifically, the circular fragments were described as (0.05,0.90) and the linear ones by (0.90,0.05). Such an addition of two inputs at the end position of each coded sequence resulted in the data matrix of size (54×23) for the three dinucleotide-based codes and a matrix of size (54×170) for the CODE-4. The experimental R_L values that formed the target output for

each input pattern (coded sequence) were also normalized to lie in the [0.05, 0.95] range. Upon normalization, the data set of 54 coded sequences (inputs) and their R_L values (outputs) was divided into the training (40 patterns) and test (14 patterns) sets, respectively (see Table I from reference [6]). During network training, the training set is used for adjusting the network weights while the test set is used to evaluate the generalization performance of the network.

The optimal values of the EBPN's structural parameters, GDR parameters, and the RMSE values corresponding to the training and test sets for all the five coding strategies are listed in Table II-A. A rigorous statistical analysis has been additionally performed for comparing: (i) the predictions of the five ANN models with the experimental R_L values, and (ii) the predictions of a combination of ANN models, wherein all possible model combinations have been considered. In here, apart from computing the correlation coefficient (r_{xy}) values, we have performed the Z-test (for large sample size i.e., the number of points, $n > 30$) and the F-test. The procedures for the Z- and F-tests are described in the Appendix. The purpose of performing these tests, in essence, is to answer the query "How significant are the differences between the means and variances of the R_L predictions made by two coding strategies, namely, x and y?" The r_{xy} values along with the results corresponding to the Z- and F-tests are tabulated in Table II-B.

3.2.3 Case Study II: Prediction of promoter strength

A promoter is a start signal at the beginning of a gene or a gene cluster that directs RNA polymerase to initiate RNA synthesis. RNA polymerase measures the efficiency of transcription in terms of the promoter strength that refers to the relative rate of synthesis of the full-length RNA product from a given promoter. The transcription efficiency of a given promoter sequence is regulated by many factors such as: (i) nucleotide sequence of the -35 region, (ii) nucleotide sequence of the -10 region, (iii) spacing between the -35 and -10 regions, and (iv) nucleotide sequence especially A+T content in the 5'-flanking region upstream from the -35 region [17]. The additive rule states that the individual contributions of nucleotide sequence spacer length, deoxyribonucleic acid (DNA) conformation, and electrostatic binding within a promoter, collectively establish the total promoter strength. It can thus be noticed that a number of factors influence the strength of a promoter. Owing to

the difficulties in the experimental evaluation of the stated contributing factors, it is advantageous to build a promoter strength prediction model that does not require explicit knowledge of the various factors influencing the transcription efficiency. With this objective, we have examined the efficacy of the wedge and twist codes vis-à-vis CODE-4, EIIP and random dinucleotide codes for the ANN-based prediction of the promoter strength.

For this study, an EBPN was trained using the experimental data by Deuschle et al. [18], where *in vivo* promoter strengths of the various promoters transcribed by *E. coli* RNA polymerase have been determined. The data set comprising 14 promoter sequences and their corresponding strengths was divided into training (10 patterns) and test (4 patterns) sets, respectively (refer Table III). In these data, all but one promoter sequences are 70 nucleotides long; the remaining one is 69 nucleotides long. For ANN modeling, the sequences were coded using the wedge, twist and random code values specified earlier. For coding the 69-nucleotide long promoter sequence, the last nucleotide was paired with the last-but-one nucleotide, i.e., from the group of three nucleotides (AAG) at the sequence end, two dinucleotide pairs (AA and AG) were formed, and coded accordingly. To make all the input vectors of same size, the 69 base-pair long sequence was uniformly padded with 0.1 till it was 280 (=70×4) units long for the CODE-4 scheme and 70 (=70×1) units long for the EIIP-code. The resulting data can be viewed as a matrix of size (14×35) for the wedge, twist and random dinucleotide codes and, matrices of sizes (14×280) and (14×70) for the CODE-4 and EIIP-code, respectively. Each column element of the (14×35) and (14×70) matrices was normalized such that it lies between 0.05 and 0.95 upon normalization. The values of the experimental promoter strength that formed the target output for each input pattern were also normalized to lie in the [0.05, 0.95] range.

The five networks utilizing different coding schemes were rigorously trained and optimized as described earlier. The details of the optimized network architectures and the GDR parameters are listed in Table II-A. The table also gives the RMSE values corresponding to the training and test sets for the five coding schemes.

As in case study I, a rigorous statistical analysis has been conducted by employing the Student's t test (for small sample size, i.e., when the number of data points $n < 30$) and the F-test. The procedure for Student's t test has been described in the Appendix.

3.2.4 Case Study III: Prokaryotic transcription terminator prediction

Terminators are sequences that primarily regulate the gene expression by providing stop signals at the end of transcription units and, thus, allowing adjacent genes and/or operons to be transcribed and regulated independently [20]. Studies have shown that the factor-independent terminators shared features like G/C-rich dyad symmetry followed by a stretch of 4-8 adjacent thymine residues immediately upstream of the last nucleotide incorporated into the RNA chain. It has been witnessed that many independent terminators do not comply with the consensus pattern of the dyad symmetry and T-stretch [21] and, therefore, conditions for termination are not well defined. It is thus important to develop methods for identifying (classifying) the terminators comprising inconsistent consensus patterns. ANNs utilizing the CODE-4 and EIIP formalisms have been already found to be successful in this task [3]. Our objective in the present case study is to examine the classification efficiency of the wedge and twist codes vis-à-vis CODE-4, EIIP and the random dinucleotide coding schemes. Towards this objective, three network models utilizing wedge, twist and random codes have been developed and their classification performance is compared with the CODE-4 and EIIP code results obtained by Nair et al. [3].

The terminator sequences for the ANN simulations were taken from the compilation by Brendel et al. [22]. From a total of 128 terminators of length 51 nucleotides, 88 were chosen for training the network and the remaining 40 were used as the test data. A pseudo-random number generator was used for constructing the random sequences with equal compositions of A, T, G and C. These random sequences were combined with the terminator sequences in 1:3 ratio. The resulting 352 patterns formed the training set inputs; the test set inputs (160 patterns) were constructed analogously. Since the length of terminator sequences is an odd number (51 nucleotides), the last nucleotide was paired with the last-but-one nucleotide of the same sequence and coded accordingly. Subsequently, the column elements of the resulting matrices of size (512×26) were normalized to lie between 0.05 and 0.95. In this case study, the target output equal to one represents a terminator sequence, and the target output of zero refers to a random (non-terminator) sequence.

The three networks utilizing the wedge, twist and random dinucleotide input coding schemes were rigorously optimized following the procedure described earlier. The details of the optimized network structures and the GDR parameters along with the percentage classification accuracy for all the coding schemes can be found in Table II-A.

3.3 RESULTS AND DISCUSSION

3.3.1 Case Study I

The statistical *Z*-test checks whether or not the mean values of two large samples drawn from respective populations are statistically different. In the present context, a sample refers to a set of the R_L values either determined experimentally or those predicted by each of the five ANN models. In essence, the *Z*-test verifies the validity of the null hypothesis (H_0) that the difference in the means (μ_x and μ_y) of two populations is statistically insignificant. It can be noted (see Table II-B) from the *Z*-statistic values (Z_c) corresponding to the fifteen different combinations of x and y samples that the absolute value of Z_c is less than both $Z_{0.01}$ (=2.33) and $Z_{0.05}$ (=1.64). Thus, we may accept the null hypothesis, H_0 (with 1% and 5% levels of significance), that the differences in the respective μ_x and μ_y values are insignificant.

The *F*-test is meant for testing whether there exists a statistically significant difference between the variance values (σ_x^2 and σ_y^2) of two populations. When the *F*- test is used on the samples consisting of variance values of the experimental and CODE-4 predicted R_L values, it is seen (see Table II-B, row 1, column 9) that the absolute value of F_c (=1.40) is less than $F_{53,53,0.01}$ (=1.60), but greater than $F_{53,53,0.05}$ (=1.39). Hence, we may accept: (i) the null hypothesis (H_0), that σ_x^2 is equal to σ_y^2 at 1% level of significance and, (ii) an alternative hypothesis (H_1), that σ_x^2 is greater than σ_y^2, at 5% level of significance. For the rest fourteen combinations of samples x and y, the absolute values of the F_c are smaller than both $F_{53,53,0.01}$ and $F_{53,53,0.05}$ and, therefore, we may accept H_0 at both 1% and 5% levels of significance.

From Tables II-A and II-B, it can be noticed that the RMSE and r_{xy} values for the wedge, twist, random dinucleotide and EIIP codes are comparable, although the last one fares marginally better than the two new coding strategies. On the other hand, the RMSE

values (0.32, 0.098) corresponding to the training and the test set of CODE-4 are the highest among five coding schemes. Also, the magnitude of the coefficient of correlation (r_{xy}=0.87) between the experimental and CODE-4 based network predicted R_L value is the lowest among all coding schemes. These trends suggest that the CODE-4 is the least efficient of the five input coding strategies for the ANN-based prediction of R_L. This is consistent with the *F*-test results where it was observed that the variances in respect of the experimental and CODE-4 based network predicted R_L values are different at 5% level of significance. The result indicates that the CODE-4 based model has not captured the variations in the experimental R_L values with statistically significant accuracy. It can be also noticed from the r_{xy} values listed in Table II-B (column 7, entries 3, 4 and 5) that the wedge and twist codes perform better, albeit marginally, than the random dinucleotide code. These wedge and twist code results essentially indicate that the codes possess good potential as sequence coding schemes since both the strategies resulted in relatively high r_{xy} values ($\geq$ 0.92) and low RMSE values ($\leq$ 0.069) for the test set. Also, the mean (1.06) and variance (0.016) values associated with the R_L- predictions of the ANNs using these codes are statistically consistent with the mean (1.05) and variance values (0.021) of the experimental R_L values.

While coding the DNA sequences in this case study, the effect of overlapping dinucleotides was not taken into account though it is well known that the curvature of a sequence depends on the overlapping dinucleotides. Such a simplified coding approach though leaves out half of the relevant information contained in a sequence, was used still with a view of keeping the complexity of the coding procedure to a bare minimum. To check whether this simplification has any effect on the prediction accuracy of the trained network, we performed a control study for the networks utilizing the wedge and twist codes. In here, the first nucleotide was removed from each DNA sequence (Table I from chapter 2), and the remaining portion of the sequence was coded using wedge and twist codes. The resultant input patterns are different from those wherein the first nucleotide was retained during sequence coding. These input patterns were then used to re-predict the R_L values for which the optimal weights obtained originally were utilized. It was observed that the re-predicted R_L values match their desired (experimental) values with the same accuracy as obtained when first nucleotide was considered for the input coding. The

correlation coefficient for the experimental and repredicted R_L values for the wedge and twist codes were found to be 0.93 and 0.924, respectively, which almost match those listed in Table II-B (0.931 and 0.92). It can thus be inferred from the results of control simulations that it is not essential in ANN-based R_L-prediction studies to account explicitly for the overlapping dinucleotides.

While partitioning the available data (54 patterns), a care was exercised that the 14 examples in the test set are the true representatives of the 40 examples in the training set. It is however essential to verify whether the available data was adequate at all for effecting the said partition. Accordingly, "cross-validation" simulations were performed using the *leave-k-out* methodology. In this approach, the entire set of available data is randomly divided into N subsets each comprising k patterns. Next, the network is trained N times using each subset in turn as the test set with the remaining (N-1) subsets collectively representing the training set. Upon completing this exercise, the RMS errors corresponding to the training and test sets are averaged; the mean RMSE in respect of the test set gives an estimate of the overall network performance that could be achieved if more data were available for the network training.

For performing the above-described cross-validation simulations, the available data of 54 DNA sequences and their corresponding R_L values were partitioned into six subsets (N = 6, k = 9). The results of the cross-validation simulations in respect of the five coding schemes are presented in Table II-C. A comparison of the test set RMSE values listed in Tables II-A and II-C indicates that the cross-validation results are better only in the case of CODE-4 scheme. This result suggests that the available data of 54 patterns was adequate for all the network models except the one using the CODE-4 coding scheme. The result is a natural consequence of the CODE-4 scheme producing largest (as compared to other codes) sized networks, thus needing more training data.

3.3.2 Case Study II

In this case study also, a rigorous statistical analysis was performed on the promoter strengths predicted by the five ANN models. The results of the Student's t and F- tests conducted thereby on the sample sets comprising experimental and ANN-predicted promoter strengths are tabulated in Table IV. It is noted from the various Table

IV entries that for all the fifteen different combinations of x and y samples, the absolute values of t_c are less than $t_{2\alpha}$ (=1.315) and $t_{2\alpha}$ (=1.706), which correspond to 1% (α=0.01) and 5% (α=0.05) levels of significance, respectively. Thus, we may accept the null hypothesis (H_0) that the mean values (μ_x and μ_y) of the respective populations are statistically equal in all the fifteen combinations of x-y sample sets at 1% and 5% levels of significance.

The *F*-statistic (F_c) values (see column 9) corresponding to the two combinations of x and y involving experimental promoter strengths and those predicted by the CODE-4 and EIIP based networks indicate that the respective F_c magnitudes (4.96 and 8.52) are greater than both $F_{13,13,0.01}$ (=2.42) and $F_{13,13,0.05}$ (=3.59). This result in essence suggests that the variance value (534.99) in respect of the experimental promoter strengths is greater (at 1% and 5% significance levels) than the variance values, 107.78 and 62.78, corresponding to the predictions of the CODE-4 and the EIIP code based networks. Since the absolute values of F_c for the remaining thirteen combinations of x and y samples are always less than $F_{13,13,0.01}$ (=2.42) and $F_{13,13,0.05}$ (=3.59), we may accept the null hypothesis (H_0) that the respective variances are equal at both 1% and 5% levels of significance.

As can be noticed from Table II-A, the RMSE values for the test sets of the wedge and twist coded networks are the lowest and the second lowest, respectively. Also, the r_{xy} magnitudes (refer Table IV, column 7, entries 3 and 4) for the predictions made by the wedge and twist code based networks are very high ($\cong$1). These results suggest that the networks utilizing the two codes have near-accurately approximated the relationship between a DNA sequence and its promoter strength. In comparison, the prediction performance of CODE-4 (r_{xy}= 0.63) and EIIP (r_{xy}=0.75) strategies is very poor. This conclusion is consistent with the *F*-test results where it was observed that the sample variances of the experimental, and CODE-4 and EIIP based network predicted promoter strength values are different at both 1% and 5% levels of significance. The result indicates that the CODE-4 and EIIP based models have not captured the variations in the experimental promoter strength values with statistically significant accuracy. Among the three dinucleotide coding schemes, the random dinucleotide coding approach (r_{xy}=0.96) performs only marginally worse than the other two (wedge and twist) schemes. A plausible explanation for such a behavior is: since the random code - unlike wedge and twist codes -

does not explicitly take into account any DNA sequence dependent property or characteristic (such as the curvature), it fails to predict with comparable accuracies.

A graphical comparison of the experimental and the network-predicted promoter strengths (P_{bla} units) for the training and test sets of the wedge code is shown in Figures. 3-2(a) and 3-2(b), respectively, wherein for clarity the promoter strengths are arranged in the descending order of their magnitudes. A similar comparison for the twist code is depicted in Figures. 3-2(c) and 3-2(d).

The cross-validation test was performed for this case study also wherein the available data of 14 patterns was partitioned into seven (N = 7) subsets each comprising two (k = 2) patterns. The results of the cross-validation simulations using the "leave-2-out" scheme are given in Table II-C. A comparison of the cross-validation results with those in Table II-A for the test set indicates that the RMSE values corresponding to the cross-validation simulations are lower for all the codes. This suggests that the prediction performance of all the five networks can improve further if more data are available for training the networks. It can however be inferred from the approximately equal RMSE values for the wedge (0.036 and 0.033) and twist (0.05 and 0.045) codes (see Tables II-A and II-C) that such an improvement, though possible, can only be marginal. In essence, the results of this case study indicate that the dinucleotide coding schemes fare better than the mononucleotide based schemes (CODE-4 and EIIP). The results corresponding to the wedge and twist codes are important in the sense that even under extreme paucity of the training data, the two new coding strategies have performed significantly better than the existing ones.

3.3.3 Case Study III

In this case study, which examines the performance of wedge and twist codes for classification applications, the accuracy of classification is defined as the percentage of correctly classified input patterns; for a given input sequence, the network output in [0.5,1.0] range signifies a terminator, otherwise it is regarded as a random sequence. The network utilizing the random dinucleotide code was found to possess poorest classification accuracy as it could correctly classify only 120 (75%) of the 160 test patterns and 270 (76.7%) of the 352 training patterns. On the other hand, the wedge and twist code based networks could correctly classify 148 (92.5%) and 140 (90%) test patterns, and 335

(95.17%) and 336 (95.45%) training patterns, respectively. Although the classification accuracy of the wedge and twist codes for the test patterns is reasonably good, it is lower than that obtained using the EIIP (95.62%) and CODE-4 (98.12%) schemes. In the classification study by Nair et al. (1994), a similar observation has been made where it was found that the CODE-4 strategy fares better than the EIIP-code. The higher classification accuracy of CODE-4 was attributed to the larger EBP network size, which means larger parameter space as compared to the EIIP-code. This explanation also holds when the classification accuracies corresponding to the CODE-4 and EIIP schemes are compared with those of the wedge and twist codes. It can thus be observed from Table II-A that as the size of the network's input space decreases (CODE-4 > EIIP-code > wedge / twist codes), the classification accuracy for the test patterns decreases accordingly (98.12% > 95.62% > 92.5%/90%). Notwithstanding this observation, it is important to note that the performance of the wedge and twist coding schemes is still acceptable since on an average 91.25% of the test patterns have been correctly classified.

3.4 CONCLUDING REMARKS

In this chapter, two input coding strategies namely, wedge code and twist code have been introduced for representing dinucleotides in the ANN-based modeling of DNA sequences. These codes make use of the helical parameters namely, the wedge angle, twist angle, and the direction of deflection angle of a DNA. The principal advantage of the new coding strategies over the commonly used mononucleotide-based coding schemes such as CODE-4 and EIIP, is that they reduce the network's input dimensionality to one-eighth as compared to the CODE-4 strategy, and to one-half as compared to the EIIP scheme. Consequently, a smaller network that can be trained faster results. Such a network i.e., possessing less adaptable parameters (weights), in general possesses better generalization capability than the network with more parameters. The efficiency of the proposed strategies vis-à-vis other input coding schemes namely, CODE-4, EIIP and random dinucleotide code, has been evaluated by conducting three case studies involving ANN-based mapping and classification applications. In all the case studies, both the proposed coding strategies have been found to perform equally well. Also, the proposed codes have been found to perform better than the conventional strategies especially when the training data was limited

(case studies I and II). In these studies, although the CODE-4 scheme that results into large input dimensionality did not perform well, the proposed codes with smaller input dimensionality have lead to some significant results. This feature of the proposed schemes is important since for many real systems the available data are often limited and generation of additional data can be an involved and costly task. It has been also observed that the networks using the wedge and twist codes fare better (i.e., yield higher correlation coefficient magnitudes and classification accuracy) than the networks using the random dinucleotide code. Such a superior performance may be attributed to the DNA shape related property i.e., the helical parameters of a DNA used by the wedge and twist codes. Since the proposed codes are sufficiently general, they can also be used for representing DNA sequences in "non-ANN-based" mapping and classification applications. The present work has also opened up a new gateway for tri- and tetra-nucleotide based DNA coding strategies.

3.5 APPENDIX-II

In the following, the computational procedures for evaluating *Z*, *F* and the Student's *t* statistics are described.

(A) Z- test (for large sample, i.e. when the number of data points, *n*, exceeds 30)

This test, also known as the *Normal test*, checks whether the difference between two population means is statistically significant. In this test, *Z* statistic (Z_c) is computed to test the null hypothesis (H_0): the means μ_x and μ_y of two populations are equal (i.e., $\mu_x = \mu_y$), against an alternative hypothesis, $H_1: \mu_x > \mu_y$. The Z_c is evaluated as:

$$Z_c = \frac{\bar{X} - \bar{Y}}{\sqrt{\frac{s_x^2}{n_x} + \frac{s_y^2}{n_y}}} \qquad \text{(I)}$$

where $\bar{X}$ and $\bar{Y}$ are the means of population samples *x* and *y*, respectively; s_x^2 and s_y^2 refer to the variances of x and y, respectively, and n_x, n_y denote the respective sample sizes. The decision rule for the *Z*-test at α% level of significance is given as:

If $|Z_c| \geq Z_\alpha$, *then* reject H_0; *otherwise* accept H_0.

(B) F- test

Similar to the *Z*-test for two means, the *F*-test is performed to check the validity of hypothesis involving two population variances (s_x^2 and s_y^2). The *F* statistic (F_c) is computed as given below to validate the null hypothesis (H_0): $\sigma^2_x = \sigma^2_y$, against an alternative hypothesis (H_1): $\sigma^2_x > \sigma^2_y$.

$$F_c = \frac{s_x^2 / n_x}{s_y^2 / n_y} \qquad \text{(II)}$$

The decision rule for the *F*-test at α% level of significance and for (n_x-1), (n_y-1) degrees of freedom is:

If $|F_c| \geq F_{(n_x-1),(n_y-1),\alpha}$, *then* reject H_0; *otherwise* accept H_0.

(C) Student's t test (for small sample size i.e., $n \leq 30$)

In an event when the sample size is small ($n \leq 30$), Student's t test is performed to check the validity of the null hypothesis (H_0): $\mu_x = \mu_y$, against an alternative hypothesis (H_1): $\mu_x > \mu_y$. The corresponding t-statistic (t_c) is evaluated as:

$$t_c = \frac{\overline{X} - \overline{Y}}{s\sqrt{\frac{1}{n_x} + \frac{1}{n_y}}} \qquad \text{(III)}$$

where

$$s = \sqrt{\frac{n_x s_x^2 + n_y s_y^2}{n_x + n_y - 2}} \qquad \text{(IV)}$$

Note that the test statistic t_c follows Student's t distribution with (n_x+n_y-2) degrees of freedom. The decision rule for the t- test at α% level of significance is:

If $\left| t_c \right| \geq t_{2\alpha}$, *then* reject H_0; *otherwise* accept H_0.

3.6 REFERENCES

1. Nair, M. (1996) In *Elements of Artificial Neural Networks with Selected Applications in Chemical Engineering and Chemical and Biological Sciences*, by Tambe, S., Kulkarni, B. and Deshpande, P., pp 395-437. Simulation and Advanced Controls, Inc., Louisville, KY.
2. Mahadevan, I. and Ghosh, I. (1994) *Nucl. Acids Res.*, **22**, 2158-2165.
3. Nair, M., Tambe, S. and Kulkarni, B. (1994) *FEBS Lett.*, **346**, 273-277.
4. Bisant, D. and Maizel, J. (1995) *Nucl. Acids Res.*, **23,** 1632-1639.
5. Nair, M., Tambe, S. and Kulkarni, B. (1995) *Comp. Applic. Biosci.*, **11**, 293-300.
6. Parbhane, R., Tambe, S. and Kulkarni, B. (1998) *Bioinformatics*, **14**, 131-138.
7. Rumelhart, D., Hinton, G. and Williams, R. (1986) *Nature*, **323**, 533-536.
8. Rumelhart, D. and McClelland, J. (1986) *Parallel and Distributed Processing: Explorations in the Microstructure of Cognition*. MIT Press, Cambridge, MA.
9. Freeman, J. and Skapura, D. (1992) *Neural Networks Algorithms, Applications, and Programming Techniques*. Addison-Wesley, Reading, MA.
10. Dickerson, R.E. et al. (1989) *J. Mol. Biol.*, **205**, 787-789; *EMBO J.* **8**, 1-4; *J. Biomol. Struct. Dyn.* **6**, 627-630.
11. Bolshoy, A., McNamara, P., Harrington, R. and Trifonov, E. (1991) *Proc. Natl. Acad. Sci. USA*, **88**, 2312-2316.
12. Kabsch, W., Sander, C. and Trifonov, E. (1982) *Nucl. Acids Res.*, **10**, 1097-1104.
13. Shpigelman, E., Trifonov, E. and Bolshoy, A. (1993) *Comp. Applic. Biosci.*, **9**, 435-440.
14. Demeler, B. and Zhou, G. (1991) *Nucl. Acids Res.*, **19**, 1593-1599.
15. Dlakic, M. and Harrington, R. (1996) *Proc. Natl. Acad. Sci. USA*, **93**, 3847-3852.
16. Marini, J., Levene, S., Crothers, D. and Englund, P. (1982) *Proc. Natl. Acad. Sci. USA*, **19**, 7664-7668.
17. McClure, W. (1985) *Annu. Rev. Biochem.*, **54**, 171-204.
18. Deuschle, U., Kammerer, W., Gentz, R. and Bujard, H. (1986) *EMBO J.*, **5**, 2987-2994.
19. Knaus, R. and Bujard, H. (1988) *EMBO J.*, **7**, 2919-2923.

20. von Hippel, P.H., Bear, D., Morgan, W. and McSwiggen, J. (1984) *Annu. Rev. Bioch.*, **53**, 389.
21. Brendel, V. and Trifonov, E. (1984) *Nucl. Acids Res.*, **12**, 4411-4427.
22. Brendel, V., Hamm, H. and Trifonov, E. (1986) *J. Biomol. Struct. Dyn.*, **3**, 705-723.

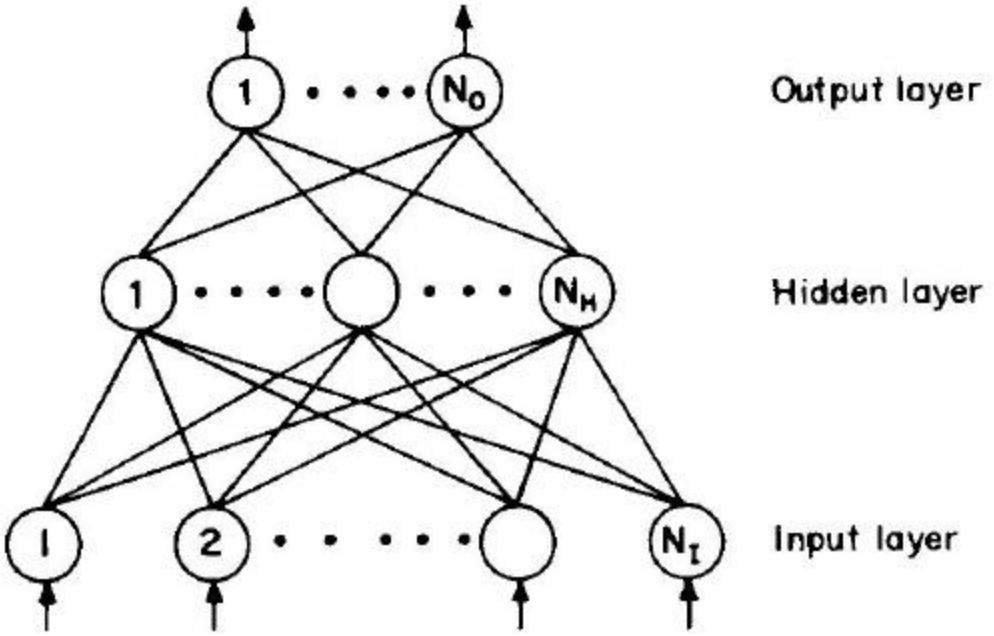


Figure 3-1: General architecture of EBPN consisting of N_I, N_H and N_O neurons in the input, hidden and output layers, respectively. Each neuron in the input and hidden layers is connected to all the neurons in the next layer by means of "weighted" links. In the present study, the input to an EBPN is an appropriately coded DNA sequence and the network output is either a functional property or the class (type) of the input sequence.

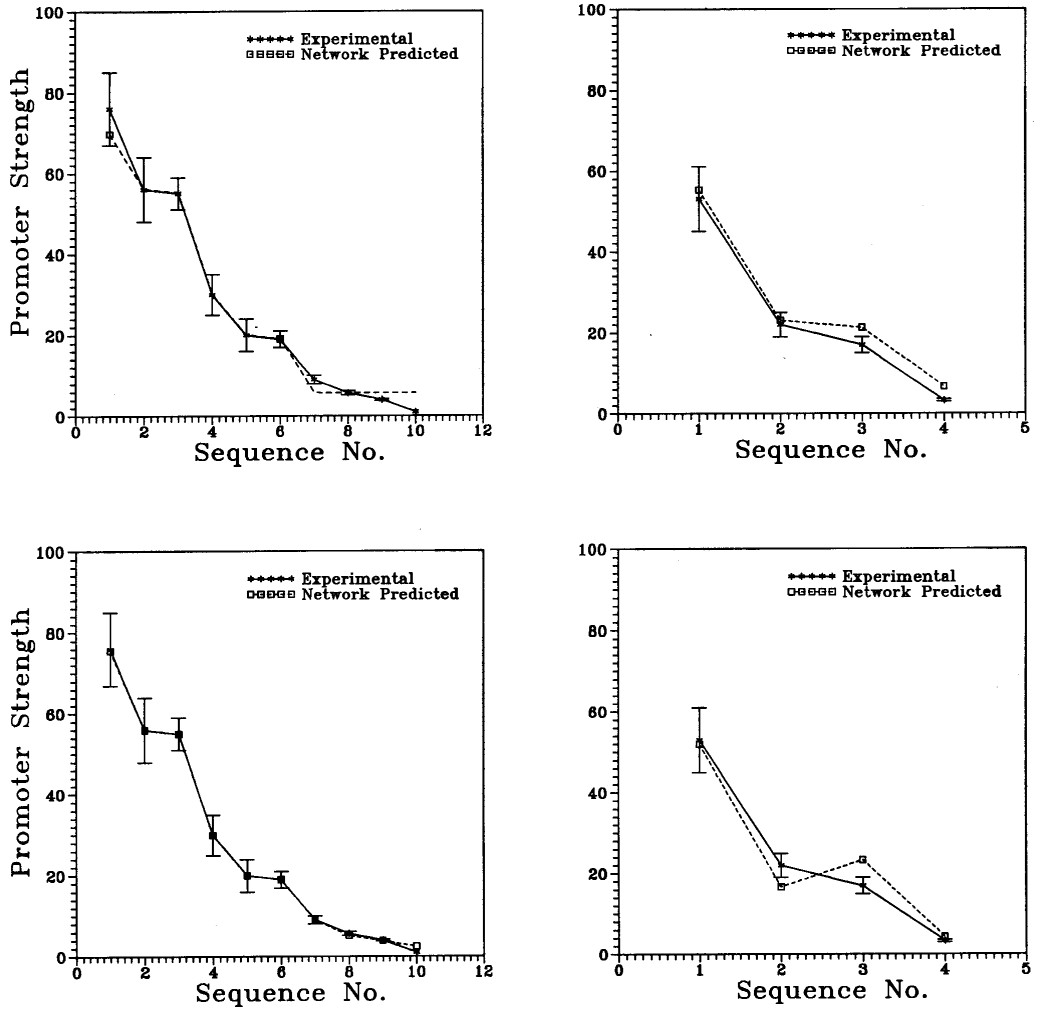


Figure 3-2: Graphical comparison of experimental and network predicted strength (P_{bla} units) values using: (a) wedge code for the training data set, (b) wedge code for the test data set, (c) twist code for the training data set, and (d) twist code for the test data set.

Table I: Wedge and twist code values for different dinucleotides

Dinucleotide	Wedge Code	Twist Code
AA	-7.2	-35.62
AC	1.1	34.40
AG	8.4	27.70
AT	2.6	31.50
CA	-3.5	-34.50
CC	-2.1	-33.67
CG	6.7	29.80
CT	-8.4	-27.70
GA	5.3	36.90
GC	5.0	40.00
GG	2.1	33.67
GT	-1.1	-34.40
TA	0.9	36.00
TC	-5.3	-36.90
TG	3.5	34.50
TT	7.2	35.62

Table II-A: Details of optimal EBPN architectures and RMSE values corresponding to three case studies

Coding Strategy	Case study I: DNA curvature prediction η=0.15 , α=0.10			Case study II: Promoter strength prediction η=0.3 , α=0.15			Case study III: Prokaryotic transcription terminator prediction η=0.5 , α=0.9		
	N_I:N_H:N_O *	RMSE		N_I:N_H:N_O	RMSE		N_I:N_H:N_O	Classification Accuracy¨	
		Training set	Test set		Training set	Test set		Training set	Test set
CODE-4	170:1:1	0.320	0.098	280:1:1	0.244	0.147	204:7:1	99.43	98.12
EIIP	44:1:1	0.055	0.051	70:1:1	0.237	0.146	51:7:1	96.59	95.62
Wedge	23:1:1	0.072	0.064	35:1:1	0.032	0.036	26:1:1	95.17	92.50
Twist	23:1:1	0.074	0.069	35:1:1	0.006	0.050	26:1:1	95.45	90.00
rnd-di#	23:1:1	0.071	0.072	35:1:1	0.016	0.138	26:1:1	76.70	75.00

* N_I: number of input neurons, N_H: number of hidden neurons, N_O: number of output neurons, η: learning rate, α: momentum coefficient

¨ Percentage of correctly classified sequences

[#] rnd di denotes Random dinucleotide coding scheme.

Table II-B: Statistical analysis of different combinations of sample sets comprising experimental and network predicted R_L values

No.	Sample set of R_L values	$\overline{X}$	$\overline{Y}$	s_x^2	s_y^2	r_{xy}	Z_c ¨	F_c ·
1	*x*=expt *y*=code4	1.05	1.06	0.021	0.015	0.877	-0.386	1.40
2	*x*=expt *y*=eiip	1.05	1.05	0.021	0.019	0.954	-0.111	1.09
3	*x*=expt *y*=wedge	1.05	1.06	0.021	0.016	0.931	-0.209	1.30
4	*x*=expt *y*=twist	1.05	1.06	0.021	0.016	0.920	-0.305	1.31
5	*x*=expt *y*=rnd di	1.05	1.05	0.021	0.016	0.890	-0.167	1.28
6	*x*=code4 *y*=eiip	1.06	1.05	0.015	0.019	0.901	0.274	0.78
7	*x*=code4 *y*=wedge	1.06	1.06	0.015	0.016	0.892	0.186	0.93
8	*x*=code4 *y*=twist	1.06	1.06	0.015	0.016	0.890	0.081	0.94
9	*x*=code4 *y*=rnd di	1.06	1.05	0.015	0.016	0.867	0.230	0.91
10	*x*=eiip *y*=wedge	1.05	1.06	0.019	0.016	0.920	-0.095	1.20
11	*x*=eiip *y*=twist	1.05	1.06	0.019	0.016	0.911	-0.193	1.21
12	*x*=eiip *y*=rnd di	1.05	1.05	0.019	0.016	0.906	-0.052	1.18
13	*x*=wedge *y*=twist	1.06	1.06	0.016	0.016	0.989	-0.103	1.00
14	*x*=wedge *y*=rnd di	1.06	1.05	0.016	0.016	0.968	0.044	0.98
15	*x*=twist *y*=rnd di	1.06	1.05	0.016	0.016	0.963	0.147	0.98

¨ $Z_\alpha = 2.33$ at $\alpha = 0.01$, and $Z_\alpha = 1.64$ at $\alpha = 0.05$

· $F_{53,53,0.01} = 1.60$ for $n_x = n_y = 54$ at $\alpha = 0.01$, and $F_{53,53,0.05} = 1.39$ at $\alpha = 0.05$

Table II-C: Comparison of different coding strategies using *leave-k-out* cross-validation method

Coding Strategy	Case Study I $k=9$, $\eta=0.15$, $\alpha=0.10$			Case Study II $k=2$, $\eta=0.3$, $\alpha=0.15$		
	N_I:N_H:N_O	Average RMSE		N_I:N_H:N_O	Average RMSE	
		Training	Test		Training	Test
CODE-4	170:1:1	0.161	0.046	280:1:1	0.148	0.107
EIIP	44:1:1	0.137	0.094	70:1:1	0.112	0.081
Wedge	23:1:1	0.112	0.091	35:1:1	0.013	0.033
Twist	23:1:1	0.102	0.074	35:1:1	0.006	0.045
md di	23:1:1	0.159	0.098	35:1:1	0.124	0.055

Table III: Listing of various promoters transcribed by *E.coli* RNA polymerase and their in vivo promoter strengths expressed in P_{bla} units.

No.	Promoter	Promoter Strength
01	P_{H207}	55 (4)
02	$P_{D/E20}$	56 (8)
03	P_{N25}	30 (5)
04	P_{G25}	19 (2)
05	P_{J5}	9 (1)
06	P_{A1}	76 (9)
07	P_{A2}	20 (4)
08	P_{A3}	22 (3)*
09	P_{L} Å	53(8) *
10	P_{lac}	5.7 (0.5)
11	P_{lacUV5}	3.3(0.3) *
12	P_{tacI}	17 (2) *
13	P_{con}	4 (0.2)
14	P_{bla}	1

* Promoter sequences that were part of the test set.
Å Promoter strength taken from Knaus and Bujard [19].

Table IV: Statistical analysis of different combinations of sample sets comprising experimental and network predicted promoter strength values

No.	Sample set of promoter strength values	$\overline{X}$	$\overline{Y}$	s_x^2	s_y^2	r_{xy}	t_c [a]	F_c [©]
1	*x*=expt *y*=code4	26.50	26.20	534.991	107.783	0.631	0.043	4.96
2	*x* =expt *y*=eiip	26.50	26.21	534.991	62.779	0.753	0.042	8.52
3	*x* =expt *y*=wedge	26.50	27.12	534.991	475.569	0.994	-0.070	1.12
4	*x* =expt *y*=twist	26.50	26.61	534.991	521.070	0.995	-0.012	1.03
5	*x* =expt *y*=rnd di	26.50	29.23	534.991	509.097	0.969	-0.304	1.05
6	x=code4 *y*=eiip	26.20	26.21	107.783	62.779	0.326	-0.004	1.72
7	*x*=code4 *y*=wedge	26.20	27.12	107.783	475.569	0.604	-0.138	0.23
8	*x*=code4 *y*=twist	26.20	26.61	107.783	521.069	0.588	-0.059	0.21
9	*x*=code4 *y*=rnd di	26.20	29.23	107.783	509.098	0.590	-0.440	0.21
10	*x*=eiip *y*=wedge	26.21	27.12	62.779	475.569	0.772	-0.141	0.13
11	*x*=eiip *y*=twist	26.21	26.61	62.779	521.069	0.759	-0.058	0.12
12	*x*=eiip *y*=rnd di	26.21	29.23	62.779	509.098	0.703	-0.454	0.12
13	*x*=wedge *y*=twist	27.12	26.61	475.569	521.069	0.992	0.058	0.91
14	*x*=wedge *y*=rnd di	27.12	29.23	475.569	509.098	0.978	-0.242	0.93
15	*x*=twist *y*=rnd di	26.60	29.23	521.069	509.098	0.972	-0.294	1.02

[a] $t_{2\alpha} = 1.315$ at $\alpha = 0.01$, $t_{2\alpha} = 1.706$ at $\alpha = 0.05$

[©] $F_{13,13,0.01} = 2.42$ for $n_x = n_y = 14$ at $\alpha = 0.01$, $F_{13,13,0.05} = 3.59$ at $\alpha = 0.05$

CHAPTER

4

Optimum DNA curvature using a hybrid approach involving an Artificial Neural Network and Genetic Algorithm

In the present chapter, a hybrid technique involving artificial neural network (ANN) and genetic algorithm (GA) has been proposed for performing modeling and optimization of complex biological systems. In this approach, first an ANN approximates (models) the non-linear relationship(s) existing between its input and output example data sets. Next, the GA, which is a stochastic optimization technique, searches the input space of the ANN with a view to optimize the ANN output. The efficacy of this formalism has been tested by conducting a case study involving optimization of DNA curvature characterized in terms of the R_L value. Using the ANN-GA methodology, a number of sequences possessing high R_L values have been obtained and analyzed to verify the existence of features known to be responsible for the occurrence of curvature. A couple of sequences have also been tested experimentally. The experimental results validate qualitatively and also near-quantitatively, the solutions obtained using the hybrid formalism. The ANN-GA technique is a useful tool to obtain, ahead of experimentation, sequences that yield high R_L values. The methodology is a general one and can be suitably employed for optimizing any other biological feature.

4.1 INTRODUCTION

A situation is often encountered in biological sciences wherein development of a "first principles" (i.e., phenomenological) model becomes impossible owing to the lack of sufficient understanding of the involved biochemical phenomenon. In such situations, Artificial Neural Networks (ANNs) are widely utilized for model development. The main reason behind the extensive use of ANNs being their ability of recognizing and classifying patterns not only from the quantitative data but also from the qualitative data, such as DNA sequences [1]. ANNs trained with the error-back-propagation (EBP) algorithm [2-3] represent the most commonly utilized network paradigm. An EBP-based network (EBPN) is a multi-layered feedforward structure that undergoes supervised learning, i.e., for training it requires an example data set comprising pairs of input and the corresponding output patterns. Once trained adequately, the network can make predictions for the new input data. In essence, ANNs serve as an empirical modeling technique to approximate relationships (especially nonlinear) between two sets of data. For example, an ANN model can be developed to correlate DNA sequences and a sequence-dependent property wherein the sequence and the corresponding property would form the network input and the output, respectively.

In addition to modeling, often an experimenter is interested in knowing the optimal values of the model parameters and / or variables that either maximize or minimize the model output. Such a problem falls in the domain of optimization and suitable optimization schemes need to be devised for optimizing the ANN model. Conventionally, gradient based methods are used for performing function optimization. Their usage presupposes that the objective function to be minimized/maximized is smooth, continuous and differentiable. The validity of these assumptions in the case of ANNs cannot be guaranteed since the model represented by an ANN cannot be conveniently written as a closed-form expression. Therefore, an alternate optimization formalism, which is lenient towards the form of the objective function, must be devised for optimizing ANN models.

In recent years, a class of robust algorithms - known as "Genetic Algorithms" (GAs) - has been used with great success in solving optimization problems involving very large search spaces [4-6]. GAs were originally developed as genetic engineering models mimicking the population evolution in natural systems. Given a functional form, genetic algorithm searches its solution space so as to maximize (or minimize) the prespecified

objective function. In GA procedure, possible solutions to an optimization problem are randomly initialized using binary or real valued strings. The GA begins its search for the optimal solution from this random population of candidate solutions. The candidate solution represented by each string in the population is tested using an objective function, following which all the population strings are ranked. Specifically, when optimization goal involves maximization (minimization) of the objective function, all the population strings are ranked in the decreasing (increasing) order of their objective function scores. Such a ranking, in essence, arranges the candidate solutions in the descending order of their "fitness", which is an indicator of "how well the solution performs at fulfilling the optimization goal". Subsequently, GA operations namely, reproduction, crossover and mutation are performed on the fitter solutions in the population and the operations are repeated until convergence is achieved.

The traditionally employed gradient-based optimization methods are deterministic whereas GAs are stochastic optimization techniques. GAs possess several advantages over the gradient-based methods, the principal one being they do not impose preconditions such as smoothness, continuity and differentiability on the form of the objective function. This GA characteristic assumes special significance in the case of ANN models for which the fulfillment of the above-stated conditions cannot be guaranteed. In essence, GA is one paradigm that can be fruitfully employed for performing optimization of ANN models. The objective of this chapter, therefore, is to present a hybrid strategy involving an EBPN and a GA for the optimization of a biologically important feature or a property. The strategy is a general one and has been exemplified by addressing a specific problem involving optimization of DNA curvature that is expressed in terms of the retardation anomaly value. Retardation anomaly is a measure of electrophoretic anomaly of the curved DNA and reflects the additional friction of the DNA in the gel due to curvature [7]. Relative electrophoretic mobility of most curved DNA fragments monotonously decreases with the fragment length. This is usually characterized as the ratio of the apparent to actual DNA length and the ratio termed as "R_L factor" is found to increase with the increase in the fragment length.

4.2 SYSTEM AND METHODS

4.2.1 Implementation of ANN-GA methodology

Implementation of the ANN-GA methodology is a two-part procedure. In the first, an EBPN is trained to model the input-output example data. An EBPN usually comprises three layers (input, hidden, and output) of processing elements (termed as "nodes"). The nodes in successive layers are connected using weighted connections. During training, the inputs and the outputs of the example data set are used as the network input and the desired output, respectively. Network training involves minimization of an error function [e.g., root-mean-squared-error (RMSE)] using a steepest descent strategy, such as the generalized delta rule (GDR), wherein the network outputs are compared with their desired values and the difference (error) is used to update the inter-layer connection weights. The weights are updated till a convergence criterion is satisfied at which point the network is assumed to be trained. The detailed description of EBPN training can be found at numerous places [8-9].

In the second part of the ANN-GA procedure, a GA is used to optimize the output of the ANN model by rigorously searching the input space of the trained network. This way, the EBPN plays the role of an objective function in the GA implementation wherein the converged weights corresponding to the trained EBPN are used to compute the value (score) of the objective function. The objective function score known also as "fitness score" is essentially the EBPN output when a GA-searched solution string is applied as an input to the trained EBPN. A simple five-step GA for maximizing the objective function can now be summarized as (for details see [4-5, 10]):

Step 1 (Initialization): Create an initial population (size=N) of candidate solution strings (chromosomes) whose elements (binary digits or real numbers) are chosen randomly. Each chromosome in the population is of same length, l. Evaluate each chromosome in the population using the trained EBPN as the objective function and rank the chromosomes as described earlier. Set the initial population as the current population.

Step 2 (Selection): Choose two parent chromosomes from the current population; the selection procedure is carried out using the weighted Roulette-Wheel algorithm [5]. In this strategy, the fittest string on a priority basis chooses its partner at random from among the remaining strings where the probability of selecting a particular

mate is proportional to its fitness. This way only fitter chromosomes are selected as parents for offspring production.

Step 3 (Crossover): Crossover is the most important step of GA. It is responsible for passing significant genetic information to the next generation strings. It is performed as follows: choose randomly a crossover point along the lengths of the parent chromosomes and cut each parent string at that point to generate two substrings. Exchange the substrings between the parent strings to obtain two offspring.

Step 4: Repeat steps 2 and 3 until the total number of offspring generated equals N following which the offspring population is merged with the parent population; the post-merger population has $2N$ chromosomes.

Step 5 (Mutation): Mutate elements of each of the $2N$ strings randomly where the probability of mutation (P_{mut}) is kept small. During mutation, exclude the top ranking string in the parent population so as not to lose it. Next, evaluate each of the $2N$ chromosomes using the objective function and rank them. Discard the lower half of the $2N$-sized population and set the resulting population of size N to the new population (generation).

The above described procedure is repeated till a preselected convergence criterion, such as the GA has evolved a fixed number of generations (N_{gen}), or successive generations have produced similar chromosomes, is satisfied. The best (i.e. first ranked) chromosome in the converged population represents the final result of the genetic algorithmic search. The essence of GA-implementation is that an initial population of randomly generated chromosomes with low objective function scores improves as parents are replaced by better (fitter) offspring. As the steps involved in the GA implementation are stochastic, the final solution depends upon the series of random numbers generated during the search. Thus, to get an overall optimal solution, it may be necessary to repeat the search procedure giving each time a different seed to the random number generator. In the following, results of the case study wherein the proposed hybrid technique has been used to optimize the R_L factor are presented.

4.2.2 Optimization of R_L factor

In a recent study [11], the authors have addressed the problem of modeling DNA curvature wherein based on the experimental data of Bolshoy et al. [12], an EBPN was trained to predict the R_L factor of a given DNA sequence. The data comprised the R_L values of circular, curved, and straight synthetic fragments extrapolated to 90 base-pair length. The trained EBPN architecture has 44 neurons in the input layer for representing the DNA sequence, one neuron in the hidden layer, and one neuron in the output layer to represent the R_L factor. The optimal values of the EBPN training parameters, namely, the learning rate and momentum coefficient were 0.15 and 0.1, respectively. The EBP based model could predict the R_L value of a given sequence with significant accuracy as suggested by the high magnitude (=0.954) for the correlation coefficient between the network-predicted and experimental R_L values. Although the DNA sequences considered for the network training were of variable lengths (i.e., 10, 21, 31, and 42 base-pair long), a single EBPN could predict the R_L factors of all the four sequence-types. The GA-based optimization, however, has been performed separately for the four types with the objective of obtaining sequences possessing high R_L value. The GA procedure for optimizing R_L was implemented as follows.

The flow-chart corresponding to the five GA steps of the ANN-GA methodology is depicted in Figure 4-1. A pseudo-random number generator was used for creating (step 1) an initial random population of 100 (= N) DNA strings with equal composition of A, T, G and C. The encoding of these four nucleotides was performed using their Electron Ion Interaction Potential values (0.1260=A, 0.1335=T, 0.0806=G, 0.1340=C). This nucleotide-encoding scheme, known as the "EIIP code" [13], is the same as used to represent the DNA sequences during the EBPN training [see 11]. The EIIP coding strategy has an advantage over other binary schemes, such as CODE-2 and CODE-4, that it requires just one real number to code a nucleotide. As a result, the input space of the EBPN and, consequently, the chromosome length, l, get significantly reduced. The chromosomes that are shorter than 42 base pairs were uniformly padded with a dummy number (0.01). Each chromosome in the population was 44 elements long (l=44) wherein two more dummy numbers (0.05 and 0.90) were assigned to 43rd and 44th locations to distinguish linear fragments from the circular ones. The steps in the flow chart concerning

the R_L factor evaluation were implemented using the optimal EBPN weights obtained by Parbhane et al. [11].

After selecting the parent pairs as described in step 2, the crossover operation (step 3) was performed on each pair separately as illustrated in Figure 4.2. Performing crossover on $N/2$ pairs of parent strings produced N number of offspring. This offspring population was then added to the parent population to obtain a total of $2N$ strings.

The mutation (step 5) operation simply interchanges the elements of the population strings in a random manner. That is, a string element representing the EIIP value of either A, T, G or C is replaced by the EIIP value of any one of the four nucleotides. Whether a string element undergoes mutation or not was determined using a small value (P_{mut}=0.01) of the mutation probability.

Each EIIP coded DNA sequence in the post-mutation population was evaluated for its R_L value following which the strings were arranged in the decreasing order of their R_L magnitudes. The lower half of the population so arranged was discarded and the resulting population (size=N) was set as the new generation. The procedure barring step 1 was repeated till the convergence criterion that GA has evolved over 100 generations was satisfied. The best-ranked string in the converged population representing the solution of a GA search, possesses highest R_L magnitude as compared to the remaining strings in the population.

4.3 RESULTS AND DISCUSSION

Using different random seeds for initializing the chromosome population (step 1), and following the methodology outlined above, we have obtained several 10, 21, 31 and 42 base-pair long DNA sequences possessing R_L values greater than 1.10. The R_L values exceeding 1.10 can be considered "high" in view of the R_L range [0.54-1.21] represented by the trained EBPN. It may be noted that $R_L > 1.0$ signifies a curved DNA sequence [7]. In Table 1, a sample of DNA sequences possessing high R_L values is provided. For brevity, only five examples of DNA sequences belonging to each of the four types (10, 21, 31 and 42 bp) have been shown, although the ANN-GA methodology is capable of generating a large number of sequences meeting the selection criterion. Examination of

these sequences from the viewpoint of extracting curvature-inducing features is now in order.

From sequence numbers 1-3, it can be noticed that each A_nT_m tract ($n+m \geq 3$) produces a small bend in the DNA helix axis; repetition of these elements in phase with the helix screw results in their coherent addition to form a large overall bend. Thus, these sequences are the examples of the role of A_nT_m tract and influence of phasing (junction model) in determining the extent of curvature [14].

It can be observed from sequence nos. 4 and 5 that non-AA fragments can also induce curvature. The high R_L for these sequences can be explained in terms of the dinucleotide (wedge) model representing the simplest form of the nearest-neighbor interactions [12]. According to this model, the base pair steps other than AA/TT introduce proper wedge angles phased with each other that add coherently.

Sequences 4 and 5 have GGCC as a sequence element repeated in phase with each other. Also, the element appears in the absence of A/T tracts in the sequence context. This feature seems to be responsible for the high R_L values and is in good agreement with the recent X-ray data showing that the GGCC element is intrinsically curved towards the major groove [15].

Sequences 10, 11, 14, 17, 18 and 20 also contain GGCC element, but in the A/T tracts as a sequence context. The corresponding high R_L values can be explained using the trends exhibited by another DNA bending related quantity, namely "$\ln(p)$". It is well known that bovine pancreatic deoxyribonuclease I (DNase I) digestion profiles are used to obtain $\ln(p)$ values, which are realistic DNA bending propensity parameters of trinucleotides. High $\ln(p)$ values for trinucleotides signify that these base sequences owing to the introduction of a positive roll [16] are flexible or inherently bent towards the major groove. By invoking this analogy, the high R_L values for sequences 10, 11, 14, 17, 18 and 20 can be attributed to the additive effects of GCC/GGC trinucleotides and other combinations of trinucleotides possessing high $\ln(p)$ values. Such an explanation also holds for sequences 6-9, 12, 13, 15, 16 and 19 that possess various combinations of trinucleotides with high $\ln(p)$ values viz., TCA/TGA, ATA/TAT, CAG/CTG, ATG/CAT, GCC/GGC, CTA/TAG and GCA/TGC. It is quite clear from above discussion that the features responsible for high curvature (R_L)

and contained in the DNA sequences in Table I, could be explained using other approaches as well; for instance, by analyzing the ln(p) values.

A few of the oligonucleotides (sequence number 11 [31-mer] and 20 [42-mer] from Table I) were synthesized for experimentally validating the results provided by the ANN-GA strategy. The overall framework of the experimental analysis was the same as reported earlier [12, 14] but with minor variations. The oligonucleotides were synthesized chemically (Gibco BRL) with unique two base overhangs to allow head-to-tail polymerization. The oligonucleotides were resolved in 15% denaturing polyacrylamide gels and eluted in TE (10 mM Tris-HCl [pH 8], 1 mM EDTA [pH 8]), purified using a NAP-5 column (Amersham Pharmacia Biotech) dried under vacuum and quantified. 100 pmoles of the oligonucleotide was radioactively labeled with 5 μCi of [γ^{32}-P] ATP (DuPont/NEN, > 6000 Ci/nmol) using 10 U of T4 polynucleotide kinase (PNK, Gibco BRL) at 37^{0}C. After 10 minutes, the reaction was supplemented with 200 pmoles of the complementary strand, 1000 pmoles of cold ATP and 10 U of PNK. After an hour, the reaction mixture was heated to 70^{0}C, held there for 10 minutes and then allowed to cool slowly to room temperature. The reaction mixture was passed through a Sephadex G-50 (Amersham Pharmacia Biotech) spun column and dried to 5 μl. The ligation was set up with T4 DNA ligase (Gibco BRL) at 16^{0}C for 24 hours. The reaction products were subjected to electrophoresis on a 40 cm 8% native polyacrylamide gel (mono: bisacrylamide 29:1) in 90 mM Tris-borate (pH 8.3), 2.5 mM EDTA (pH 8) with an applied voltage of 7 V/cm at room temperature (30^{0}C). The mobility of the ligation products was measured relative to the migration of a 10 base pair *Bam*HI linker ladder (which is known to have normal mobility). The R_L values for 90 base-pair DNA were interpolated from plots of apparent to actual length in base pair units.

The experimentally determined R_L values for sequences 11 and 20 are in the range 1.08 ($\pm$0.03) as against the respective ANN-GA predicted values of 1.23 and 1.22 (refer Table I). It is important to note that the ANN-GA model used the data on electrophoresis measurements carried out at 20-22^{o}C while the above-described experiments were conducted at 30^{o}C. The effect of temperature on the mobility of curved DNA is well documented [7, 14, 17] and it is observed that higher temperatures enhance mobility and, hence, lower the R_L magnitudes. This feature may be responsible for the approximately 9%

difference observed between the ANN-GA predicted and the experimental R_L values of sequences 11 and 20. It can thus be inferred that the experimental results qualitatively and near-quantitatively validate the trends in the optimal sequences provided by the ANN-GA methodology.

The debate on the generality of the first-principles models for predicting the curvature continues in the literature [18]. Extension of the dinucleotide model to tri- and tetranucleotide levels is clearly desirable since such models would then include more sequence context information. It however requires rigorous experimentation on 32 independent trinucleotides and 136 tetranucleotides. This clearly is a difficult proposition although some efforts have already been made [16]. The results presented here suggest that the ANN-GA methodology possess the potential of providing DNA sequences having desired R_L values ahead of experimentation. Thus, trial and error approach may be avoided while performing experiments. Another positive feature of the ANN-GA strategy is that the entire 'modeling-optimization' exercise can be performed using representative experimental data. Since the observed data contain information about the underlying biochemical phenomena, no explicit knowledge of these details are necessary (unlike in phenomenological modeling).

It is important to realize that the hybrid formalism is not intended to replace the laboratory work, but should be used as a guide in designing experiments. The proposed strategy is sufficiently general and, therefore, can be exploited for optimizing other biologically important features or properties.

4.4 REFERENCES

1. Nair, T.M. (1996) In Tambe, S. S., Kulkarni, B. D. and Deshpande, P. B. (Eds), *Elements of Artificial Neural Networks with Selected Applications in Chemical Engineering and Biological Sciences*. Simulation and Advanced Controls, Inc., Louisville, 395-437.
2. Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986) *Nature*, **323**, 533-536.
3. Rumelhart, D. E. and McClelland, J. L. (1986) *Parallel and Distributed Processing: Explorations in the Microstructure of Cognition*, MIT Press, Cambridge, MA.
4. Davis, L. (1991) ed. *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York.
5. Goldberg, D. E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, Mass.
6. Holland, J.H. (1992) *Adaptation in Natural and Artificial Systems*, (2nd ed.), University of Michigan Press, Ann Arbor.
7. Marini, J.C., Levene, S.D., Crothers, D.M. and Englund, P.T. (1982) *Proc. Natl. Acad. Sci., USA*, **79**, 7664-7668.
8. Freeman, J. A. and Skapura, D. M. (1992) *Neural Networks Algorithms, Applications, and Programming Techniques*. Addison-Wesley.
9. Tambe, S. S., Kulkarni, B. D. and Deshpande, P. B. (1996) eds., *Elements of Artificial Neural Networks with Selected Applications in Chemical Engineering and Biological Sciences*. Simulation and Advanced Controls, Inc., Louisville.
10. Deb, K. (1995) *Optimization of Engineering Desig n: Algorithms and Examples*, Prentice-Hall, India.
11. Parbhane, R. V., Tambe, S. S. and Kulkarni, B. D. (1998) *Bioinformatics*, **14**, 131-138.
12. Bolshoy, A., McNamara, P., Harrington, R. E. and Trifonov, E. N. (1991) *Proc. Natl. Acad. Sci. USA*, **88**, 2312-2316.
13. Nair, T. M., Tambe, S. S. and Kulkarni, B. D. (1994) *FEBS Lett.*, **346**, 273-277.
14. Koo, H.-S., Wu, H.-M., and Crothers, D. M. (1986) *Nature*, **320**, 501-506.

15. Brukner, I., Susic, S., Dlakic, M., Savic, A. and Pongor, S. (1994) *J. Mol. Biol.*, **236**, 26-32.
16. Brukner, I., Sanchez, R., Suck, D. and Pongor, S. (1995) *EMBO J.*, **14**, 1812-1818.
17. Marini J.C., Efforn, P.N., Goodman, T.C., Singleton, C.K., Wells, R.D., Wartell, R.M. and Englund, P.T. (1986) *J. of Biol. Chem.*, **259**, 8974-8979.
18. Dlakic, M. and Harrington, R. E. (1996) *Proc. Natl. Acad. Sci., USA*, **93**, 3847-3852.

Table 1: Optimized retardation anomaly (R_L) values along with their DNA sequences obtained using ANN-GA methodology

No.	Sequence Unit	R_L
1	GGGTATTGCG	1.13
2	GGGTTAAGTG	1.13
3	GGTTACGGAG	1.13
4	GGCCCGTGGG	1.12
5	GGCCGTCGGG	1.12
6	GGGCTCTGCGTTGGTGTGCAA	1.23
7	GGCATGAGCGCGGGTCTACTT	1.23
8	GGAACCTGACTAGGCGTGTTA	1.22
9	TTATGCAGATTGGGGGATCTT	1.22
10	GGCCCATGTGCGGTAGTTTCC	1.22
11	CGGAATTGCTTGGGCATATTCGAGCGGGGCC	1.23
12	GGAACCAGATCGGGGCCTATAGCGAGGGTAG	1.23
13	CGTTGTTGCAATGGCTGCACTGAGAGGAGCG	1.23
14	GGGCGTAACACCGGCCACTATGATTGGCATC	1.23
15	GGGCATATTATCGGCTGACATGTGCAGCGTT	1.22
16	GGCAGTTGTCACAGTTCTCCCTGGAGGTCACTGTCAGGCGCG	1.23
17	AGACAGTCAAACGGAGATCGTGGCAGGCCTTCGATAGGTGTC	1.23
18	GGTCCGTGATATTGTGCGACAGAGTAGGCCGTACCGCGCGAG	1.23
19	GCAATGTGGACAGGGGTGCTCATGAGGCAACGCTAATATGAT	1.23
20	AGGCCCATCCACAGTGACCTCGAGATGCCTTGAACGGCCGGG	1.22

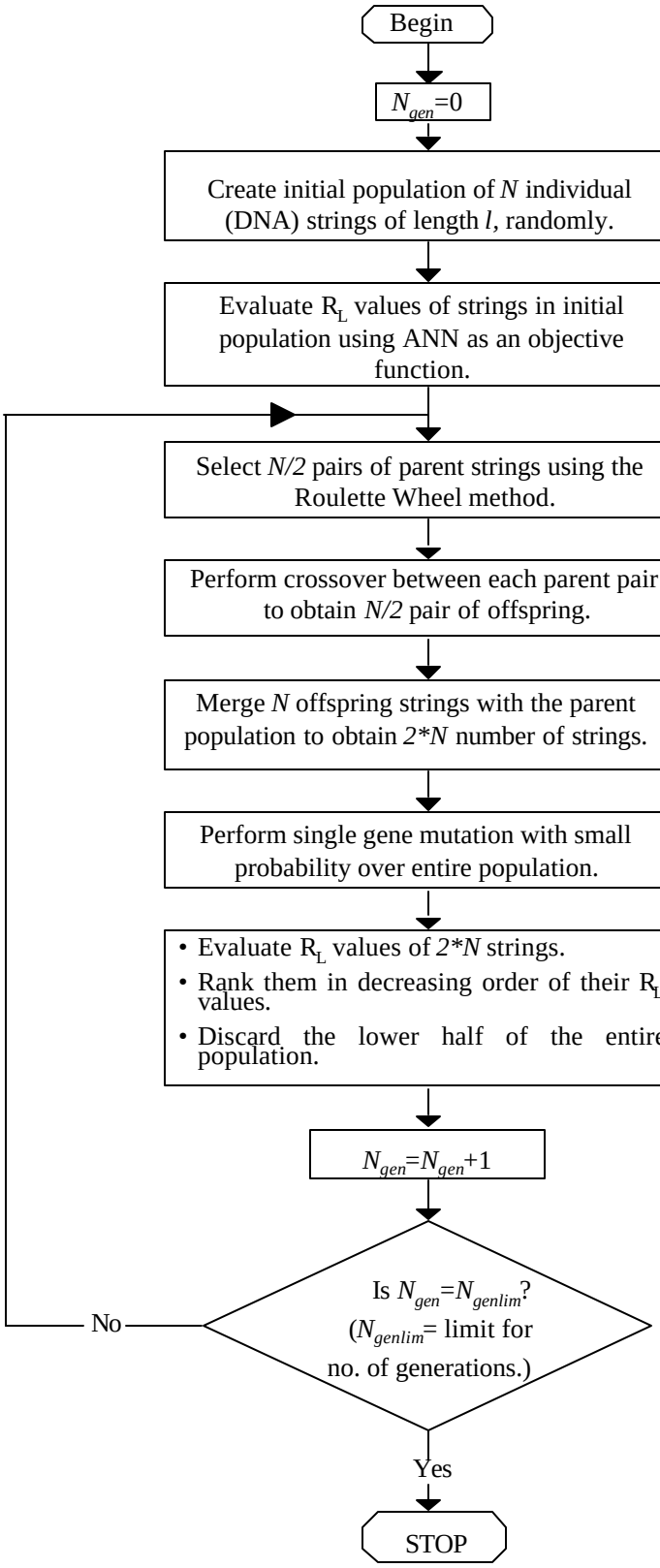


Figure 4-1: Flow chart for the implementation of ANN-GA strategy for the optimization of retardation anomaly values of DNA sequences.

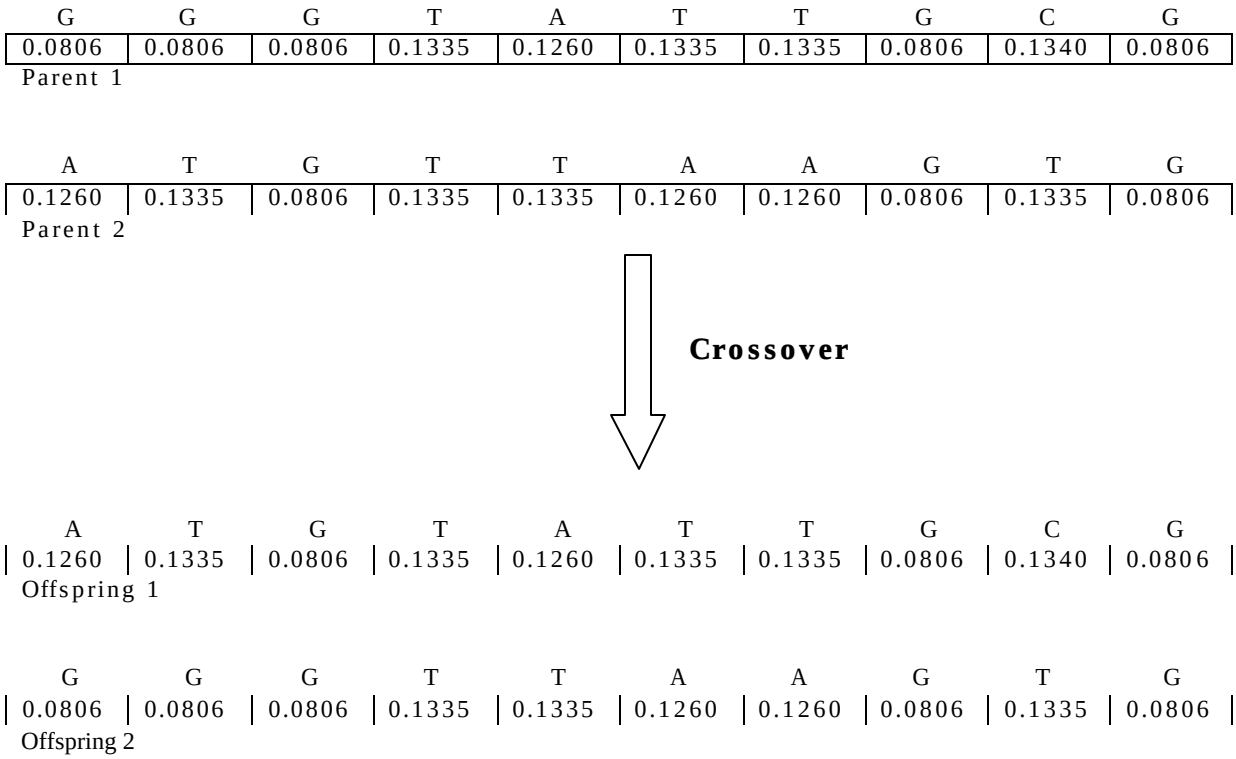


Figure 4-2: Basic crossover of EIIP coded DNA strings (for simplicity crossover between 10-mers is shown).

CHAPTER

5

Optimum transcription efficiency in eukaryotic systems using a hybrid approach involving an Artificial Neural Network and Genetic Algorithm: a case study of β-globin gene

Effects of single base substitutions in the upstream region of the β-globin gene are known to alter the relative transcription level (RTL). Information with regard to multiple base substitutions leading to higher RTL is however very scanty. The motivation of this work is to obtain maximum gene expression using multiple base substitutions. Using an Artificial Neural Network (ANN) and Genetic Algorithm (GA) based hybrid strategy we study the effects of multiple base mutations with particular emphasis on those that can cause enhanced RTL. The study reveals that multiple base substitutions in the conserved as well as non-conserved regions can cause substantial enhancements in RTL. We identify positions in the nucleotide sequences, which preferably should not be altered, as well as those positions where mutations can lead to increased RTL. The various trends observed are rationalized. The ANN-GA strategy can help in experimental planning and reducing the search space.

5.1 Introduction

The mechanism of the level of gene expression governing the fate of a cell, cell proliferation, and survival of the organism continues to be one of the intriguing questions to molecular biologists. Even more interesting is the mechanism underlying the switching on and off of a particular gene according to development programs. Failure to follow these programs accurately may result in gross abnormalities in the gene structure. Most control mechanisms in the regulation of gene expression occur at the level of transcription and translation. The efficiencies of these critical processes are determined by the nucleotide sequences of the promoter and the ribosome binding sites (RBS) on the encoded mRNA. Although the nucleotide sequences of many promoters and the RBS are known, the specific features determining the efficiency of transcription and translation are not well understood. The very first step of gene expression i.e. transcription is an intricate, highly regulated process and its role in eukaryotes is still not clear. The biochemical events in transcription involve a series of highly specific interactions between regulatory sequences in DNA and the cellular enzyme RNA polymerase that catalyzes the transcription reaction.

The eukaryotic promoters that have been most thoroughly studied by the molecular genetic approach are: (i) the herpesvirus thymidine kinase (tk) [1-3], (ii) the SV40 T-antigen [4], and (iii) mammalian β-globin genes [5]. These studies have focused on the DNA sequences immediately upstream from the messenger RNA (mRNA) initiation sites and provided an evidence for the establishment of transcription efficiency via signals contained within the eukaryotic genes. However, the problem of prediction of the mutations in the upstream region that may lead to maximum expression of a gene has so far remained unresolved. The problem essentially is that of an optimization where the nucleotide content of a promoter sequence needs to be rigorously searched such that the corresponding transcription efficiency represented in terms of relative transcription level (RTL) is maximized. The general objective in optimization is to obtain a set of values of the variables and/or parameters subject to various constraints (if applicable) that will produce the desired optimum response for the chosen objective function [6]. For performing such an optimization, the conventional methods such as gradient-based algorithms require: (i) a mathematical model described by a smooth, continuous closed functional form,

and (ii) derivatives of the function to be optimized. Biological systems often being non-linear and complex, are difficult to be modeled phenomenonlogically, or even empirically. Consequently, such systems are not amenable to representation in an exact mathematical form and, therefore, to optimization using gradient-based methods. In view of these difficulties, it becomes necessary to explore newer tools for solving problems such as the optimization of transcription efficiency alluded to above. The objective of this chapter is two-fold: (i) to present a hybrid non-linear strategy involving an artificial neural network (ANN) and genetic algorithm (GA) for the optimization of transcription efficiency, and (ii) to obtain an insight - from the results of the ANN-GA based optimization simulations - about the structural aspects of β-globin gene leading to high transcription efficiency.

5.1.1 Philosophy of ANN-GA optimization technique

In the last decade, ANNs have been extensively used for modeling biological systems; the main reason being their ability of modeling not only quantitative data but also qualitative data, such as DNA sequences [7]. ANNs trained with the error-back-propagation (EBP) algorithm [8-9] represent the most widely used neural network paradigm. An EBP-based network (EBPN) possesses a multi-layered feed-forward structure that undergoes supervised learning, i.e. for training it requires an example data set comprising pairs of input and the corresponding output patterns. Once trained adequately, an EBPN is capable of making output predictions for new input data. In essence, an EBPN serves as a non-phenomenological modeling technique for approximating (particularly nonlinear) relationships existing between two sets of data. For instance, an ANN model has been developed to correlate a DNA sequence and the sequence-dependent property, namely, transcription efficiency [10]. ANNs though a powerful modeling technique possess an undesirable characteristic that they essentially lead to "black-box" models. It means that an ANN model cannot be easily expressed as a closed form equation relating its inputs and outputs. Consequently, utilization of the gradient descent-based optimization methodologies becomes cumbersome. A novel technique known as "genetic algorithms (GAs)" that helps in overcoming the said difficulty is described below.

5.1.2 Genetic Algorithms

GAs are nonlinear optimization techniques based on the mechanisms of natural selection and genetics [11-13]. They combine the "survival of the fittest" principle of natural selection with a randomized information exchange procedure known as *crossover* to arrive at a robust search and optimization technique. A prerequisite to optimization using the GA methodology is a functional form (model) whose parameters/variables are to be optimized. Given such a functional form, a GA searches its solution (parameter) space so as to maximize a pre-specified objective criterion (function). In GA parlance, the objective function is referred to as *fitness* function. The salient features of GAs are [14-15]:

- GAs perform global search as against the local one performed by the gradient-based methods. Thus, GAs are most likely to arrive at the global optimum of the objective function.
- During optimization, search is conducted from a population of probable candidate solutions to the problem under study.
- GA search procedure is stochastic requiring only values of the function to be optimized and it does not impose preconditions such as smoothness, derivability, and continuity, on the form of the function.
- GAs can easily handle functions that are highly non-linear, complex, and noisy; in such cases the traditional gradient-based methods are found to be inefficient.

It may be noted that owing to GA's leniency towards the form of the function to be optimized, it is possible to use an ANN model in place of a closed form function. In the resulting ANN-GA optimization approach, a trained ANN serves as an input-output model whose inputs are optimized using the GA methodology. The GA in essence finds the optimal values of the network inputs such that the corresponding values of the network outputs are maximized.

5.1.3 ANN-GA based optimization of eukaryotic transcription efficiency

In order to address the optimization problem of maximizing the eukaryotic transcription efficiency, we have chosen the globin gene as a test case. The mouse globin gene family is an ideal candidate for the study of gene expression since differentiation of these genes exhibits both the temporal and coordinate regulation.

Thus, the globin gene has been extensively studied for its expression, function, and abnormalities. It has been observed that the mutations in the β-globin gene and its upstream regions can cause many genetic disorders [16].

5.2 SYSTEM AND METHODS

5.2.1 Implementation of ANN-GA methodology

Implementation of the ANN-GA methodology is a two-part procedure; the first part consists of training an EBPN with a view to model the input-output example data. An EBPN architecture in general possesses three layers (input, hidden, and output) of neurons (also termed as "nodes"). The nodes in the successive layers are connected using weighted links. The two sets of example data to be modeled (correlated) by training an EBPN form the network input and the desired output, respectively. In the present study, DNA sequences of the β-globin gene and the corresponding transcription efficiency values form the EBPN input and output, respectively. Training of EBPN involves minimization of an error function such as the *sum-squared-error* (SSE) using a strategy known as the *generalized delta rule* (GDR). While minimization, the network outputs are compared with their desired values and the corresponding SSE is used to update the values of the inter-layer connection weights. The weight-updation continues till a convergence criterion is satisfied. At this point the network is assumed to be trained. The detailed description of EBPN training can be found at numerous places (see e.g., [17-18]).

In the second part of the ANN-GA hybrid methodology, a GA rigorously searches the input space of the trained EBPN so as to maximize its output. In essence, the GA searches the sequence space with a view to maximize the magnitude of the transcription efficiency. GA begins by randomly encoding a set (population) of possible solutions to the optimization problem in the form of "chromosome strings". A pre-specified objective function returns the fitness value (score) of each chromosome string in a population that serves as a measure of the goodness of the solution searched by the GA. In the ANN-GA methodology, the trained EBPN acts as an objective function wherein the network output also represents the fitness score of the GA-searched solution string (a DNA sequence). For computing the fitness value, the DNA solution string is applied as an input to the trained EBPN and the network output is evaluated. Since a nonlinear activation function such as the logistic

sigmoid is used to compute the output of EBPN's output nodes, the fitness value is always constrained between zero and one. With this background, a simple five-step GA has been described in the following:

Step 1 (Initialization): Create a random initial population of N chromosome strings where each string contains l elements. A string element characterizing a nucleotide is chosen randomly with equal probability of selecting either *A*, *T*, *G*, or *C*. Evaluate each chromosome in the initial population using ANN as the objective function. Set the initial population as the current population.

Step 2 (Selection): Select chromosome strings from the current population with a view to form a mating pool to be used subsequently for the offspring production. The selection procedure is stochastic in nature and carried out using the weighted Roulette-wheel algorithm wherein fitter chromosome strings on a priority basis select their partner from among the remaining strings. The probability of selecting of a particular partner string is directly proportional to its fitness score. Such a selection procedure gives rise to a mating pool comprising $N/2$ number of parent pairs.

Step 3 (Crossover): The action of this most important GA operator results in creating two offspring chromosomes from each parent-pair. Typically, the two parent chromosomes are cut at the same randomly selected crossover point to obtain two sub-strings per parent string. The second sub-strings are then mutually exchanged between the parent chromosomes and combined with the respective first sub-strings to generate two offspring chromosomes (see Figure 5-1). The probability of crossover (P_{cross}) is kept high. The crossover operator essentially generates new solution strings (DNA sequences) thereby searching hitherto unexplored regions in the solution space. Repeating crossover operation on $N/2$ parent pairs generates N number of offspring strings following which the offspring population is merged with the parent population; the post-merger population has $2N$ strings.

Step 4 (Mutation): Randomly change (mutate) elements of the offspring strings where the probability (P_{mut}) an element undergoing mutation is kept small. The objective of mutation is to create new solutions in the neighborhood of the region represented by the $2N$ number of chromosome strings and thereby perform a local search around the region. Subsequently, evaluate fitness of each chromosome using EBPN as the objective function and rank the $2N$

number of strings in the descending order of their fitness scores. Next, discard the lower half of the *2N*-sized population and set the resulting population of size *N* to the new population (generation).

The above-described procedure is repeated till a pre-selected convergence criterion such as, the GA has evolved a fixed number of generations or the fitness of the best solution does not improve in successive generations, gets satisfied. The best chromosome as judged by the highest fitness score following convergence, represents the final solution of the genetic search. The essence of GA-implementation can be stated as: better solutions in the current population are selected for the reproduction and their offspring generated via crossover and mutation operations replace the sub-optimal solutions. The population of candidate solutions, owing to the repetitive actions of the crossover and mutation operators, improves itself from one generation to the next till convergence is achieved.

As most steps involved in the GA implementation are performed stochastically, the final solution depends upon the series of random numbers used during the search. Thus, it may be necessary - for securing an overall optimal solution - to repeat the search procedure giving each time a different seed to the random number generator. This way GA begins with different initial populations, which help in the exploration of widely different solution space.

5.2.2 Optimization of transcription efficiency

In an earlier study [10], the problem of modeling transcription efficiency was addressed using EBPN as the modeling tool. The data for modeling was taken from the mutation studies carried out by Myers et al. [19-20] wherein saturation mutagenesis has been used to introduce random single base substitutions into the mouse β-globin promoter region. The effects of single base substitutions in the β-globin promoter have been determined by comparing the levels of correctly initiated RNA derived from the test and reference plasmids co-transfected into HeLa cells and expressed as the relative transcription level (RTL) of each mutant. The expression used for computing the RTL value has been:

$$RTL = \frac{M / R_1}{WT / R_2} \qquad (1)$$

where M refers to signal of the mutant test gene; WT is the signal from the wild-type

test gene; R_1 represents the signal from the reference gene co-transfected with the mutant test gene, and R_2 denotes the signal from the reference gene co-transfected with the wild-type test gene.

The data used by Nair et al. [10] consisted of the β-globin promoter and its mutant sequences (network input) and their corresponding RTL values (network output). In the present work we used the available data on single base substitution in the upstream region of β-globin and its effects on the RTL value. It is important to note that the data on effects of multiple base substitutions is practically nonexistent. It is expected, however, that a properly trained neural network would capture the intrinsic patterns. For EBPN training, the sequences with mutations were coded using the CODE-4 strategy [21], wherein *A*, *T*, *G* and *C* were represented by four binary digits: 0001 = C, 0010 = G, 0100 = A, and 1000 = T. The desired (target) output of each sequence was the experimentally determined RTL values normalized by dividing with ten so that they lie between zero and one. The EBPN architecture had 484 neurons in the input layer for representing the DNA sequences each of length 121 bp, eight neurons in a single hidden layer, and one neuron in the output layer to represent the RTL value (refer Figure 5-2). The values of the GDR parameters, namely, the learning rate and momentum coefficient that resulted in the optimal values of the EBPN weights were 0.6 and 0.9, respectively.

The flow-chart of the ANN-GA hybrid methodology as applied to the RTL optimization problem is depicted in Figure 5-3. The steps in flow-chart concerning the objective function (RTL) evaluation were executed using the optimal EBPN weights obtained by Nair and co-workers [10]. This essentially involves operating the trained EBPN in the prediction mode and multiplying the output by ten. The specific steps in the flow-chart relating to GA were implemented as given below.

Instead of creating the initial population (step 1) of candidate solutions representing the DNA sequences randomly, we used the promoter sequence of the mouse β-globin gene and its mutants as the initial population for the GA analysis. Specifically, 130 patterns of DNA promoter sequences and their mutants whose experimental RTL values are known, were used as the strings in the initial population. This was done purposely so that the GA search begins directly from the most plausible solution space. The values of the GA parameters used for simulation are: population size (N) = 130, probability of crossover (P_{cross}) = 1.0, probability of mutation (P_{mut}) = 0.01,

total number of generations over which the GA evolves (N_{gen}) = 100, and the length of each chromosome string (l) = 121.

5.3 RESULTS AND DISCUSSION

In this study, we have specifically analyzed the transcriptional control signals of a eukaryotic protein-coding gene for establishing a relationship between the site of mutation and increased level of the process of eukaryotic gene transcription. Experimentally, Myers and co-workers [20] could obtain only one single base substitution pattern of upstream region of β-globin gene whose transcription efficiency was 3.5. However, using the ANN-GA methodology, it was possible using multiple base substitution to obtain a large number of sequences having transcription efficiency greater than 3.5. This was achieved by repeating the ANN-GA procedure several times while utilizing every time a different seed value for initializing the random number generator. In the ensuing paragraphs we discuss the significance of the results obtained using the ANN-GA optimization approach. For brevity, the discussion is limited to only ten sequences possessing RTL magnitudes in excess of 3.5. These sequences and their corresponding RTL values are listed in Table I.

Myers and co-workers [20] have shown that single base substitutions in three conserved regions of the promoter resulted in a significant decrease in the level of transcription in: (i) CACCC box, (ii) CCAAT box, and (iii) the TATA box. It was also shown that a promoter containing two base substitutions, one at -75 and the other at -74 results in a 40 to 50-fold decrease in the RTL. In contrast, two different mutations in nucleotides immediately upstream from the CCAAT box caused a 3- to 3.5- fold increase in transcription. Thus, positions -78 and -79 were termed "up mutations". With these two minor exceptions, single base substitutions in all other regions of the promoter were shown to have no effect on transcription. The ANN-GA approach, on the other hand, could arrive at multiple base substitutions that synergistically shows a significant increase in the transcription efficiency.

A comparison of sequences in the upstream region of β-globin gene (glo, RTL=1.00) with the ANN-GA predicted sequences from the same region (R1 to R10, RTL > 3.5) has been made using FASTA package [22]. Such a comparison helps to understand the role of nucleotide variation leading to high transcription efficiency of ANN-GA simulated patterns vis-a-vis original sequence of upstream region of β-

globin gene. The results of comparison, shown in Table II, indicate that sequences from the upstream region of β-globin gene possessing maximum transcription efficiency show 74.4-95.8% sequence homology with the upstream region having transcription efficiency value of one. The nucleotide positions in the sequences predicted by the ANN-GA method that are not similar to the upstream region of β-globin gene can be considered as effective mutation points (listed in Table III) for sequences indexed as R1 to R10. These points are most probably responsible for enhancing the transcription efficiency of β-globin gene.

The ANN-GA simulation results show that not all mutations in three conserved regions decrease the RTL as is generally believed based upon the available experimental results [20]. In order to interpret the results and better understand the role of mutations in enhancing the transcription efficiency, a close look at the sequences R1 to R10 reveal the following: (i) mutations in conserved regions can enhance RTL (sequences R1, R3, R4, R7, R8, and R9), and (ii) mutations in non-conserved regions can also enhance RTL (sequences R2, R5, R6 and R10). In what follows we shall analyze these cases separately. Also, to understand the role of individual positions of mutations and their surroundings we further subdivide the sequence into seven different segments consisting of : (i) upstream region of CACCC box (i.e., -101 to -96 position), (ii) CACCC box (located between -95 to -87 position), (iii) region between CACCC box and CCAAT box (i.e., -86 to -78 position), (iv) CCAAT box (present between -77 to -72 position), (v) region between CCAAT box and TATA box (-71 to -31 position), (vi) TATA box (lying between -30 to -26 position), and (vii) region between -25 to cap site and the region below cap site.

I. Mutations in conserved regions leading to higher RTL

CACCC box (located between -95 to -87 position):

- The optimal sequences having value of RTL in excess of 3.5 searched by the genetic algorithm, including the representative examples of sequences shown here (R1 to R10), reveal that the positions -87, -90, -91, -92 and -93 remain unaltered. This feature is therefore relevant for obtaining sequences with higher RTL.
- Mutations at positions other than those listed above can cause enhancement in RTL. We show one example of each such alteration. Thus mutation at position -

88 (sequence R9), -89 (sequence R8), along with the changes at few other positions (see sequences R8 and R9 for details) cause several fold increase in RTL. It is important to note that these sequences also include the mutations at the 'up-mutation points'. Sequences R4 and R7 show case examples when mutation occurs at the other remaining positions viz. -94 and -95 and cause enhancement. These examples also show that mutation at these positions is also accompanied by change at few other locations, but this time the mutations at the 'up-mutation points' is not involved.

CCAAT box (present between -77 and -72 positions):

- Sequences R1 to R10, show that the nucleotide positions -73, -75, -76 and -77, remain unchanged. No alteration in these positions seem to be important for high transcription efficiency. Other positions viz. -72 and -74 within this region can undergo mutations to cause increased RTL. We show one example of each.
- Sequence R3 indicates that if mutation at -74 position is accompanied by mutation at the "up mutation points" (positions -78 and -79), then an increase in RTL value is witnessed. Note that -74 position is responsible for lowering the RTL magnitude, whereas -78 and -79 position causes increase. The simultaneous mutations has an synergistic effect-causing enhancement more than known for the up mutation point.
- Upon examining sequence R8 it can be noted that if nucleotide position -72 is mutated in combination with "up mutation point" (position -78), and other favorable mutation points (especially in the region -71 to -31 and -25 to cap site), then it causes high magnitude of RTL.

TATA box (lying between -30 and -26 positions):

- For sequences R1 and R8, mutations at -27 and -30 positions effect increase in RTL value if they possess mutation at -78 position and, additionally, at other favorable mutation points such as -47 and -66 positions. These results once again underline the importance of up mutation point, such as position -78.
- At -26 and -29 positions of sequence R4, transition (A → G i.e. R → R) mutations are witnessed. In here, despite presence of mutations in the TATA box, high RTL value has been obtained. This can be interpreted as: if specific mutations (positions -26 and -29) in the TATA box are supported by drastic variation in the nucleotide content of the region surrounding TATA box (i.e., region

between -71 and -31, and -25 and cap site), then they result in increased RTL.

- The % identity (homology) of sequence R4 with original β-globin gene promoter is 74.4. This value despite being the lowest among the ten ANN-GA predicted patterns (refer Table II), the corresponding RTL value (=4.8404) is high.

II. Mutations in non-conserved regions leading to higher RTL

Upstream region of CACCC box (positions -101 to -96):

- If mutations in this region are in favorable agreement with other mutation points, especially in the region -71 to -31, they cause increase in the magnitude of RTL. This is evidenced from the sequence entries R2, R4 and, R7-R10 listed in Table III. The sequences also indicate that *G* at -97, -84 and -78 positions is always mutated by *A*, *T* and *C* respectively.
- For the ten patterns in Table III, positions -99 and -100 are always conserved thus indicating their importance in maintaining high transcription efficiency.

Region between CACCC box and CCAAT box (positions -86 to -78):

- The region is of prime importance since it includes the most important positions i.e., -78 and -79. These two "up mutation points" are primarily responsible for increased transcription efficiency (see sequences R1, R3, R6, R8 and R9).
- Sequences R1-R10 do not exhibit any effective mutation at -77 position. Moreover, as verified experimentally [20], the mutation at -77 position, which is in the nearest-neighbor position of up mutation points (i.e., -78 and -79 position), does not seem to help in increasing transcription efficiency.
- At position -78 of sequences R1 and R3, and at position -84 of sequences R5 and R9, transversion type of mutation (-84 and -78 G → C or T i.e., R → Y) can be observed. It can therefore be inferred that the transversion mutation at these positions can cause increased magnitude of RTL.

Region between CCAAT box and TATA box (positions -71 to -31):

- Table III lists various combinations of multiple base substitutions for sequences R1-R10 in the region between CCAAT box and TATA box, which result in the increased RTL value. However, the average trend in the ten sequences suggests that nucleotide positions -71, -70, -68, -67, -65, -55, -48 and -43, despite remaining unchanged, still cause high RTL. Thus these positions seem to be important in obtaining high RTL.

- Transversion type of mutations (-60 G□T, -59 and -57 A □ T or C i.e. R □ Y) seen at position -60 (sequences R4, R5 and R6), at position -59 (sequences R2, R4 and R8), and at position -57 (sequences R4, R7 and R8) appear to cause high transcription efficiency.

Region between -25 to cap site and in the region below the cap site:

- In most of the cases, the mutations in these regions have favorably supported the multiple base substitutions in the upstream region of gene. It is also of interest to study the role of this region, in causing increased transcription efficiency for sequences where % identity between the original β-globin promoter sequence and the ANN-GA simulated promoter patterns is greater than 90% (refer Table II). Although R6, R9, and R10 meet the stated criterion, we will concentrate only on sequence R10 since sequences R6 and R9 show presence of up mutation points. The % identity of sequence R10 with β-globin promoter is 94.2 and its RTL is 3.6896. Interesting feature of this sequence is that all the three conserved regions i.e., CACCC, CCAAT and TATA box, are not subjected to any mutational changes; the sequence shows variation only in regions -101 to -96, -71 to -31, and below the cap site (position +14). Since R10 possesses maximum homology with the original β-globin gene, only eight effective mutation points that can lead to higher RTL are possible. Thus mutations at positions -101, -98, -97, -56, -51, -46, -41 and +14 can cause increased RTL.
- Among the ten sequences, R8 possesses highest RTL magnitude (=6.7307). This pattern includes mutation at position -78 (up mutation point) and has % identity value of 79.3. Hence, sequence R10 gives us an idea about the effective multiple mutation points, in regions -71 to -31, -25 to the cap site, and below the cap site, that eventually lead to the highest RTL value. This is an example of how the ANN-GA optimization methodology could be exploited for a priori estimation of multiple base substitutions before conducting the mutation experiments.

5.3.1 Role of curvature in gene expression

Sequence dependent DNA structure is important in packaging, recombination and transcription. Therefore it is of interest to study the role of sequence-dependent DNA structure in governing the extent of transcription efficiency. For this purpose, CURVATURE program [23] can be used. This program is useful for plotting the

sequence-dependent spatial trajectory of the DNA double helix and/or distribution of curvature along the DNA molecule. The routine calculates the overall DNA path using experimentally determined local helix parameters, namely, helix twist angle, wedge (deflection) angle, and direction (of deflection) angle [24]. The CURVATURE software can thus be used to investigate possible role of curvature in modulation of gene expression and to locate curved portions of DNA that may play an important role in sequence specific DNA-protein interactions.

For conducting the above-mentioned investigation, the DNA sequence of upstream region of β-globin gene (glo, RTL=1.00) and ANN-GA predicted patterns of β-globin gene were used as inputs to the CURVATURE program and the likely degree of curvature at each point along the molecule was computed. The graphical comparison of the curvature map of promoter sequence of β-globin gene and the ANN-GA predicted promoter sequences is depicted in Figure 5-4. The results suggest that sequences having maximum transcription efficiency show the sequence-dependant bendability or deformability of duplex DNA. This can be justified on the fact that certain nucleic acid sequences take up a particular structure required for binding to a protein at lower free energy than other sequences. The comparison also reveals that a change in the superstructure results in the alteration of transcriptional activity. These results in essence indicate that the ANN-GA methodology is able to capture the relationship between DNA superstructures and transcriptional activity.

Figure 5-5 shows the comparison of spatial trajectories of the DNA double helix of upstream region of β-globin gene (glo, RTL=1.0) and the promoter sequence (R8) having highest RTL (=6.7307). In both the cases, the projections are chosen such that the most curved regions of the fragments are seen best. This is done by placing the plane - where the axis is curved - perpendicular to the viewing direction. Any other orientation would result in false impression of excessive curvature. It can be seen in Figure 5-5 that the promoter pattern R8 is more curved at the center than the promoter sequence of β-globin gene (glo). This structural variation that changes the signature of β-globin gene is responsible for RNA polymerase to recognize and thus facilitate the transcription.

5.4 CONCLUSION

Highly intricate process like transcription can be well captured using the hybrid approach of two novel intelligent tools. This approach helps us to study the effect of multiple base substitutions causing the increase in transcription efficiency. These simulation results can be used as a guide in designing mutation experiments since a priori estimate of the possible outcome of multiple mutations can be obtained. This methodology has also captured the role of DNA superstructures in gene expression. Such a hybrid approach, involving an ANN that maps the given inputs onto the outputs, and a genetic algorithm (GA) that maximizes the output by searching the input space of ANN can be used for optimizing any biological property.

5.5 REFERENCES

1. McKnight, S.L. and Kingsbury, R. (1982) *Science,* **217**, 316-324.
2. McKnight, S.L., Kingsbury, R.C., Spence, A. and Smith M. (1984) *Cell*, **37**, 253-262.
3. Graves, P.F. Johnson, S.L. McKnight (1986) *Cell*, **44**, 565-576.
4. Giodoni, J.T. Kadonaga, H. Barrera-Saldana, K. Takahashi, P. Chambom and Tijian, R. (1985) *Science*, **230**, 511-517.
5. Grosveld, G.C., de Boer, E., Shewmaker, C.K., Flavell, R.A. (1982) *Nature*, **295**, 120-126.
6. Edgar, T.F. and Himmelblau, D.M. (1989) Optimization of Chemical Processes. McGraw-Hill.
7. Nair T.M., Tambe S.S. and Kulkarni B.D. (1995) *Comp. Applic. Biosci*., **3**, 293-300.
8. Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1986) *Nature*, **323**, 533-536.
9. Rumelhart, D.E. and McClelland, J.L. (1986) *Parallel and Distributed Processing: Explorations in the Microstructure of Cognition.* MIT Press, Cambridge, MA.
10. Nair, T.M., (1996) In Tambe, S.S., Kulkarni, B.D. and Deshpande, P.B. (Eds), *Elements of Artificial Neural Networks with Selected Applications in Chemical Engineering and Biological Sciences.* Simulation and Advanced Controls, Inc., Louisville.
11. Goldberg, D.E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, Mass.
12. Davis, L., ed. (1991) *Handbook of Genetic Algorithms.* Van Nostrand Reinhold, New York.
13. Holland, J.H. (1992) *Adaptation in Natural and Artificial Systems*. 2nd ed. (University of Michigan Press, Ann Arbor.
14. Hanagandi, V.; Ploehn, H.; Nikolaou, M. (1996) *Chem. Eng. Sci.*, **51**, 1071.
15. Schoenauer, M.; Michalewicz, Z. (1997) Evolutionary Computation. *Control and Cybernetics*, **26**, 307-338.
16. Efstratiadis, A., Posakony, J. W., Maniatis, T., Lawn, R.M., O'Conell, C., Spriz, R.A., Deriel, J.K., Forget, B.G., Weissmann, S.M., Slightom, J.L., Blechl, A.E., Smithies, O., Barale, F.E., Shoulders, C.C. & Proudfoot, N.J. (1980) *Cell* **21**,

653-668.

17. Freeman, J.A. and Skapura, D.M. (1992) *Neural Networks Algorithms, Applications, and Programming Techniques*. Addison-Wesley.
18. Tambe, S.S., Kulkarni, B.D. and Deshpande, P.B. (Eds) (1996), *Elements of Artificial Neural Networks with Selected Applications in Chemical Engineering and Biological Sciences*. Simulation and Advanced Controls, Inc., Louisville.
19. Myers, R.M., Lerman, S.L. and Maniatis, T. (1985) *Science*, **229**, 242-247.
20. Myers, R.M., Tilly, K. and Maniatis, T. (1986) *Science*, **232**, 613-618.
21. Demeler, B. and Zhou, G. (1991) *Nucl. Acids Res.*, **19**, 1593-1599.
22. Pearson, W. R. (1990) *Methods Enzymol*, **183**, 63-98.
23. Shpigelman, E.S., Trifonov, E.N. and Bolshoy, A. (1993) *Comput. Applic. Biosci.*, **9**, 435-440.
24. Bolshoy, A., McNamara, P., Harrington, R.E. and Trifonov, E.N. (1991) *Proc. Natl. Acad. Sci. USA*, **88**, 2312-2316.
25. Trifonov, E.N. and Ulanovsky, L.E. (1987) In Wells, R.D. and Harvey, S.C. (eds) Unusual DNA structures. Springer-Verlag, Berlin, pg. 173-187.

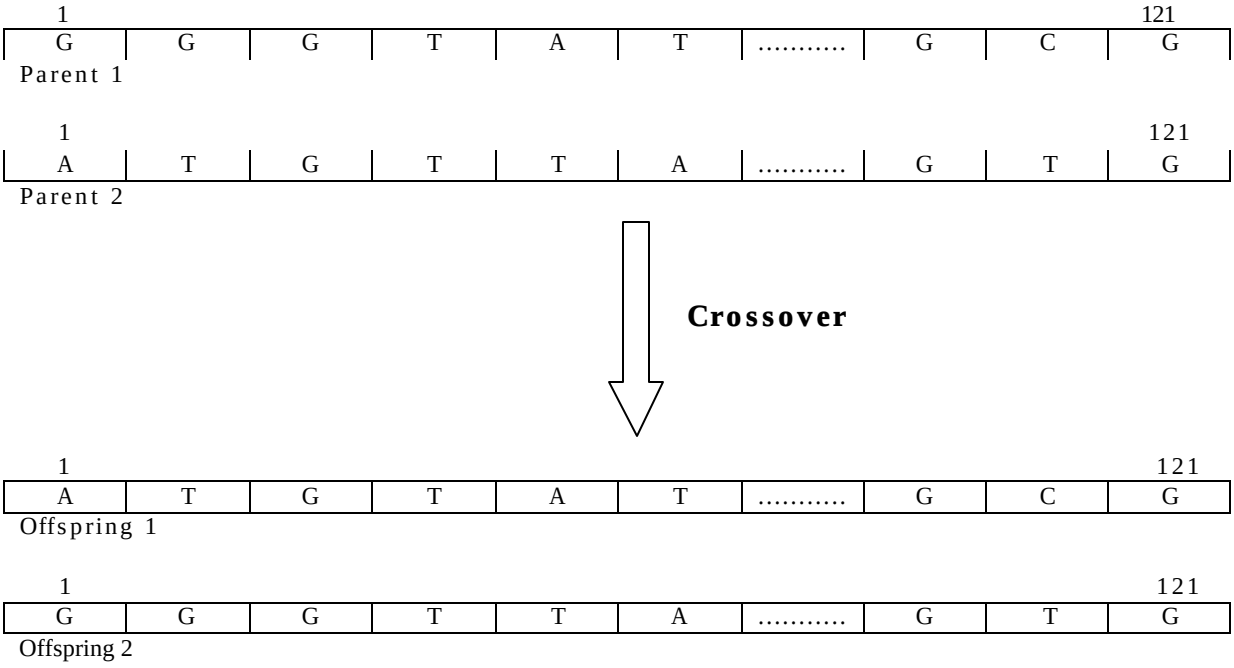


Figure 5-1: Basic crossover of the nucleotide sequence of the two parent strings.

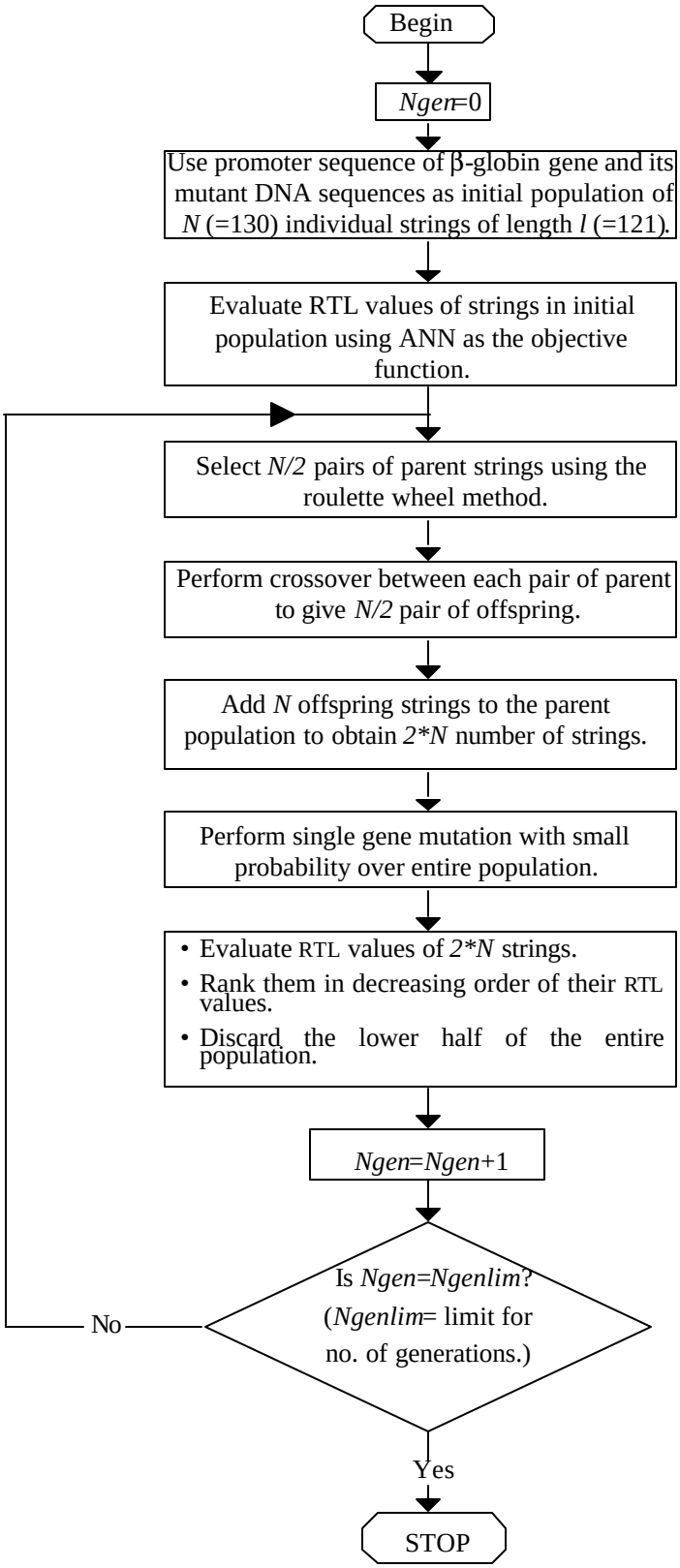


Figure 5-3: Flow chart for the implementation of ANN-GA strategy for the optimization of transcription efficiency (in terms of its RTL value) of β-globin gene.

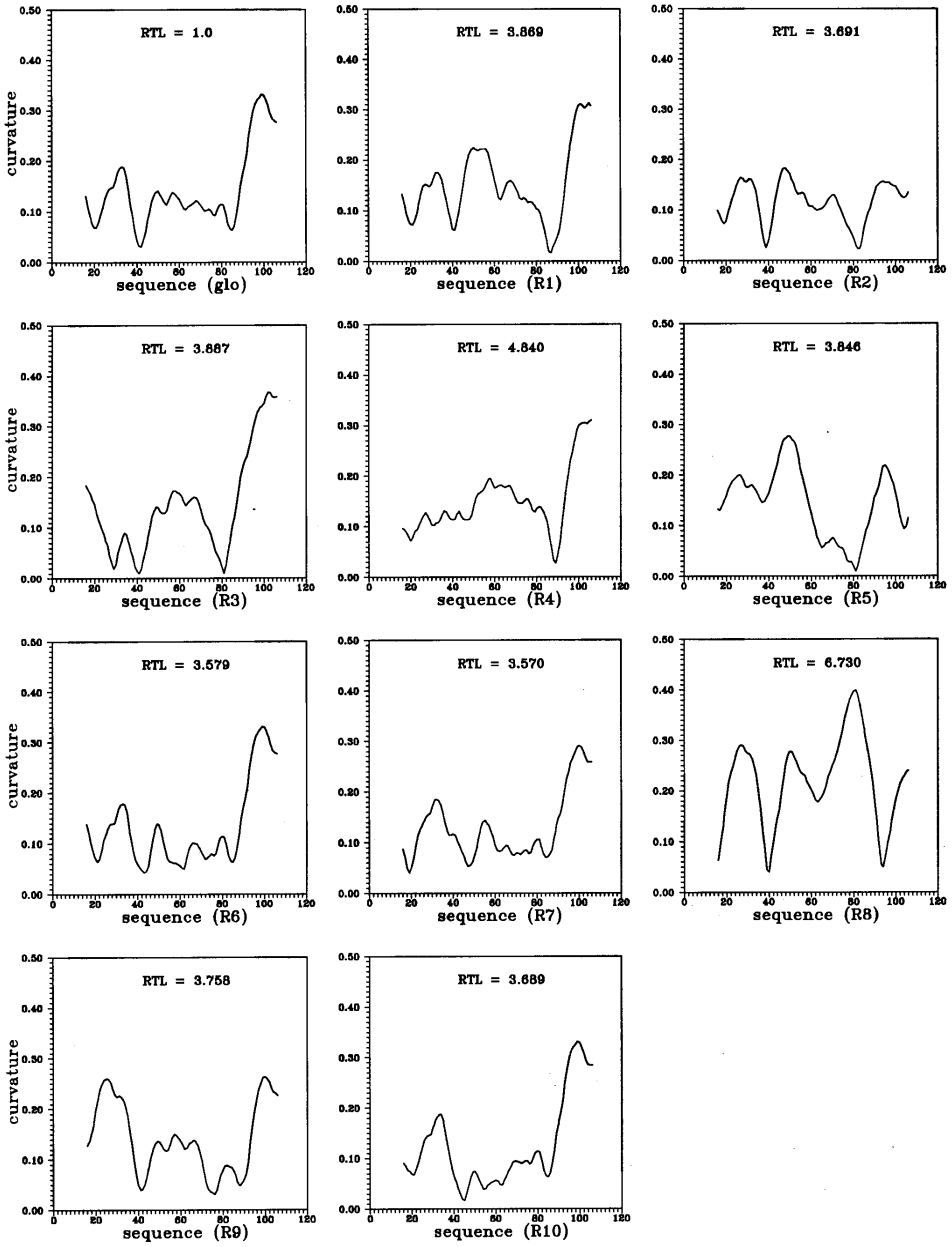


Figure 5-4: Comparison of the curvature map of the upstream region of β-globin gene (glo, RTL=1.0) and ANN-GA predicted promoter patterns of β-globin gene (R1 to R10, RTL > 3.5). Curvature is given in DNA curvature units [25] which is the mean DNA curvature in the crystalline nucleosome (1/42.8 $\overset{o}{A}$).

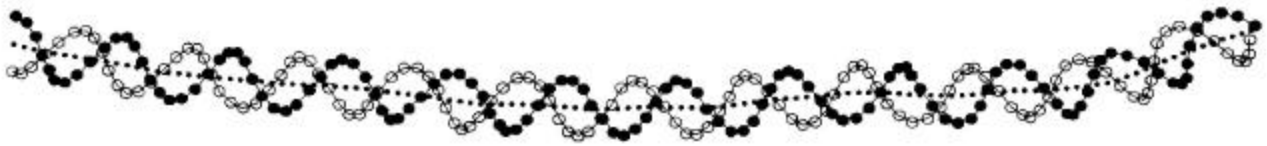


Figure 5-5a: DNA path of β-globin gene (glo, RTL=1.0) calculated using CURVATURE software.

Figure 5-5b: DNA path of the ANN-GA predicted promoter sequence (R8, RTL=6.73) calculated using CURVATURE software.

Table I: Sequence (simulated patterns of upstream region of b-globin gene) details along with their ANN-GA predicted Relative Transcription Level (RTL) value.

No.	Relative Transcription Level (RTL)
R1	**3.8690**
R2	**3.6919**
R3	**3.8870**
R4	**4.8404**
R5	**3.8465**
R6	**3.5799**
R7	**3.5703**
R8	**6.7307**
R9	**3.7589**
R10	**3.6896**

Table II: Comparison of upstream region of b-globin gene with ANN-GA predicted promoter patterns for sequence homology using FASTA package.

```
1210 residues in    10 sequences
 results sorted and z-values calculated from opt score
   10 scores better than 1 saved, ktup: 6, fact: 6
 DNA matrix, gap penalties: -16,-4
 joining threshold: 46, optimization threshold: 31, width: 16
  scan time:  0:00:00
The best scores are:                                      initn init1 opt
R6, 121 bases, 34DF602C checksum.                          555 555  555
R10, 121 bases, 1203C265 checksum.                         535 535  537
R9, 121 bases, EA1BF176 checksum.                          492 492  492
R7, 121 bases, BC9B2289 checksum.                          453 453  470
R3, 121 bases, 492F8031 checksum.                          461 461  461
R1, 121 bases, ACD08F2C checksum.                          450 450  452
R2, 121 bases, C5BA8B94 checksum.                          417 417  438
R5, 121 bases, A7CC5C50 checksum.                          363 363  401
R8, 121 bases, F8B2CAAC checksum.                          365 365  380
R4, 121 bases, A6543FA4 checksum.                          286 286  326
>>R6, 121 bases, 34DF602C checksum.                        (121 nt)
initn:  555  init1:  555  opt:  555
  95.8% identity in 120 nt overlap

               10        20        30        40        50        60
glo,    CGTAGAGCCACACCCTGGTAAGGGCCAATCTGCTCACACAGGATAGAGAGGGCAGGAGCC
        X:::::::::::::::::::: ::::::::::::::: : :::::::::::: ::::
R6,     CGTAGAGCCACACCCTGGTAAGCGCCAATCTGCTCACACTGTATAGAGAGGGCAGAAGCC
               10        20        30        40        50        60

               70        80        90       100       110       120
glo,    AGGGCAGAGCATATAAGGTGAGGTAGGATCAGTTGCTCCTCACATTTGCTTCTGACATAG
        ::: ::::::::::::::::::::::::::::::::::::::::::::::::::::::X
R6,     AGGACAGAGCATATAAGGTGAGGTAGGATCAGTTGCTCCTCACATTTGCTTCTGACATAG
               70        80        90       100       110       120

glo,    T

R6,     G

>>R10, 121 bases, 1203C265 checksum.                       (121 nt)
initn:  535  init1:  535  opt:  537
  94.2% identity in 120 nt overlap

               10        20        30        40        50        60
glo,    CGTAGAGCCACACCCTGGTAAGGGCCAATCTGCTCACACAGGATAGAGAGGGCAGGAGCC
         ::  X:::::::::::::::::::::::::::::::::: :::: :::: ::::
R10,    AGTTAAGCCACACCCTGGTAAGGGCCAATCTGCTCACACAGGATACAGAGTGCAGAAGCC
               10        20        30        40        50        60

               70        80        90       100       110       120
glo,    AGGGCAGAGCATATAAGGTGAGGTAGGATCAGTTGCTCCTCACATTTGCTTCTGACATAG
         :::::::::::::::::::::::::::::::::::::::::::::::::: :::::
R10,    GGGGCAGAGCATATAAGGTGAGGTAGGATCAGTTGCTCCTCACATTTGCTTCTGTCATAG
               70        80        90       100       110       120

glo,    T
        X
R10,    T
```

```
>>R9, 121 bases, EA1BF176 checksum.                  (121 nt)
initn:  492  init1:  492  opt:  492
  90.0% identity in 120 nt overlap

               10        20        30        40        50        60
glo,   CGTAGAGCCACACCCTGGTAAGGGCCAATCTGCTCACACAGGATAGAGAGGGCAGGAGCC
        X::::: ::::: : :  ::: :::::::::::::::::::::::::: :::::::::
R9,    TGTAGAGGCACACGCGGTGAAGAGCCAATCTGCTCACACAGGATAGAGAGCGCAGGAGCC
               10        20        30        40        50        60

               70        80        90       100       110       120
glo,   AGGGCAGAGCATATAAGGTGAGGTAGGATCAGTTGCTCCTCACATTTGCTTCTGACATAG
       : :::::::::::::::: :::::::: ::::::::::::::: ::::::: :::::
R9,    ATGGCAGAGCATATAAGGTGCGGTAGGATTAGTTGCTCCTCACATTAGCTTCTGGCATAG
               70        80        90       100       110       120

glo,   T
       X
R9,    T

>>R7, 121 bases, BC9B2289 checksum.                  (121 nt)
initn:  453  init1:  453  opt:  470
  87.6% identity in 121 nt overlap

               10        20        30        40        50        60
glo,   CGTAGAGCCACACCCTGGTAAGGGCCAATCTGCTCACACAGGATAGAGAGGGCAGGAGCC
       ::: :: X:::::::::::  ::::::::::::::: ::::: ::: ::: :: ::::
R7,    CGTTGACCCACACCCTGGTAGCGGCCAATCTGCTCACAGAGGATCGAGTGGGGAGTAGCC
               10        20        30        40        50        60

               70        80        90       100       110       120
glo,   AGGGCAGAGCATATAAGGTGAGGTAGGATCAGTTGCTCCTCACATTTGCTTCTGACATAG
        ::: ::: : ::::::::::::::: :::::: :::::::::::::::::::::::::
R7,    TGGGTAGATCGTATAAGGTGAGGTAGGCTCAGTTCCTCCTCACATTTGCTTCTGACATAG
               70        80        90       100       110       120

glo,   T
       X
R7,    T

>>R3, 121 bases, 492F8031 checksum.                  (121 nt)
initn:  461  init1:  461  opt:  461
  86.8% identity in 121 nt overlap

               10        20        30        40        50        60
glo,   CGTAGAGCCACACCCTGGTAAGGGCCAATCTGCTCACACAGGATAGAGAGGGCAGGAGCC
       X:::::::::::::::::::  ::: ::::::::::::::::::::::::::::::::
R3,    CGTAGAGCCACACCCTGGTAAGACCCACTCTGCTCACACAGGATAGAGAGGGCAGGAGCC
               10        20        30        40        50        60

               70        80        90       100       110       120
glo,   AGGGCAGAGCATATAAGGTGAGGTAGGATCAGTTGCTCCTCACATTTGCTTCTGACATAG
       : ::  :::::::::::::: ::: :  :::  ::::: ::: ::::::: : :::::::
R3,    AAGGGTGAGCATATAAGGTGGGGTGGCCTCACATGCTCTTCAAATTTGCTGCGGACATAG
               70        80        90       100       110       120

glo,   T
       X
R3,    T
```

```
>>R1, 121 bases, ACD08F2C checksum.                  (121 nt)
initn:  450  init1:  450  opt:  452
  86.0% identity in 121 nt overlap

               10        20        30        40        50        60
glo,    CGTAGAGCCACACCCTGGTAAGGGCCAATCTGCTCACACAGGATAGAGAGGGCAGGAGCC
        X:::::::::::::::::::::: :::::::: :: ::::::::::::::::: : ::::
R1,     CGTAGAGCCACACCCTGGTAAGGCCCAATCTGATCCCACAGGATAGAGAGGGAATGAGCA
               10        20        30        40        50        60

               70        80        90       100       110       120
glo,    AGGGCAGAGCATATAAGGTGAGGTAGGATCAGTTGCTCCTCACATTTGCTTCTGACATAG
           :::::::::::: :: :::::::::::: : ::: :::::: ::::::::::::X :
R1,     GACGCAGAGCATATTAGATGAGGTAGGATCGGGTGCCCCTCACTTTTGCTTCTGACAGAT
               70        80        90       100       110       120

glo,    T
        :
R1,     T

>>R2, 121 bases, C5BA8B94 checksum.                  (121 nt)
initn:  417  init1:  417  opt:  438
  85.0% identity in 120 nt overlap

               10        20        30        40        50        60
glo,    CGTAGAGCCACACCCTGGTAAGGGCCAATCTGCTCACACAGGATAGAGAGGGCAGGAGCC
        ::::: X:::::::: :::: :::::::::::::::::: :: ::::: ::::::::::::
R2,     CGTAGCGCCACACCCAGGTATGGGCCAATCTGCTCACACCGGTTAGAGCGGGCAGGAGCC
               10        20        30        40        50        60

               70        80        90       100       110       120
glo,    AGGGCAGAGCATATAAGGTGAGGTAGGATCAGTTGCTCCTCACATTTGCTTCTGACATAG
        :::::: ::: :::::::::  : ::::: : : ::::::::   ::::::: ::::::X
R2,     AGGGCATAGCCTATAAGGTGTCGCAGGATTAATGGCTCCTCAGCGTTGCTTCGGACATAG
               70        80        90       100       110       120

glo,    T

R2,     G

>>R5, 121 bases, A7CC5C50 checksum.                  (121 nt)
initn:  363  init1:  363  opt:  401
  82.2% identity in 118 nt overlap


               10        20        30        40        50        60
glo,    CGTAGAGCCACACCCTGGTAAGGGCCAATCTGCTCACACAGGATAGAGAGGGCAGGAGCC
        X:::::::::::::::: :::::::::::::: :::: ::  :::::::::  : : :::
R5,     CGTAGAGCCACACCCTGTTAAGGGCCAATCTGATCACCCATTATAGAGAGGAAACGGGCC
               10        20        30        40        50        60

               70        80        90       100       110       120
glo,    AGGGCAGAGCATATAAGGTGAGGTAGGATCAGTTGCTCCTCACATTTGCTTCTGACATAG
        :::::  : : ::::: :::  : ::: ::::::::::::::::X :: ::: :::::
R5,     AGGGCGCACCTTATAACGTGGTGAAGGTTCAGTTGCTCCTCACATCTGTTTCCGACATGC
               70        80        90       100       110       120

glo,    T

R5,     T
```

```
>>R8, 121 bases, F8B2CAAC checksum.                  (121 nt)
initn:  365  init1:  365  opt:  380
  79.3% identity in 121 nt overlap

               10        20        30        40        50        60
glo,   CGTAGAGCCACACCCTGGTAAGGGCCAATCTGCTCACACAGGATAGAGAGGGCAGGAGCC
       ::: X::::::: :::::: ::: ::::: ::::: ::::::   ::::::::: :::::
R8,    CGTGGAGCCACATCCTGGTGAGGCCCAATATGCTCTCACAGGTCCGAGAGGGCAAGAGCC
               10        20        30        40        50        60

               70        80        90       100       110       120
glo,   AGGGCAGAGCATATAAGGTGAGGTAGGATCAGTTGCTCCTCACATTTGCTTCTGACATAG
       :::::::::   ::::  :::::: :: : :::: :::  ::::::::::::X   :: :
R8,    AGGGCAGAGGGCATAAACTGAGGTCGGTTGAGTTTCTCTGCACATTTGCTTCTTCTATTG
               70        80        90       100       110       120

glo,   T
       :
R8,    T

>>R4, 121 bases, A6543FA4 checksum.                  (121 nt)
initn:  286  init1:  286  opt:  326
  74.4% identity in 121 nt overlap

               10        20        30        40        50        60
glo,   CGTAGAGCCACACCCTGGTAAGGGCCAATCTGCTCACACAGGATAGAGAGGGCAGGAGCC
       :::: :: X::::::: :::::::::::::::: :: ::     : :: :  :::: : :
R4,    CGTAAAGACACACCCTTGTAAGGGCCAATCTGTTCTCAAGCTTTCGATATAGCAGAAACA
               10        20        30        40        50        60

               70        80        90       100       110       120
glo,   AGGGCAGAGCATATAAGGTGAGGTAGGATCAGTTGCTCCTCACATTTGCTTCTGACATAG
       ::::::  :::: ::   :::   :: :::::::::: ::::::::::::X :  ::::
R4,    AGGGCATCGCATGTAGTCTGACTCAGAATCAGTTGCTTCTCACATTTGCTTCAGGAATAG
               70        80        90       100       110       120

glo,   T
       :
R4,    T
Library scan:  0:00:00  total CPU time:  0:00:00
```

Table III: Effective mutation points for ANN-GA predicted promoter patterns in accordance with various sub-regions.

No.	-101 to -96 region	-95 to -87 (CACCC box)	-86 to -78 region	-77 to -72 (CCAAT box)	-71 to -31 region	-30 to -26 (TATA box)	-25 to cap site	Below cap site
R1	_	_	-78GC*	_	-69CA -66AC -49CA -47GT -41AG -40GA -39GC	-27AT	-24GA -11AG -9TG -5TC	+3AT +17TG +19GT
R2	-96AC	_	-86TA -81AT	_	-62AC -59AT -53AC -35GT -31AC	_	-21AT -20GC -18TC -12CT -10GA -8TG	+2CG +3AC +4TG +12TG
R3	_	_	-78GC -79GA	-74AC	-40GA -37CG -36AT	_	-21AG -17AG -15GC -14AC -10GC -9TA -3CT	+2CA +10TG +12TG
R4	-97GA	-94CA	-85GT	_	-69CT -66AT -63CA -62AG -61GC -60GT -59AT -57AC -54GT -52GT -51GA -46GA -44GA -42CA -35GT -34AC	-29AG -26AG	-25GT -24GC -20GC -19GT -18TC -15GA -4CT	+12TA +14AG +15CA

* -78GC indicates that G at -78 position is mutated by C.

Table III continued…

No.	-101 to -96 region	-95 to -87 (CACCC box)	-86 to -78 region	-77 to -72 (CCAAT box)	-71 to -31 region	-30 to -26 (TATA box)	-25 to cap site	Below cap site
R5	_	_	-84GT	_	-69CA -64AC -61GT -60GT -50GA -49CA -47GC -45AG -36AG -35GC -33GC -31AT	_	-25GC -21AG -20GT -18TA -14AT	+5TC +8CT +12TC +18AG +19GC
R6	_	_	-79GC	_	-62AT -60GT -46GA -38GA	_	_	+20TG
R7	-98AT	-95GC	-81AG -80GC	_	-63CG -57AC -53AT -49CG -46GT -41AT -37CT -33GT -31AG	_	-14AC -7GC	_
R8	-98AG	-89CT	-82AG -78GC	-72CA	-66AT -59AT -58TC -57AC -47GA -32CG -31AG	-30TC	-25GA -24GC -17AC -14AT -12CG -7GT -3CT -2TG	+13GT +14AC +15CT +18AT
R9	-101CT	-94CG -88CG	-86TG -84GT -83TG -79GA	_	-51GC -40GT	_	-21AC -12CT	+6TA +14AG
R10	-101CA -98AT -97GA	_	_	_	-56GC -51GT -46GA -41AG	_	_	+14AT

CHAPTER

6

Compilation and analysis of mycobacterial promoters

In this chapter, we have compiled 125 mycobacterial promoter sequences, out of which 80 promoters have their transcription start-site (TSS) mapped while the other 45 are the putative promoters. Mycobacterial promoters have been analyzed for various features like: i) TSS, ii) -35 and −10 regions, iii) σ factor, iv) spacer length, v) upstream region of −35 box, and vi) % G+C content. These features are compared to similar features known for *E. coli* promoters. Further, the study suggests a broad classification of these promoters into three main types viz., i) *E. coli* type, ii) Mycobacterial (*Non-E. coli*) type, and iii) Extended −10 promoters. The results throw some light on the mycobacterial transcription machinery and structure of mycobacterial promoters, which is an important step to understand the low level of its transcription, and the possible mechanisms of regulation of gene expression.

6.1 INTRODUCTION

The genus *Mycobacterium* is of immense importance to human health because pathogenic species like *Mycobacterium tuberculosis*, *Mycobacterium leprae* etc. belong to this group. The avirulent strain of *Mycobacterium bovis*, which has been extensively used as a tuberculosis vaccine, *BCG* (*bacille Calmette-Guerin*), is also a very attractive vector for the construction of live recombinant vaccines particularly because of its strong immunogenicity. Thus, it is necessary to understand the essential features of transcription machinery for clear understanding of gene expression in these organisms.

Transcription is the very first and critical step in the process of gene expression. Transcription initiation involves interplay between RNA polymerase (RNAP) and promoter region. RNAP occupies the central role in transcription process. In prokaryotes, structure and function of enzyme RNAP seems to be conserved during evolution. The size, composition and function of different subunits of core polymerase do not vary much in different bacteria. On the other hand, promoter structures vary significantly from species to species and even within species depending on the kind of sigma factor (protein that binds to core enzyme and direct correct initiation) bound to the polymerase. Further, different trans-factors influence promoter recognition by the holoenzyme. In short, transcription only occurs from defined sites and in a specific direction, and the nature of the promoter will influence the affinity of the RNAP for that site, and hence determine the efficiency of transcription. Transcription efficiency is ultimately the major determinant of the level of gene expression.

Mycobacterial genome has high G+C content. Since the G+C content of genome affects codon usage and promoter recognition sites in an organism, it is reasonable to predict that transcription and other regulatory processes in *Mycobacteria* may differ from *E. coli* and many other bacteria. For expression of mycobacterial genes, *Streptomyces* is shown to be preferred host compared to *E. coli*. This is mainly because *Streptomyces* also has a high G+C content and they appear to be less stringent than *E. coli* in their promoter specificity [1]. *Mycobacteria* and *Streptomyces* belong to the same bacterial order i.e., *Actinomycetales*; hence, they may share some similarities in their transcriptional signals.

A significant finding that *Mycobacteria* have a low transcription rate and a low RNA content per unit DNA was reported in late seventies by Harshey and Ramkrishnan [2]. Understanding the reasons for this low level of transcription, and the possible mechanisms of regulation of gene expression, requires examination of the mycobacterial transcription machinery and the structure of mycobacterial promoters. With this objective, we have compiled different mycobacterial promoters and analyzed their DNA sequences for various features in this chapter.

Transcription is well studied in *E. coli* as compared to *Mycobacteria*. *E. coli* promoter paradigm forms the basis to analyze promoters in other systems. In *E. coli*, a large number of the genes that are expressed during normal vegetative growth have recognizably similar sequences at -35 and -10 positions (TTGACA and TATAAT, respectively) with respect to the transcription start site (TSS). The spacer-length between these two conserved regions is usually 15-20 bases. A combination of conserved -10 and -35 elements along with optimal spacer length (17 ± 1 bp) is referred to as the typical *E. coli* consensus promoter. More precisely, *E. coli* consensus promoter is recognized by RNAP when the enzyme is combined with one specific sigma factor, sigma 70. Under certain circumstances, sigma 70 is replaced by other sigma factors, and the promoter specificity of the RNAP is altered so that a different group of genes is expressed A majority of promoters using sigma factor 70 have at least two of the three most highly conserved bases in the -10 (TA…T) region, and at least one of the most highly conserved residues in the -35 (TTG…) region [3]. The majority of *E. coli* promoters fall into two basic categories: (i) those recognized by $E\sigma^{70}$, the activities of which are modulated by negative and positive regulators that must 'communicate directly' with the RNAP; and (ii) those promoters recognized by $E\sigma^{54}$, which are mainly regulated by activation, where the location of activator binding site could be remote from the binding of the RNAP [4].

6.2 COMPILATION AND ANALYSIS OF VARIOUS FEATURES OF THE MYCOBACTERIAL PROMOTERS

To define the DNA sequence features associated with mycobacterial RNAP, we have compiled 125 mycobacterial promoter sequences. Out of these 125 promoters, TSS is mapped for the 80 promoters and the remaining 45 promoters are the putative promoters based on the location of their consensus sequence. In this

analysis, we have considered a long stretch of nucleotides in the promoter region for the following reasons: i) RNAP from *E. coli* protects a large region in the promoter; DnaseI footprinting experiments show that this coverage extends up to region 50 to 70 nucleotides including regions upstream and downstream of –35 and –10 sequence. Considering that the mycobacterial RNAP architecture is similar to that of *E. coli* [5], it is reasonable to expect larger area of occupancy by mycobacterial RNAP as well; and ii) in many promoters, regions upstream and downstream play important role in influencing promoter efficiency. Hence, we have considered the sequence stretches between –50 and +10 bp with respect to the TSS for the promoters where TSS is mapped. The promoter sequence length varies based on the availability of the nucleotide sequence upstream of the –35 region and downstream of the –10 region. For the putative promoters, we have documented the sequence stretch between 15 bp upstream region of –35 box and 20 bp downstream of the –10 region. In few cases, for the same gene two or more different sequence frames are considered based on the alternate consensus probability. The compilation is presented in Table I.

From the extensive studies on transcription in *E. coli*, it is clear that several factors can affect the strength of the promoter. Major factors which influence promoter strength are: (i) nucleotide sequence of the –35 region, (ii) nucleotide sequence of the –10 region, (iii) spacing between the –35 and –10 regions, (iv) nucleotide sequence (especially A+T content) in the 5' flanking region upstream from the –35 regions [46]. In order to make a meaningful comparison, for the mycobacterial promoters, we have listed i) total length of the promoter region, ii) -35 region, iii) spacer length, iv) occurrence of TG motif, v) -10 region, vi) distance between the –10 region and TSS, vii) TSS, viii) % A+T content within individual promoter, and ix) % G+C content within individual promoter. A compilation of this information on 125 promoters is listed in Table II. In sections 2.1 to 2.7, we compare and contrast several features of mycobacterial promoter to those known for *E. coli* promoter sequences. Based on features discussed in these sections and choice of expression host, mycobacterial promoters can be broadly classified into three main types. Such classification may help us to better understand the mechanisms behind their expression. These classes are discussed in the sections 3.1 to 3.3. Later in sections 4.0 and 5.0, we have briefly documented stable RNA expression and influence of DNA topology and curvature on transcription.

6.2.1 Transcription Start Site

E. coli σ^{70} dependent promoters generally initiate transcription at a purine, adenine being more frequently utilized than guanine [47]. The selection of this nucleotide is influenced by the sequence around –35 and by the composition of the –2 to –5 positions [48-49]. We have analyzed the promoter compilation of the 80 sequences where TSS is mapped for occurrence of nucleotide type at TSS. The results of this analysis are shown in Figure 6-1. Occurrence of G at TSS among the four different nucleotides is about 49% and that of A is 28%. Thus, it appears that the purines (especially G) seems to be preferred first nucleotide of RNA.

GTG is often used as a start codon in *Mycobacteria* as opposed to ATG in *E. coli*. Mycobacterial genes show a relatively high degree of codon bias, reflected by a predominance of G or C at position 3, especially in *M. tuberculosis* [50].

In *E. coli* σ^{70} dependent promoters the spacing between the first nucleotide of the –10 region and the TSS is usually six or seven nucleotides, although functional examples between 4 and 10 nucleotides have been reported [47]. For our compilation of mycobacterial promoters, this distance also varies between 4 and 10 nucleotides. However, 92% of them show 5 to 8 nucleotides as a spacing distance between first nucleotide of the –10 element and TSS (see Table II).

6.2.2 –35 and –10 region

The importance of the –10 and –35 promoter sequences lies in their interaction with the σ factor bound to the RNAP for the initiation of transcription. The regions of σ factors responsible for binding to the –10 sequences are designated as 2.3 and 2.4, and for –35 sequences are specified as 4.2 [51].

Some mycobacterial promoters contain –35 and –10 regions, which resemble to *E. coli* σ^{70} type promoters. As can be seen from Table II, *M. tuberculosis rrnA PCL1*, *16S rRNA*, *ahpC*, *10kDa* (one with spacer length 17), *metA*, *rpsL*; *M. bovis BCG rRNA*, *ahpC*, *rpsL*; *M. leprae groE1*, *rpsL*; *M. smegmatis acetamidase*, *rrnB*, *rrnA P2*, *rrnA P3*, *rrnA PCL1*, *rpsL* (one with spacer length 18); *M. fortuitum rrnA PCL1*, *rrnA P2a*, *rrnA P2b*; *M. phlei rrnA PCL1*, *rrnA P2*; *Mycobacteriophage L5 71P2*; *M. neoaurum rrnA PCL1*, *rrnA P1*, *rrnA P3*, *rrnA P2*; *M. abscessus rrnA P4*, *rrnA PCL1*, *rrnA P2*, *rrnA P3*; and *M. chelonae rrnA PCL1*, *rrnA P4* promoters resemble *E. coli* σ^{70} type promoter.

M. smegmatis rpsL promoter has two potential –10 hexamers, i) five bp upstream of the TSS, which has 50% nucleotides matching with the typical *E. coli* σ^{70} region, and ii) eight bp upstream of TSS, which has 66% matched nucleotides. There are also two potential –35 regions, with spacer length 17 and 18 bp which has 33 and 50% matching nucleotides, respectively. By oligonucleotide directed mutagenesis, Kenny and Churchward [23] have shown that mutations in the *rpsL* promoter region, which result in deviation from the consensus *E. coli* sequences, abolished promoter activity. They have also demonstrated that –10 region present eight bp upstream of the TSS is recognized *in vivo* and the –35 region is not essential for promoter activity of *M. smegmatis*.

Dellagostin et al. [52] have demonstrated that in *M. smegmatis*, the *M. leprae 18 Kda* gene utilizes a single TSS located 66 bp upstream of the start codon. Immediately upstream of the TSS of *M. leprae 18 kDa*, putative -10 and –35 hexamers are present. They are similar to *E. coli* σ^{70} consensus promoter sequences. The region of the *18kDa* gene promoter (CTATATC) containing the putative –10 sequence, when compared to the *E. coli* –10 consensus sequence (TATAAT), shows a mismatch at either the first or the last T residue, both of which are highly conserved in *E. coli.* An alternative interpretation by the authors is that the functional *18 kDa* –10 sequence is a pentamer (TATAT).

Timm et al [53] have shown the importance of –10 region in promoter efficiency in *Mycobacteria* by point mutations in *M. fortuitum blaF* gene, and *M. smegmatis alrA* gene. Essentially these mutants map to the putative –10 hexamer, and increase the overall A+T content of the –10 hexamer, consequently resulting in increase in transcription efficiency.

Mycobacteriophage I3 promoters when compared to *E. coli* σ^{70} type promoters, showed greater sequence homology in the –35 region, ranging from 33 to 83% and comparatively weaker sequence homology in their –10 region, ranging from 17 to 50%. Not surprisingly these promoters are shown to be active in *E. coli* [42].

M. tuberculosis 85A has two putative –35 regions. One of these is positioned at 17 bp from the –10 region, showing 50% identity with the *E. coli* consensus promoter, and the other positioned at 22 bp from the –10 region, showing 83% identity with the *E. coli* consensus sequence. The second –35 region is identical to

the −35 region of the *M. leprae* and *M. tuberculosis 16S rRNA* promoter region. Interestingly, the −10 hexamer of the *85A* promoter shows some similarities to several *Streptomycete* promoters, such as the *kgmB-p, strpB-p, aacC9-p, afsA-p*, and *vph-p2* promoters. Moreover, like the *85A* promoter, these *Streptomycete* promoters are not typically expressed in *E. coli* [4].

Kremer et al [11] observed that deletion of 4 bp or insertion of 64 bp between the −10 and −35 regions of the *M. tuberculosis 85A* antigen promoter abolished only 50% of the promoter activity. Based on this finding the authors have suggested that although the sequence at the −35 position is essential for transcription, its location may not be critical, a feature similar to *Streptomyces*, and dissimilar to *E. coli* promoters.

Hoopes and McClure [54] have shown that during isomerization step of promoter-polymerase interactions, the DNA sequence around −10 region of the promoter opens up along with a conformational change in RNAP. In almost all the cases including eukaryotic systems, the AT rich region is perhaps crucial for the formation of open complex. In sharp contrast, many (30%) mycobacterial promoters have high GC content, instead of AT in their −10 region (Table II). Mycobacterial promoters having high GC content (≥ 50%) in their −10 region are *M. tuberculosis gyrA, cpn60, gyrB P1*, *85A*, *gyrB P2*, *katG* P_C, *T6, T80*, *gyrB P3*, *KatG* P_A, *purC, metA*; *M. bovis BCG 18K, mpb70*; *M. smegmatis S65*, *ask*, *rpsL* (one with spacer length 17), *ahpC*; *M. paratuberculosis pAJB303, pAJB86, pAJB300*, *pAJB304, pAJB73*, *pAJB301 pAJB125*, *pAJB305*; *M. fortuitum rrnA P1*, *rrnA P3*; and *M. phlei rrnA P1, rrnA P3*; *Mycobacteriophage I3 ORF1*; *M. avium pLR7*; *M. abscessus rrnA P4*, *rrnA P2*, *rrnA P3*; and *M. chelonae rrnA P2*, *rrnA P3, rrnA P4*. This raises the question, how DNA melting (isomerization step of promoter-polymerase interaction) is occurring in these mycobacterial promoters having high GC rich −10 region? One possibility is that this step might be controlled by specific sigma factors or some additional transcriptional activators may perform the task. Out of these promoters, *M. tuberculosis cpn60, 85A, M. bovis BCG mpb70*, *M. smegmatis ask* are shown to be non-functional in *E. coli* and hence they might be typical mycobacterial promoters. This observation encourages to say that promoters rich in GC at −10 region are a different class (and perhaps genuine) mycobacterial promoters.

It is interesting to note that although promoters *M. tuberculosis metA; M. abscessus rrnA P4, rrnA P2, rrnA P3; M. chelonae rrnA P4* have high GC content (=50%) in their –10 region, they also have *E. coli* σ^{70} type –10 region. The reason for such a pattern is G and C nucleotides are present at the third, fourth, and fifth position of –10 hexamer, keeping TA---T requirement of *E. coli* σ^{70} type –10 region undisturbed.

Interestingly, eight out of nine *M. paratuberculosis* promoters listed here showed high GC content in their –10 region. Thus, there is absence of *E. coli* σ^{70} type conserved –10 region in this species of *Mycobacteria*. *M. paratuberculosis* promoters show presence of at least one residue out of TTG… at –35 region, a feature typical of majority of *E. coli* σ^{70} type promoters. *M. paratuberculosis* promoters thus seem to be very different from the *E. coli* σ^{70} type consensus promoters. It will be interesting to study mutational analysis at the presumptive –10 and -35 region to assess the promoter strength and to carry out foot printing analysis with the RNAP to address the contact points. Such studies would ultimately reveal the characteristics features of "typical" *Mycobacteria* like promoters.

M. smegmatis and *M. tuberculosis* promoter analysis by Bashyam et al [6] showed that their –10 regions are highly similar to those of *E. coli* σ^{70} promoters, in contrast to their –35 regions, which can tolerate a greater variety of sequences. This could presumably be due to the presence of multiple sigma factors with different or overlapping specificities for –35 regions, like *Streptomyces* promoter. In case of promoters where nonfunctional –35 region is seen, occurrence of extended TG motif near –10 region is functionally significant. There are many promoters, which do contain TG motif next to –10 region. This is discussed in more details in the extended –10 promoter types (section 3.3).

Thus, the more general trend is, -10 consensus sequence of *E. coli* appears to be conserved in one group of mycobacterial promoters. A large variety of sequences can be accommodated in the –35 region [6, 55-56] as absence of a conserved –35 region is a distinctive feature of a class of mycobacterial promoters.

It is well-established fact that elements in the –35 and –10 regions are crucial in the transcription initiation process as σ^{70} RNAP holoenzyme makes direct contact in these two regions. Hence, to get an insight into transcription initiation mechanism among different mycobacterial species, it will be important to study the role of

consensus sequences and their percentage occurrence for particular nucleotide at each position in these conserved hexamers. We have carried out the analyses for available promoter sequences from different species of *Mycobacteria* with an idea to evaluate species specific differences, if any, which may reflect differential gene expression. Upon inspection of promoters of each species separately, we have calculated the percentage conserved homology for –35 and –10 regions for different mycobacterial species and listed them in Table III. For each position in the hexamer, we have considered the predominantly occurring nucleotide and its percentage homology in maintaining conserved sequence. Thus, percentage conserved homology obtained for entire mycobacterial promoter compilation is as follows: **-35:** T (87%), T (60%), G (66%), A (46%), C (56%), T (39%); and **–10:** T (70%), A (74%), T (34%), A (35%) / G (33%), C (34%) / G (27%), T (74%). For inter-species variation of the –35 and –10 consensus occurrence, readers are advised to refer Table II. This analysis reflects the large variations among the mycobacterial promoters characterized thus far, and suggests that the consensus sequences are representative of only a fraction of mycobacterial promoters. The variation in promoter structure may reflect the presence of larger number of σ factors in the genus *Mycobacteria* (see section 2.3).

6.2.3 σ factors

Sigma factors are essential components for promoter recognition and transcription initiation. All known σ factors belong to two different families: i) those evolutionary related to the *E. coli* housekeeping factor σ^{70}, and ii) those related to the alternative factor σ^{54} [57]. Each family of σ factors shows different promoter recognition, isomerization, and regulation properties [58]. $E\sigma^{70}$ does not show formation of stable closed-promoter complexes, and therefore transcription can be initiated spontaneously in the absence of activator proteins [59]. However, $E\sigma^{54}$ forms physically detectable closed-promoter complexes and is unable to initiate transcription spontaneously as it requires additional transcriptional factors (denominated enhancer-binding proteins) to initiate RNA synthesis [60].

The principal sigma factors of *M. smegmatis*, *M. tuberculosis* and *M. leprae* are nearly identical to the principal sigma factors of *Streptomyces auerofaciens*. They are also nearly identical to the principal sigma factor of *E. coli* (RpoD) in the region

responsible for binding to the –10 box, and differ substantially in the region involved in binding to the –35 box [6].

The genome sequence analysis of *M. tuberculosis* (genome size 4.1 Mb) has revealed presence of 14 sigma factors in the DNA of virulent strain [61]. Thus, organisms like *M. smegmatis* are larger than *M. tuberculosis* in their genome size, so it is obvious to have substantial number of sigma factors in them. The presence of a large number of sigma factors is a characteristic feature very similar to *Streptomyces* species and allows for greater transcriptional initiation flexibility as also for providing an efficient means of gene regulation in these organisms. The presence of many σ factors with different consensus sequence requirements may also be the reason for the large variations or heterogeneity in the –10 and –35 sequences of mycobacterial promoters as already discussed in section 2.2. The features viz., promoter sequence heterogeneity and plethora of σ factors seem to be a more general phenomenon of regulating transcription initiation specificity in the members of the *Actinomycetales*.

6.2.4 Spacer length

In *E. coli* σ^{70} type promoters, the optimal spacing between the –35 and –10 elements is 17±1 nucleotides, although, functional promoters with spacing ranging between 15 and 20 non-conserved nucleotides have also been reported [47, 62]. These σ^{70} class of promoters are the strongest when they have consensus –10 and –35 region along with optimal spacing of 17 bp separating the two conserved elements. Spacing less than 16 or more than 18 often results in conserved contact points lying on the opposite face of the DNA helix [3]. Mycobacterial promoters identified to date, show spacing between –10 and –35 regions as 7 to 24 bp (see Table II). As genus *Mycobacteria* seems to comprise of many sigma factors, it is expected that each type of sigma factor will require different spacing length, and thus explain to some extent the larger variation in the spacer length. We have analyzed the promoter compilation for percentage occurrence of each spacer length type. From Figure 6-2, it can be inferred that although, spacer length varies over a wide range (7 to 24 bp), occurrence of 17 (27%) and 18 bp (35%) as a spacer length is predominant. Thus, the major sigma factor recognition pattern in *Mycobacteria* appears to be similar to that of *E. coli* σ^{70} type.

6.2.5 Upstream region of the –35 box

Studies on *E. coli* promoter sequences have shown that the upstream element enhances the initial association of RNAP with the DNA. This association is independent of the presence of σ factor. Inspection of far-upstream region of –35 box, may provide insight into promoter architecture which can be compared to that of *E. coli*. Similarities, if any, might suggest common mechanisms of regulation between the *Mycobacteria* and *E. coli*. In most of the promoters in our compilation, such analysis did not reveal any special features. However, *M. tuberculosis glnA* (the one with spacer length 10), *KatG* P_C, *purC*, *ahpC*, *65kDa*; *M. bovis BCG ahpC*, *18K*, *mpb70*, *alpha*; *M. leprae 18kDa*, *28kDa*, *65kd*; *M. smegmatis gyrB*, *rrnA P1*, *rpsL* (the one with spacer length 17), *ahpC*, *M. paratuberculosis pAJB86*; *M. fortuitum rrnA P1*; *M. phlei rrnA P2*; *Mycobacteriophage I3 ORF2*; *Mycobacteriophage L5 71* P_{left}, *71 P1*; *M. neoaurum rrnA P1*; *M. abscessus rrnA P4*, *rrnA P2*, *rrnA P3*; *M. chelonae rrnA P2*, *rrnA P3*, *rrnA P4* contain occurrence of A_nT_m (n+m ≥ 3) stretch in the immediate upstream of –35 region. Out of these promoters *ahpC* from *M. smegmatis*, *M. tuberculosis*, and *M. bovis BCG*; M. *leprae 18kDa*, *28kDa*; *Mycobacteriophage I3 ORF2*; *Mycobacteriophage L5 71 P1*; and *M. abscessus rrnA P3* have more than 50% of A+T content. Hence, occurrence of A_nT_m (n+m ≥ 3) in the upstream region of –35 element is not surprising, but it is certainly remarkable for others where G+C content of the promoter is more than 50%. For these promoters, to accommodate high GC content, occurrence of GC intrusions might be somewhere other than upstream region. The *ahpC* promoter from *M. tuberculosis and M. bovis BCG*, *M. smegmatis rpsL*, *M. phlei rrnA P2*, *M. neoaurum rrnA P1*, *M. abscessus rrnA P4*, *rrnA P2*, *rrnA P3* contain *E. coli* σ^{70} type conserved hexamers along with the A_nT_m (n+m ≥ 3) tract in the upstream region of –35 box. Perhaps these might be the strongest promoters among *Mycobacteria*. All the promoters having A_nT_m (n+m ≥ 3) tract in the immediate upstream of –35 region are not repeated in phase with each other. But *M. tuberculosis KatG* P_C (the one with spacer length 22), *purC*, *ahpC; M. bovis BCG ahpC, alpha; M. leprae 65KD; M. smegmatis ahpC; M. abscessus rrnA P4, rrnA P2, rrnA P3; M. chelonae rrnA P3, rrnA P4* promoters have A_nT_m (n+m ≥ 3) tract repeated in phase with each other.

The *recA* gene of *M. tuberculosis* and *M. smegmatis* is regulated in a similar manner. In both the species, this gene contains upstream region, which has a

sequence motif with homology to Cheo box LexA regulatory site of *B. subtilis*, while there is no similarity to the SOS box of *E. coli.* The region of DNA 300 bp upstream of the *recA* gene was shown not to contain a promoter, suggesting that it functions as an upstream activator sequence [56]. The upstream region of *M. leprae 18kDa* promoter has also been shown to be essential for expression [52].

6.2.6 % G+C content

There is dramatic variation in the percentage of G+C content in the typical *E. coli* and mycobacterial promoters. Hence, we have evaluated the average value of percentage A+T and G+C content for each mycobacterial species. In general, occurrence of GC is high compared to AT, for mycobacterial promoters on species level (refer Table III). The mycobacterial promoters show a high G+C content than the corresponding *E. coli* promoters. There are few exceptions to this observation like *M. leprae 18kDa*, *M. smegmatis S6*, *S12*, *S18*, *S21*, *S30*, *S35*, and *S119*, whose G+C content is less than or equal to 40% (refer Table II).

M. tuberculosis 85A promoter region with spacer length 17 bp has 58% G+C content and that with the spacer length 22 bp has 61% G+C content (see Table II). In this respect, it is interesting to note that mycobacterial promoters having high GC content are usually better-expressed in *Streptomycetes* than *E. coli* [1].

M. tuberculosis promoters have a higher G+C content (58%) than the *M. smegmatis* promoters (50%) which may have a bearing on the differences in the gene expression between these two species.

It is clearly observed that overall G+C content (56%) for mycobacterial promoter compilation is high compared to G+C content (40%) of *E. coli* promoters listed by Harley and Reynolds [3]. It appears that upstream region of mycobacterial promoters is relatively more susceptible to GC intrusions to accommodate the higher GC content of its promoter region.

6.2.7 Comparison of Mycobacterial promoters with E. coli promoters

Despite the fact that mycobacterial promoters function inefficiently in *E. coli*, both the mycobacterial transcription machinery and the structure of mycobacterial promoters show marked conservation with those of *E. coli*. Diversity among the

mycobacterial promoters and σ factors however is greater. It is interesting to note that the promoters of gram-positive organisms show tighter consensus sequence requirements than those of *E. coli*, which in turn are more conserved than those of the *Mycobacteria*.

There is a great deal of heterogeneity in the consensus sequences of mycobacterial promoters. Such variations perhaps reflect diversity required in transcription regulation in *Mycobacteria* . Thus it is not surprising that *Mycobacteria* has two house-keeping sigma genes compared to one in *E. coli*.

The presence of large number of sigma factors with different consensus requirements may also be the reason for the large variation in -10 and -35 sequences of mycobacterial promoters. The -10 region of a class of mycobacterial promoters and the corresponding binding domain in the major sigma factor are highly similar to *E. coli* counterparts. In contrast, the sequences in -35 regions of mycobacterial promoters and corresponding binding domain in the major sigma factor are vastly different than their *E. coli* counterparts. *E. coli* RNAP have seven types of sigma factors and hence seven classes of promoters. *Mycobacteria* genome analyses have shown that they contain at least 14 sigma factors, so minimum number of promoter classes may be 14 in *Mycobacteria*.

Spacer length between -35 and -10 hexamer is not critical in *Mycobacteria*. The upstream region of mycobacterial promoters is relatively more susceptible to GC intrusions to accommodate the higher GC content of its promoter.

6.3 CLASSIFICATION

6.3.1 *E coli* type promoters

A significant minority of mycobacterial promoters such as *M. tuberculosis 65 kDa* [63], *M. bovis BCG* 64 kDa [64], and *M. leprae 65kD* [65], the biotin carrier protein of several species [66] has been shown to be expressed in *E. coli*. These organisms might share some similarities in their transcription initiation signals with *E. coli.*

Mycobacterial promoters controlling the expression of heat shock proteins are among the rare ones that have been shown to be active in *E. coli* [29]. There are sequence similarities between the mycobacterial heat shock promoters and consensus promoters recognized by σ^{60} and σ^{32} of *E. coli*. *M. paratuberculosis* P_{AN} [41], *M*

fortuitum blaF [53], *M. leprae 18kDa* [52] were also found to contain well conserved –10 and –35 regions and active in *E. coli*. Expression was however less efficient than the natural hosts in all the cases.

Recently, *M. tuberculosis KatG* [15] promoter has been characterized and shown to be active in *E. coli.* However, expression in *E. coli* was less efficient than its natural host. The analysis of this particular promoter has shown that there is only a partial sequence homology with *E. coli* σ^{70} type sequence, which may be one of the reasons for sub-optimal expression in *E. coli.*

Suzuki et al [26] have shown that *M. bovis BCG 16S rRNA* promoter can be expressed *in vivo* and *in vitro* using *E. coli* RNAP. This promoter showed sequence similarity to *E. coli* promoters. They have also shown that the strengths of *E. coli* and *M. bovis BCG rrn* are identical in *E. coli.* The *E. coli* RNAP did not utilize another putative promoter of the *BCG rrn*, which suggests that the second promoter may be recognized by a specific sigma factor not present in *E. coli*.

M. tuberculosis 38 kD gene can be expressed in *E. coli* from a lambda gt11 recombinant, independently of IPTG addition. This indicates that transcription can be initiated from within the mycobacterial insert, presumably (but not conclusively) from the natural promoter of the gene. However, analysis of the sequence does not reveal any regions upstream from the putative translation start position that resembles a consensus prokaryotic promoter [67].

Gene coding for the *28kD* antigen of *M. leprae* revealed one region with a considerable degree of homology to the Fur-binding site of iron-regulated promoters. Although the *28 kD* gene is not known to be iron-regulated in *M. leprae* (and indeed such studies are not easy for non-cultivable bacteria), this sequence comparison indicates that it is likely to be repressed by the presence of iron. This carries a further implication that in *M. leprae* (and presumably other bacteria) iron regulation of gene expression is mediated by a protein homologous to the Fur protein of *E. coli*.

In short, *M. tuberculosis 38kD* and *M. leprae 28kD* antigen genes are associated with DNA sequences that suggest the possibility of specific regulatory mechanisms, without such control having been demonstrated directly [19].

The putative promoter region of the *16S ribosomal RNA*-encoding gene (*rRNA*) of *M. leprae* exhibits promoter activity in Gram$^-$ (*E. coli*) and Gram$^+$ (*Bacillus subtilis*) bacteria [31]. Analysis of sequence revealed a promoter –like

sequence, which is close to the canonical −10 and −35 regions found in many bacteria [3]. It is interesting to note that −35 region and spacer length of this promoter is identical to the −35 region and spacer length of *E. coli rrnP2* promoter.

The *rpsL* promoter resembles *E. coli* σ^{70} type promoter in almost all-major mycobacterial species like *M. tuberculosis, M. bovis BCG, M. leprae,* and *M. smegmatis*. Since *rpsL* is highly conserved gene, the transcriptional regulatory features also seem to be conserved. Kenny and Churchward [23] have reported that the TG motif present upstream of the −10 hexamer can play a role in the activity of the *rpsL* promoter of *M. smegmatis* (see section 3.3).

6.3.2 Mycobacterial (*Non-E. coli*) type promoters

M. tuberculosis 85A promoter was one of the first promoters to be studied in some detail [11]. A surprising observation was that the promoter is not functional in *E. coli.* These results raise the possibility of the occurrence of an entirely different set of promoters in *Mycobacteria*, which are not recognized by *E. coli* transcriptional apparatus. With the characterization of many promoters now it is clear that a larger number of mycobacterial promoters fail to function in *E. coli*, constituting a different class.

The *M. tuberculosis 85A* promoter has −35 region showing significant resemblance to *E. coli* σ^{70} like −35 region, unconventional −10 region and/or the 22-bp spacer between the −10 and the −35 regions. However, it is shown that spacer position may not be critical for promoter activity in *Mycobacteria*, like *Streptomyces* promoter [68]. In spite of having significant resemblance to −35 region of *E. coli* σ^{70} like promoters, this raises the intriguing question of why these promoters are not expressed in *E. coli*. Clearly, additional facet of regulation has to be understood including the details about the transcriptional machinery.

M. tuberculosis recA promoter contain TCTAGT and TTGTCA as −10 and −35 consensus sequences resembling to *E. coli* σ^{70} type promoters. However, spacing (9 bp) between these two elements is very different from that found in *E. coli.* The *M. tuberculosis recA* gene is not expressed in *E. coli* from its own promoter. Hence, it is possible that either mycobacterial RNA polymerase recognizes the same motifs as does *E. coli* polymerase but at a different spacing or it binds to a different sequence in the −35 region [56].

M. paratuberculosis promoters listed by Bannantine et al. [40] showed some conservation at –35 regions with *E. coli* σ^{70} type promoters and high GC content in the –10 region, dissimilar feature with *E. coli* σ^{70} type promoters. Hence these promoters belong to the *Non-E. coli* or *Mycobacteria* type promoters.

M. tuberculosis ppgk promoter has high G+C content (61%) and the absence of an *E. coli* like promoter consensus with other mycobacterial promoters [24]. *M. tuberculosis cpn60, M. bovis BCG hsp60, mpb70* and *M. smegmatis ask* promoters showed absence of *E. coli* σ^{70} like consensus regions. *M. tuberculosis cpn60*, *M. bovis BCG mpb70* and *M. smegmatis ask* promoters have high GC content in their –10 region as well as in the entire promoter stretch (–50 to +10 bp), too. One of the possible reasons for mycobacterial promoters to be non-functional in *E. coli*, might be the poor interaction between the 4.2 region of the *E. coli* sigma factor and the –35 regions of mycobacterial promoters. The promoters, which are non-functional in *E. coli*, may be more typical for *Mycobacteria*.

6.3.3 Extended –10 promoters

A large number of mycobacterial promoters seem to have –10 conserved element without apparent conservation at –35 region. Amongst them, many possess extended –10 region characterized by dinucleotide element TG in the immediate upstream of –10 hexamer. The TG motif along with the functional –10 region is an important determinant of transcriptional strength in *Mycobacteria*. The influence of the TG element on transcriptional strength is also modulated by the sequences in the –35 region. It was also shown that the thermal energy requirement for open complex formation in an extended –10 promoter was less than that for a conventional –10/-35 promoter [69].

As can be seen from Table II, *M. tuberculosis T101, T129, groE, M bovis BCG hsp60 P2, M. leprae 16S rRNA, 18kDa, 65 kD, M. smegmatis S5, S6, S16, S19, S21, S119, recA, rpsL, M. fortuitum repA, Mycobacteriophage I3 ORF2, M. abscessus rrnA P4, rrnA PCL1, rrnA P2, rrnA P3, M. chelonae rrnA P2, rrnA PCL1, rrnA P3*, and *rrnA P4* promoter contains the TG element immediately upstream of the –10 region.

Thus, based on a sample size of 125 promoters, 20% of mycobacterial promoters contain TG motif. Analysis of 183 promoters from various species of

gram-positive bacteria [70-72] reveals that frequency of occurrence of the TG motif in these promoters is around 60%. In *E. coli* promoters, the TG motif occurs with a frequency of about 16% [73].

Bashyam and Tyagi [25] have suggested three possible roles of the extended –10 promoters, viz. i) in particular regions of the bacterial chromosome having sequence constraints, where it may be difficult to maintain two specific hexameric sequences, ii) to maintain a basal level of transcription in the case of promoters that contain a weak –35 region and are regulated by protein-DNA interactions in the –35 region, and iii) to facilitate transcription initiation at cold temperatures or when the sigma factor is proteolytically cleaved. According to Burns and Minchin [69] TG motif results in an altered DNA conformation which could either directly facilitate strand separation or allow additional DNA-protein contacts which would then promote open complex formation.

6.4 STABLE RNA EXPRESSION

In our compilation, rRNA promoters from different species of *Mycobacteria* are listed. Analysis of these promoters (see previous sections) reveal that they resemble to –10 and –35 consensus sequences of *E. coli* σ^{70} promoters along with the upstream A stretch, a putative up-element. These characteristics are very similar to *rRNA* promoters found in *E. coil* and many other bacteria, underlying the importance of evolutionary conservation of stable RNA expression. However, there are differences in *rRNA* copy numbers in different mycobacterial species suggesting the linkage between growth rate and ribosome synthesis to gene dosage.

Fast growing *Mycobacteria* (e.g. *M. smegmatis*) were shown to contain two sets of *rRNA* genes whereas slow-growing *Mycobacteria* contain only one set [74]. *M. tuberculosis*, *M. leprae* and *M. bovis BCG* contain only a single set of *rRNA* genes, and hence fit into the slow-growers group. However, among the slow-growing *Mycobacteria* there is a large variability in the growth rates of different species. Also, the possession of more than one operon per genome is not essential for rapid growth as each of the fast growers *M. chelonae* and *M. abscessus* has a single *rrn* operon per genome [75]. Gonzalez-y-Merchand et al [12] have shown that these species appear to have acquired additional promoters by a process of sequence

duplication. Thus, *Mycobacteria* have at least two levels at which rRNA synthesis is regulated.

6.5 INFLUENCE OF DNA TOPOLOGY AND CURVATURE ON TRANSCRIPTION

The topological state of DNA is an important determinant of its biological activity. In prokaryotes, DNA supercoiling is known to affect the transcription of several genes [76]. With the same supercoiling change, some genes are activated, others are inhibited and others are unaffected. Thus, different mechanisms appear to operate in different systems with the same supercoiling change [77-82]. The reason behind such complexity is that supercoiling can affect the DNA helix in various ways by modifying its energy (torsional strain) and structure (helical pitch and axial writhing) [83-84]. Any of these factors can influence promoter reactivity either directly or indirectly through effects on bending and wrapping of the DNA around proteins in chromatin-like structures [85-87].

M. smegmatis RNAP has a strong dependence on supercoiling of the DNA substrate for transcription from mycobacterial promoters [88]. Hence, the differences in the expression noted by Stover et al [89] may be because of the differences in the superhelical state of the DNA, which may play a direct role in the regulation of gene expression [90]. Thus, conformation of promoter DNA may play an important role for some of the mycobacterial promoters. However, the regulation of *gyr* operon expression of *M. smegmatis* provides certain contrasting and unique features. A single promoter located upstream of *gyrB* is responsive to changes in DNA supercoiling in a contrasting manner. The phenomenon of "relaxation stimulated transcription" (RST) observed for *gyr* promoter has certain interesting features and the mechanism appears to be different than that of *E. coli* [91].

In our study, we have analyzed the promoter sequences for distribution of intrinsic curvature. For this purpose, we have used CURVATURE software [92], which is based on the nearest-neighbor interactions between the two adjacent dinucleotides [93]. This analysis revealed that 62 promoters showed presence of curvature. Mycobacterial promoter sequences having their curvature maxima equal to or greater than 0.3 curvature units are listed in Table IV. These entries are sub-grouped based on where the curvature maxima is present within a sequence. The

graphical distribution of location of curvature maxima for mycobacterial promoters is shown in Figure 6-3. Out of 62 curved promoters, 29 promoters (47%) show their curvature maxima lying between the region −30 and −40 (Figure 6-3). They are *M. tuberculosis 85A, rrnA PCL1, 16S rRNA, 65 kDa, rpsL, 38 kDa*; *M. bovis BCG 64K, rpsL*; *M. leprae rpsL, M. smegmatis S4, S19, S21, rrnB, rrnA P1*; *M. paratuberculosis pAJB 305*; *M. fortuitum rrnA PCL1, rrnA P3*; *M. phlei rrnA PCL1, rrnA P2, rrnA P3*; *Mycobacteriophage L5 71* P_{left}; *M. abscessus rrnA P4, rrnA PCL1, rrnA P2, rrnA P3*; and *M. chelonae rrnA P2, rrnA PCL1, rrnA P3, rrnA P4*. Many other promoters viz., *M. tuberculosis T150, mpt64, metA*; *M. bovis BCG 23K, mpb64, 18K, mpb70*; *M. leprae 18Kda, 28 kDa, 65 Kda*; *M. fortuitum rrnA P2a*; *Mycobacteriophage L5 71 P2*; and *M. neoaurum rrnA P2* show their curvature maxima lying between −1 and −10 region. The presence of some conserved features in DNA curvature at the promoter region might suggest some common mechanism controlling transcription initiation.

6.6 CONCLUSION

It will be interesting to see which sigma recognize which sequence during RNAP-promoter interactions. Mutational analyses, would reveal the critical residues in different classes of promoters affecting promoter strength. Foot printing analysis with RNAP would be necessary towards identifying the contact points in −10 and −35 regions. Elucidation of mechanism of isomerization step of GC rich -10 region would be a challenging task. During promoter-RNAP interaction different events like DNA binding, DNA melting (isomerization step), Phosphodiester bond formation, and promoter clearance also need to be addressed. Thus, additional regulatory mechanism has to be unraveled to understand regulation of gene expression.

The number of *rRNA* genes is not solely responsible for growth rates observed in *Mycobacteria* because the organisms have at least two levels of rRNA synthesis regulation. Study of these different levels in details is required. Conformation of DNA is also important in mycobacterial regulation. Hence, it will be interesting to study what role does DNA structure plays in determining transcription efficiency of mycobacterial promoters.

Detailed analysis of promoter structure and function becomes important from entirely different perspective. Once sufficient number of promoters are studied, it is

important to determine their promoter strength. It will also help to identify regulatable promoters to engineer appropriate control circuits to function efficiently in *Mycobacteria*. Weak promoters will serve as a model system for transcription activation mechanism. Strong promoters can be used to exploit expression technology.

6.7 REFERENCES

1. Kieser, S., Moss, M. T., Dale, W.J. and Hopwood, D. A. (1986) *J. Bacteriol.*, **168**, 72-80.
2. Harshey, R. M. and Ramkrishnan, T. (1977) *J. Bacteriol.*, **129**, 616-622.
3. Harley, C. B. and Reynolds, R. P. (1987) *Nucl. Acids Res.*, **15**, 2343-2361.
4. Strohl, W. R. (1992) *Nucl. Acids Res.*, **20**, 961-974.
5. Seibenlist, U., Simpson, R. B. and Gilbert, W. (1980) *Cell*, **20**, 269-281.
6. Bashyam, M..D., Kaushal, D., Das Gupta, S. K. and Tyagi, A. K. (1996) *J. Bacteriol.*, **178**, 4847-4853.
7. Movahedzadeh, F., Colston, M. J. and Davis, E. O. (1997) *J. Bacteriol.*, **179**, 3509-3518.
8. Gonzalez-y-Merchand, J. A., Colston, M. J. and Cox, R. A. (1996) *Microbiology*, **142**, 667-674.
9. Unniraman, S. and Nagaraja, V. (2000) unpublished data.
10. Kong, T. H., Coates, A. R., Butcher, P. D., Hickman, C. J. and Shinnick, T. M. (1993) *Proc. Natl Acad Sci, USA*, **90**, 2608-2612.
11. Kremer, L., Baulard, A., Estaquier, J., Content, J., Capron, A. and Locht, C. (1995) *J. Bacteriol.*, **177**, 642-653.
12. Gonzalez-y-Merchand, J. A., Garcia, M. J., Gonzalez-Rico, S., Colson, M. J. and Cox, R. (1997) *J. Bacteriol.*, **179**, 6949-6958.
13. Verma, A., Kinger, A. K., Tyagi, J. S. (1994) *Gene*, **148**, 113-118.
14. Harth, G. and Horwitz, M. A. (1997) *J. Biol. Chem.*, **272**, 22728-22735.
15. Mulder, M. A. (1998) Identification of a novel function and analysis of the regulation of expression {Dissertation}. University of Cape Town, Cape Town.
16. Jackson, M., Berthet, F. X., Otal, I. et al. (1996) *Microbiology*, **142**, 2439-2447.
17. Kong, T. H., Coates, A. R., Hickman, C. J. and Shinnick, T. M. The Mycobacterium tuberculosis groE operon (unpublished). *Genbank Accession No*. X60350.
18. Wilson, T. M. and Collins, D. M. (1996) *Mol. Microbiol.*, **19**, 1025-1034.
19. Dale, J. W. and Patki, A. (1990) In: McFadden J, ed. Molecular biology of the mycobacteria. London: Surrey University Press, pp173-198.
20. Baird, P. N., Lucinda, M. C. and Coates, A. R. (1989) *J. Gen. Microbiol.*, **135**, 931-939.

21. Oettinger, T. and Andersen, A. B. (1994) *Infect. Immun.*, **62**, 2058-2064.
22. Vasanthakrishna, M., Kumar, N. V., Varshney, U. (1997), *Microbiology*, **143**, 3591-3598.
23. Kenny, T. J. and Churchward, G. (1996) *J. Bacteriol.*, **178**, 3564-3571.
24. Hsieh, P. C., Shenoy, B. C., Samols, D. and Phillips, N. F. (1996) *J. Biol. Chem.*, **271**, 4909-4915.
25. Bashyam, M. D. and Tyagi, A. K. (1998) *J. Bacteriol.*, **180**, 2568-2573.
26. Suzuki, Y., Nagata, A. and Yamada, T. (1991) *Antonie van Leeuwenhoek*, **60**, 7-11.
27. Yamaguchi, R., Matsuo, K., Yamazaki, A. et al. (1989) *Infect. Immun.*, **57**, 283-288.
28. Terasaka, K., Yamaguchi, R., Matsuo, K., Yamazaki, A., Nagai, S. and Yamada, T. (1989) *FEMS Microbiol. Lett.*, **49**, 273-276.
29. Thole, J. E., Keulen, W. J., De B. J. et al. (1987) *Infect. Immun.*, **55**, 1466-1475.
30. Radford, A. J., Wood, P. R., Billman-Jacobe, H., Geysen, H. M., Mason, T. J. and Tribbick, G. (1990) *J. Gen. Microbiol.*, **136**, 265-272.
31. Sela, S. and Clark-Curtiss, J. E. (1991) *Gene*, **98**, 123-127.
32. Tobias, F., Rinke de Wit T. F., Bekelie, S., Osland, A., Mike, T. L., Hermans, P. W., Dick van Soolingen, Jan-Wouter Drijfhout, Schoningh, R., Janson, A. A. and Thole, J. E. (1992) *Mol. Microbiol.*, **6**, 1995-2007.
33. Caceres, N. E., Harris, N. B., Wellehan, J. F., Feng, Z., Kapur, V. and Barletta, R. G. (1997) *J. Bacteriol.*, **179**, 5046-5055.
34. Madhusudan, K. and Nagaraja, V. (1995) *Microbiology*, **141**, 3029-3037.
35. Papavmasasundaram, K. G., Movahedzadeh, F., Keer, J. T., Stoker, N. G., Colston, M. J. and Davis, E. O. (1997) *Mol. Microbiol.*, **24**, 141-153.
36. Cirillo, J. D., Weisbrod, T. R., Pascopella, L., Bloom, B. R. and Jacobs, W. R. Jr. (1994) *Mol. Microbiol.*, **11**, 629-639.
37. Mahenthiralingam, E., Draper, P., Davis E. O. and Colston, M. J. (1993) *J. Gen. Microbiol.*, **139**, 575-583.
38. Predich, M., Doukhan, L., Nair, G. and Smith, I. (1995) *Mol. Microbiol.*, **15**, 355-366.
39. Dhandayuthapani, S., Zhang, Y., Mudd, M. H. and Deretic, V. (1996) *J. Bacteriol.*, **178**, 3641-3649.

40. Bannantine, J. P., Barletta, R. G., Thoen, C. O., Andrews, R E Jr. (1997) *Microbiology*, **143**, 921-928.
41. Murray, A., Winter, N., Lagranderie, M. et al. (1992) *Mol. Microbiol.*, **6**, 3331-3342.
42. Ramesh, G. and Gopinathan, K. P. (1995) *Indian J. Biochem. Biophys.*, **32**, 361-367.
43. Nesbit, C. E., Levin, M. E., Donnelly-Wu, M. K. and Hatfull, G. F. (1995) *Mol. Microbiol.*, **17**, 1045-1056.
44. Yamaguchi, R., Matsuo, K., Yamazaki, A., Takahashi, M., Fukasawa, Y., Wada, M. and Abe, C. (1992) *Infect. Immun.*, **60**, 1210-1216.
45. Beggs, M. J., Crawford, J. T. and. Eisenach, K. D. (1995) *J. Bacteriol.*, **177**, 4836-4840.
46. Nishi T. and Itoh, S. (1986) *Gene*, **44**, 29-36.
47. Hawley, D. K. and McClure W. R. (1983) *Nucl. Acids Res.*, **11**, 2237-2255.
48. Fredrick, K. and Helmann, J. D. (1997) *Proc. Natl. Acad. Sci., USA*, **94**, 4982-4987.
49. Jeong, w. and Kang, C. (1994) *Nucl. Acids Res.*, **22**, 4667-4672.
50. Andersson, G. E., Sharp, P. M. (1996) *Microbiology*, **142**, 915-925.
51. Gomez, J. E., Chen, J-M and Bishai, W. R. (1997) *Tuber. Lung Dis.*, **78**, 175-183.
52. Dellagostin, O. A., Esposito, G., Eales, L. J., Dale, J. W. and McFadden, J. (1995) *Microbiology*, **141**, 1785-1792.
53. Timm, J., Perilli, M. G., Duez, C., et al. (1994) *Mol. Microbiol.*, **12**, 491-504.
54. Hoopes, C. B. and McClure, W. R. (1987) In *Escherichia coli* and *Salmonella typhimurium* (ed.) F C Neidhart (Washington DC: American Society of Microbiology) pp 1231-1239.
55. Fassler, J. S. and Gussin, G. N. (1996) *Methods Enzymol.*, **273**, 3-42.
56. Mohahedzadeh, F., Colston, M. J. and Davis, E. O. (1997) *J. Bacteriol.*, **179**, 3509-3518.
57. Gross, c. A., Loneto, M. and Losick, R. (1992) In McKnight, s. L. and Yamamoto, K. R. (eds), Transcriptional Regulation. Cold Spring Harbor Laboratory press, cold Spring Harbor, NY, pp. 129-176.
58. Barrios, H., Valderrama, B., Morett, E. (1999) *Nucl. Acids Res.*, **27**, 4305-4313.
59. Gralla, J. D. (1990) *Methods Enzymol*, **185**, 37-54.

60. Morett, e. and Segovia, L. (1993) *J. Bacteriol.*, **175**, 6067-6074.
61. Cole, S.T., Brosch, R., Parkhill, J., et al (1998) *Nature*, **393**, 537-544.
62. De Haseth, P.L. and Helmann, J.D. (1995) *Mol. Microbiol.*, **16**, 817-824.
63. Shinnick, T. M. (1987) *J. Bacteriol.*, **169**, 1080-1088.
64. Thole, J. E., Dauwerse, H. G., Das, P. K., Groothuis, D. G., Schouls, L. M. and van, E. J. D. (1985) *Infect. Immun.*, **50**, 800-806.
65. Mehra, V., Sweetser, D. and Young, R. A. (1986) *Proc. Natl. Acad. Sci., USA.* **83**, 7013-7017.
66. Collins, M. E., Moss, M. T., Wall, S., and Dale, J. W. (1987) *FEMS Microbiol. Lett.*, **43**, 53-56.
67. Andersen, A. B. and Hansen, E. B. (1989) *Infect. Immun.*, **57**, 2481-2488
68. Janssen, G. R. and Bibb, M. J. (1990) *Mol. Gen. Genet.*, **221**, 339-346.
69. Burns, H. and Minchin, S. (1994) *Nucl. Acids Res.*, **22**, 3840-3845.
70. Graves, M. C. and Rabinowitz, J. C. (1986) *J. Biol. Chem.*, **261**, 11409-11415.
71. Helman, J. (1995) *Nucl. Acids Res.*, **23**, 2351-2360.
72. Sabelnikov, A. G., Greenberg, B. and Lacks, S. A. (1995) *J. Mol. Biol.*, **250**, 144-155.
73. Kumar, A., Malloch, R. A., Fujita, N., Smillie, D. A., Ishihama, A. and Hayward, R. S. (1993) *J. Mol. Biol.*, **232**, 406-418.
74. Bercovier, H., Kafri, O. and Sela, S. (1986) *Biochem. Biophys. Res. Commun.*, **136**, 1136-1141.
75. Domenech, P., Menendez, M. C. and Gracia, M. J. (1994) *FEMS Microbiol. Lett.*, **116**, 19-21.
76. Pruss, G.J. and Drlica, K. (1989) *Cell*, **56**, 521-523.
77. Menzel, R. and Gellert, M. (1983) *Cell*, **34**, 105-113.
78. Menzel, R. and Gellert, M. (1987) *Proc. Natl. Acad. Sci., USA*, **84**, 4185-4189.
79. Jovanovich, S.B. and Lebowitz, J. (1987) *J. Bacteriol.*, **169**, 4431-4435.
80. Tse-Dinh, Y.-C. and Beran, R.K. (1988) *J. Mol. Biol.*, **202**, 735-742.
81. Higgins, C.F., Dorman, C.J., Stirling, D.A., Waddell, L., Booth, I.R., May, G. and Bremer, E. (1988) *Cell*, **52**, 569-584.
82. Richardson, S.M.H., Higgins, C.F. and Lilley, D.M.J. (1988) *EMBO J.*, **7**, 1863-1869.
83. Wang, J.C. (1985) *Annu. Rev. Biochem.*, **54**, 665-697.
84. Maxwell, A. and Gellert, M. (1986) *Adv. Protein Chem.*, **38**, 69-107.

85. Dorman, C.J., Ni Bhrian, N. and Higgins, C.F. (1990) *Nature*, **344**, 789-792.
86. Hulton, C.S.J., Seirafi, A., Hinton, J.C.D., Sidebotham, J.M., Waddell, L., Pavitt, G.D., Owen-Hughes, T., Spassky, A., Buc, H. and Higgins, C.F. (1990) *Cell*, **63**, 631-642.
87. Schmid, M.B.(1990) *Cell*, **63**, 451-453.
88. Levin, M. E. and Hatfull, G. F. (1993) *Mol. Microbiol.*, **8**, 277-285.
89. Stover, C. K., de la Cruz, V. F., Fuerst, T. R. et al. (1991) *Nature*, **351**, 456-460.
90. Galan, J. E. and Curtiss, R. 3rd. (1990) *Infect. Immun.*, **58**, 1879-1885.
91. Unniraman, S. and Nagaraja, V. (1999) *Genes Cells*, **12**, 697-706.
92. Shpigelman, E. S., Trifonov, E. N. and Bolshoy, A. (1993) *Comp. App. Biosci.*, **9**, 435-440.
93. Bolshoy, A., McNamara, P., Harrington, R. E. and Trifonov, E. N. (1991) *Proc. Natl. Acad. Sci., USA*, **88**, 2312-2316.
94. Trifonov, E.N. and Ulanovsky, L.E. (1987) In Wells, R.D. and Harvey, S.C. (eds), Unusual DNA Strustures. Springer-Verlag, Berlin, pp. 173-187.

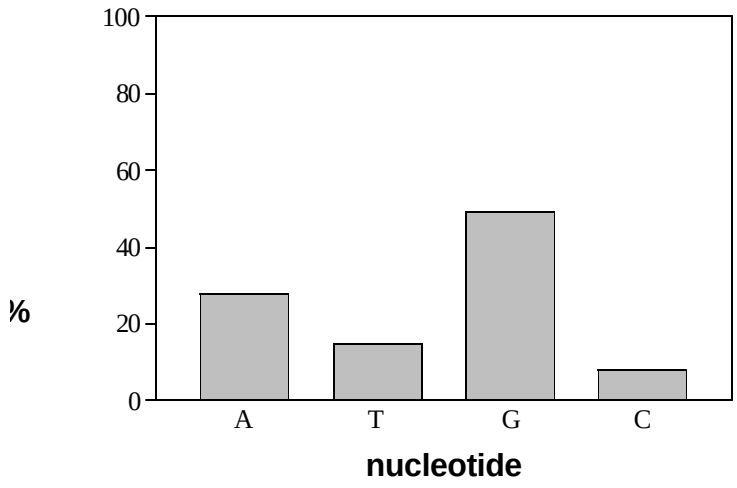


Figure 6-1: Nucleotide preference at transcription start site (tss) for mycobacterial promoter compilation.

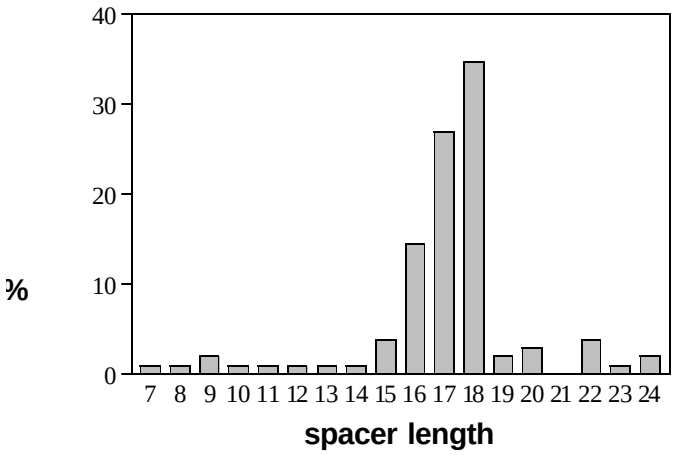


Figure 6-2: Spacer length variation in mycobacterial promoters

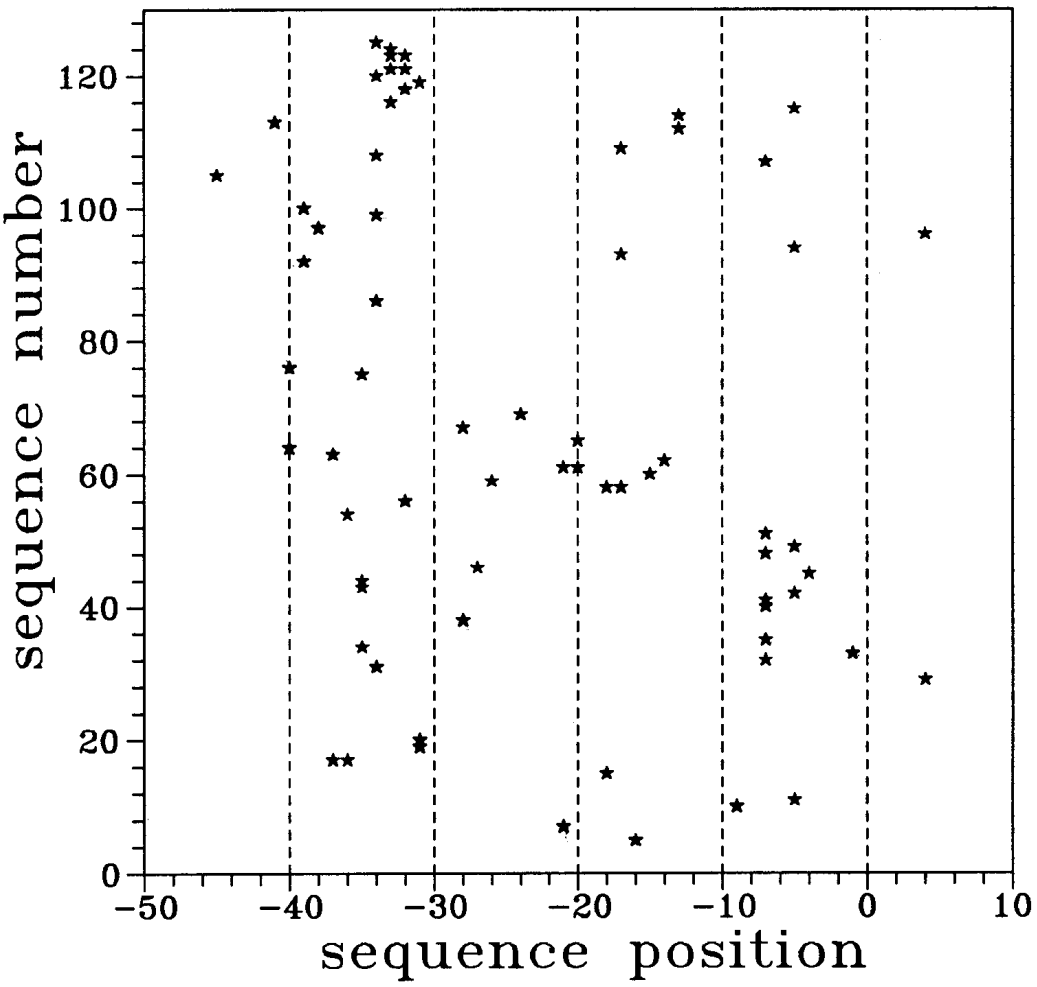


Figure 6-3: Position of curvature maxima of the mycobacterial promoter sequences according, to their sequence number in the compilation list. Mycobacterial promoter sequences having curvature maxima greater than or equal to 0.3 DNA curvature units are considered as curved sequences. (One DNA curvature unit [94] corresponds to the mean DNA in the crystalline nucleosome [1/42.8 $\mathring{A}$]).

Table I: Compilation of Mycobacterial promoters *

		Ref.
	M. tuberculosis	
T3	ATCGACGGCCACGGCTGGTCTAGGACGAGGTACCCGG(**TAACAT**)GCTGGGC[G]	**[6]**
T6	CCGTCCAGTCTGGCAGGCCGGAAACATCGGTCAGCAGA(**TAGGCT**)TTACCA[G]	**[6]**
T26	CTGCGAGCATCATATGCCGCGTGCGTGGTGATGCGGCAG(**GATGTT**)GGACC[A]	**[6]**
T180	GATCACTCCGAGCATGCGCCCATTGTTGTGCATAGGG(**CAGGAT**)GCCCTG[G]	**[6]**
T101	AGCGATCGCAGCCGACGTGATACCTGACCGTTGT*TGA*(**TAGTGT**)CGGCGGC[A]	**[6]**
T119	CCCCGTGCTCGTAGTAGGCGTCCAGCCGACCCGCCGC(**TACCAT**)GCACAAG[T]	**[6]**
T125	CCGAGGTAAGGACTGAGCATGGGCCCGATAAAGTGAC(**TATTAT**)GGATTTC[T]	**[6]**
T129	ACTCGCGGCAGATTACGCCGACGGTTCCTGGCG*TGG*(**TTCAAT**)ATTCGCCG[A]	**[6]**
T130	ACTCCAACAGGTCGATAACCTCCTGCGCCTGCTCGTC(**TATGCT**)GCGATCC[G]	**[6]**
T150	GACCCCCGCCACGTATTGACACTTTGCGACACGCTTT(**TATCAT**)TTTCCGA[C]	**[6]**
recA	TTCGGAGCAGCCGAC(**TTGTCA**)GTGGCTGTC(**TCTAGT**)GTCACGGCC[A]ACCGACCGAT	**[7]**

* Consensus regions are shown in bold letters and transcription start sites are shown in a square bracket.

rrnA P1	GAGAACCTGGTGAGT(**CTCGGT**)GCCGAGATCGAACGGG(**TATGCT**)GTTAGGC[G]ACGGTCACCT	**[8]**
gyrA	GATGGGCGAGGACGT(**CGACGC**)GCGGCGCAGCTTTATCA(**CCCGCA**)ACGCCAA[G]GATGTTCGGT	**[9]**
cpn60	CCCCGGCGATCCCCG(**TGCTCA**)CCACGGGTGATTTCCGG(**GGCGGC**)ATGCGTT[A]GCGGACTAGC	**[10]**
gyrB P1	GATGTCCGACGCACG(**GCGCGG**)TTAGATGGGTAAAAACG(**AGGCCA**)GAAGATC[G]GCCCTGGCGC	**[9]**
gyrB P3	CAAGGGGCCTCGCCA(**TATTGC**)CGGTAGGGGTCCGCGCG(**ACACCT**)ACGGATA[A]CACGTCGATC	**[9]**
85A	GAAGTTGTGGTTGAC(**TACACG**)AGCACTGCCGGGCCCAG(**CGCCTG**)CAGTCTG[A]CCTAATTCAG	**[11]**
85A	CGCCCGAAGTTGTGG(**TTGACT**)ACACGAGCACTGCCGGGCCCAG(**CGCCTG**)CAGTCTG[A]CCTAATTCAG	**[11]**
gyrB P2	AGCGGTTGGCAACGA(**TGTGGT**)GCGATCGCTAAAGATCAC(**CGGGCC**)GGCACC[A]TCGTGGCGCA	**[9]**
rrnA PCL1	TGACCGAACCTGGTC(**TTGACT**)CCATTGCCGGATTTGTAT(**TAGACT**)GGCAGG[G]TTGCCCGAAA	**[12]**
16S rRNA	TGACCGAACCTGGTC(**TTGACT**)CCATTGCCGGATTTGTAT(**TAGACT**)GGCAGG[G]TTGCCCCGAA	**[13]**
glnA	TCGGCATGCCACCGG(**TTACGA**)TCTTGCCGACCATGGCCC(**CACAAT**)AGGGCCGGGG[A]GACCCGGCGT	**[14]**
glnA	CCACCGGTTACGATC(**TTGCCG**)ACCATGGCCC(**CACAAT**)AGGGCCGGGG[A]GACCCGGCGT	**[14]**
katG P_A	GGTCATCTACTGGGG(**TCTATG**)TCCTGATTGTTCGATATCC(**GACACT**)TCGCGATC[A]CATCCGTGAT	**[15]**
katG P_A	ATCTACTGGGGTCTA(**TGTCCT**)GATTGTTCGATATCC(**GACACT**)TCGCGATC[A]CATCCGTGAT	**[15]**
katG P_B	GAGGCGGAGGTCATC(**TACTGG**)GGTCTATGTCCTGATTGTTC(**GATATC**)CGACAC[T]TCGCGATCAC	**[15]**
katG P_B	ACGAGGCGGAGGTCA(**TCTACT**)GGGGTCTATGTCCTGATTGTTC(**GATATC**)CGACAC[T]TCGCGATCAC	**[15]**
katG P_C	CCTGATTGTTCGATA(**TCCGAC**)ACTTCGCGATCACATCCGTGAT(**CACAGC**)CCGATAA[C]ACCAACTCCT	**[15]**

katG P_C	TTCGATATCCGACAC(**TTCGCG**)ATCACATCCGTGAT(**CACAGC**)CCGATAA[C]ACCAACTCCT	**[15]**
purL	CGGCTTGTCCGTTTC(**CACGCG**)GCCGCAGCGCGATGGGGCCTAGC(**TAGACT**)GCCTCC[G]TGATGTCTCC	**[16]**
purC	ATCTCATACCAGAGA(**TACCAG**)CACAGGGCGCCGTCGTGCGGCGGA(**TAGGCT**)GGCGTG[A]TGCGCCCCGC	**[16]**
groE	CAGGAAGCAAGGGGGCG(**CCCTTG**)AGTGCTAGCACTCTCA *TGT*(**ATAGAG**)TGCTAGATGGCAATCGGCTA	**[17]**
groE	CAGGAAGCAAGGGGG(**CGCCCTTG**)AGTGCTAGCAC(**TCTCATGTATAGAG**)TGCTAGATGGCAATCGGCTA	**[17]**
ahpC	TGTGATATATCACCT(**TTGCCT**)GACAGCGACTTCACGG(**TACGAT**)GGAATGTCGTAACCAAATGC	**[18]**
32 kDa	ACATGCATGGATGCG(**TTGAGA**)TGAGGATGAGGGAAGC(**AAGAAT**)GCAGCTTGTTGACAGGGTTC	**[19]**
10kDa	AAGCAAGGGGCGCCC(**TTGAGT**)GTCAGCACTCTCATGTA(**TAGAGT**)GCTAGATGGCAATCGGCTAA	**[20]**
10kDa	AAGCAAGGGGCGCCC(**TTGAGT**)GTCAGCACTCTCATG(**TATAGA**)GTGCTAGATGGCAATCGGCT	**[20]**
10kDa	AAGCAAGGGGCGCCC(**TTGAGT**)GTCAGCAC(**TCTCAT**)GTATAGAGTGCTAGATGGCA	**[20]**
65 kDa	GCGTAAGTAGCGGGG(**TTGCCG**)TCACCCGGTGACCCCCG(**TTTCAT**)CCCCGATCCGGAGGAATCAC	**[19]**
mpt64	GAGTCTGGTCAGGCA(**TCGTCG**)TCAGCAGCGCGATGCCC(**TATGTT**)TGTCGTCGACTCAGATATCG	**[21]**
metA	TCCGGCCCCCGCGAT(**TTGGCG**)AGCTTCGTGCGTGTTCGG(**TAGCCT**)GGCATTTACCGACGCGGGGT	**[22]**
rpsL	GCCGCAACGCCCGCT(**TTGACC**)TGCCAGACTGGCGGCGGG(**TATTGT**)GGTTGCTCGTGCCTGGCGGC	**[23]**
38 kDa	CGTCGCCGGACTGTCGGGGGACGTCAAGGACGCCAAGCGCG(**GAAATT**)GAAGAGCACAGAAAGGTATG	**[19]**
ppgk	CGGGCCGCAGTTTAAGGTGAGGGTCATCCACGTCTCGCCGAGGAGATTCGATGACCAGCAC	**[24]**
	M. bovis BCG	

hsp60 P2	CGGTGCGGGGCTTCTTGCACTCGGCATAGGCGAG*TGC*(**TAAGAA**)TAACGTT[G]	**[25]**
rRNA	TGACCGAACCTGGTC(**TTGACT**)CCATTGCCGGATTTG(**TATTAG**)ACTGGCAGGGTTGCCCCGAA	**[26]**
ahpC	TGTGATATATCACCT(**TTGCCT**)GACAGCGACTTCACGG(**TACGAT**)GGAATGTCGCAACCAAATGC	**[18]**
23K	GAGTCTGGTCAGGCA(**TCGTCG**)TCAGCAGCGCGATGCCC(**TATGTT**)TGTCGTCGACTCAGATATCG	**[27]**
mpb64	GAGTCTGGTCAGGCA(**TCGTCG**)TCAGCAGCGCGATGCCC(**TATGTT**)TGTCGTCGACTCAGATATCG	**[19]**
18K	TGGCGTCCGAAACAC(**TTGAGG**)TGCGGCCCAGCAAGGGGC(**TACAGG**)TTTTTTCCTTCACCTACGGA	**[28]**
64K	GCGTAAGTAGCGGGG(**TTGCCG**)TCACCCGGTGACCCCCGG(**TTTCAT**)CCCCGATCCGGAGGAATCAC	**[29]**
rpsL	GCCGCAACGCCCGCT(**TTGACC**)TGCCAGACTGGCGGCGGG(**TATTGT**)GGTTGCTCGTGCCTGGCGGC	**[23]**
mpb70	TGGCGTCCGAAACAC(**TTGAGG**)TGCGGCCCAGCAAGGGGC(**TACAGG**)TTTTTTCCTTCACCTACGGA	**[30]**
alpha	CGACTTTCGCCCGAA(**TCGACA**)TTTGGCCTCCACACACGG(**TATGTT**)CTGGCCCGAGCACACGACGA	**[19]**
	M. leprae	
16S rRNA	TAGTCAACCCGGGAC(**TTGACT**)CCTCTGCTGGATC*TGT*(**ATTAAT**)CTGGCTG[G]GTTGCCGAAG	**[31]**
18 Kda	CTTGTCTATCACAAC(**TTGCAT**)CAATATATCGACCAGTG(**CTATAT**)CAAATCTA[T]GTAGTCAGGA	**[19]**
18 Kda	CTTGTCTATCACAAC(**TTGCAT**)CAATATATCGACCAG*TGC*(**TATATC**)AAATCTA[T]GTAGTCAGGA	**[19]**
28-kDa	TCAATATAACCACTC(**TGGTCA**)CACTAACCATACTCG(**TACCAT**)CAACCGTGTGGGGCTAATCC	**[19]**
groE1	AGCAGCGGGCCGGCC(**TTGAGT**)GCTAGCACTCGCGTGTA(**TAGAGT**)GCTAGATGGCAGTCGGCCAG	**[32]**
65 kd	GAATTCCGGAA(**TTGCAC**)TCGCCTTAGGGGAG*TGC*(**TAAAAA**)TGATCCTGGCACTCGCGATC	**[19]**

36k	GTTGGG(**TTTCCT**)CTCGGAGGGCGCACCGC(**TACGTT**)AGCGGGATG	**[19]**
SOD	GG(**TGGGCG**)CGATCATGGCGCAGCGTT(**GATTAT**)GCTAGTCG	**[19]**
rpsL	CGCCGTTGGGTCGCT(**TTGACC**)TGCCCGAGCAGGGACGGG(**TATTGT**)GTTTCTCGTTCCTGACGGCT	**[23]**
	M. smegmatis	
alrA	GTCTGCGGCCTCTGG(**GACAAT**)GGGCGCC[G]GAGATTATGA	**[33]**
S4	AAGCCGAATCGAGACCTTTTGGGTTCGTACACACTTGCTT(**TATAAG**)CCTC[G]	**[6]**
S5	AACAAGATTCCGTTAATCGTGTCTGGTGGAGCTGG*TGG*(**TAAGCT**)TGATCC[G]	**[6]**
S6	CATCGATTTTAAATTTT*TGA*(**TAGAG**T)GCAAATA[A]	**[25]**
S12	ACCTCGTTATGCTTCTGGCTATTTTTGATCAACTTT(**TATACA**)TGGGCGGT[T]	**[6]**
S14	TCAAGCACCCAAGCCAACATGGT TGTAGTAGTCGTTT(**TACCAT**)GTGTACC[T]	**[6]**
S16	TCCACGCGAACCGCTTCGGCGTGCCCCGTTTTCCC*TGT*(**TATAAT**)ATCGGC[G]	**[6]**
S18	GATCATTGTCTTCTGTTGTCTTTCGTA(**TAAAGT**)TGTTACT[G]	**[6]**
S19	TTTGATGTAGCCAAAGGCTCTCACCACCTGAGCCA*TGA*(**TAGTAT**)CCATCC[C]	**[6]**
S21	ACATGGCATTTTTCATTTAAAACAGGACTCAGGTGG(**TATGGT**)TGACATCG[A]	**[6]**
S30	GATCAGCTATGTTCTTCAGTAAAATTTCGGC(**TATATG**)TTGGT[G]	**[6]**
S33	GATCCGCTCTTCTTATGATGCCAGTTATGGTATC(**TATGGT**)TATC[G]	**[6]**
S35	AACTAAAGTATGTGCCGTAATTGACAGTGTTCTAGAT(**TATGAT**)GCTGCAT[C]	**[6]**

S65	GGCACAGCTCGAAGTTCTACTACATGGCTTGCT GAA(TCCAGT)CACATTAC[T]	**[6]**
S69	ATCACGATGTCTTCATGCTTGGCTTCAATGCTCCGGTC(**TACAAT**)CAGTTC[A]	**[6]**
S119	GATCAAGAAGCCAATGATT*TGT*(**TAAACG**)CAATTAAT[G]	**[6]**
gyrB	CAGAATCGGTGCTGT(**CGCTAT**)CTCGCGG(**TAGACT**)GGACGAC[G]GATCTCAGGC	**[34]**
recA	AGAGTTCGACCGGAC(**TTGTCG**)GTGGTC*TGC* (**TCTAAC**)GTCACGGCC[A]ACCGATCGGA	**[35]**
ask	GT(**TTGCCC**)GCCGCGGCGCCC(**CACGAT**)GAACCGC[A]CGGGCTGACG	**[36]**
acetamidase	GGCCGGCGTTCACCC(**TTGACT**)TTTATTTTCATCTGGA(**TATATT**)TCGGGT[G]AATGGAAAGG	**[37]**
rrnB	CTCTGACCTGGGGAT(**TTGACT**)CCCAGTTTCCAAGGACG(**TAACTT**)ATTCCAG[G]TCAGAGCGAC	**[38]**
rrnA P1	GAAAACCTGGTCAGC(**CTCGGA**)GCCGAGATCGAGAGAG(**TAAGCT**)CGTAG[G]AAGCAAGACC	**[12]**
rrnA P2	CTCTGACCAGGCGAT(**TTGCAA**)TCGCGACGAACCTCGTAT(**TATCTT**)TATGAA[G]TCGCCGCGGA	**[12]**
rrnA P3	CCGGGCCAGAGCGAC(**TTGACA**)AGCCAGCCGAGATCGTAC(**TAAGCT**)GGCGAG[G]TTGCCTCAGA	**[12]**
rrnA PCL1	CCGGTCCAGAGCGAC(**TTGACA**)AGCCAGACAAAGCAGTAT(**TAAGCT**)GGCAGG[G]TTGCCCCAAA	**[12]**
rpsL	CCGCCGTGCACGAGT(**TTGTTT**)CGTCGCGGTCGCCCC*TGG*(**TATTGT**)GGTGGATC[G]TGCCTGGCCC	**[23]**
rpsL	CGTGCACGAGTTTGT(**TTCGTC**)GCGGTCGCCCCTGGTAT(**TGTGGT**)GGATC[G]TGCCTGGCCCGAAA	**[23]**
ahpC	TGTGATATATCACCT(**TTGCCT**)GACAGCGACTTCACGG(**CACGAT**)GGAATGTCGCAACCAAATGC	**[39]**
	M. paratuberculosis	

pAJB303	GACGACGAGGGCGG(**TGGCGT**)CGCCGGTGTAGCCGAA(**CGGCAC**)GTGCGCG[T]AGGCCCAGAT	**[40]**
pAJB86	CCACCTTACTCCCGA(**TGACGT**)TGCACGGCTGGGATTAA(**CGGTCC**)GCGTGC[T]CCAGGAGACA	**[40]**
pAJB125	GCAACGAGCGCATCA(**TTAAAG**)ATCGANGGCGCCGGGNT(**CATGTC**)CCTTCAC[C]CCGCCCAGCT	**[40]**
pAJB300	TCGAGTTCAAGACCC(**TGACGC**)TGGCCGACCTCGGCGCG(**CAGCCG**)ACCGCGC[A]GCGGTGCACG	**[40]**
pJB305	ATCCGGACGGGCAGT(**TGTTGG**)AGTTTCTGTCGGACGGT(**TGGTTG**)GCGGCAT[T]TCCGGCGAGG	**[40]**
pAJB304	CACCAGGTACACGCC(**AAGGAC**)AACGGCCGTATCCGGTA(**CCAACG**)GGTGTGC[G]AGCTGGACGG	**[40]**
P_{AN}	CTGGTGAAGGGTGAA(**TCGACA**)GGTACACACAGCCGCCA(**TACACT**)TCGCTTC[A]TGCCCTTACG	**[41]**
pAJB73	GATCGGTG(**TGCCGC**)TTGAACCGGCCCAGCTCCCG(**CTCCAG**)GGTGACG[T]GCTCGAGCTC	**[40]**
pAJB301	GATCTGGCGGGCCG(**TCCAGT**)ACACCGCGAGTTCGCGCACG(**CTGGCC**)GGCAGCGTCTTGGACGCCCG	**[40]**
	M. fortuitum	
repA	GAGCTCGTGTCGGACCATACACCGGTGATTAATCG*TGG*(**TCTACT**)ACCAAG[C]	**[25]**
rrnA PCL1	CCAGGATGATGCAAC(**TTGACT**)TGCCGGCAAGATTCGAAT(**TAAGCT**)GGCGGG[G]TTGCCCCAAA	**[12]**
rrnA P1	GAAAACCTGTTGAGC(**CTCGGA**)GCCGAGATCGAAAGAG(**TAGGGT**)CGTAAACAGCAGTCCGGGCC	**[12]**
rrnA P2a	CGCTGACCAGCCGAT(**TTGACC**)TTGTAGGCAGGCCCGCGC(**TAATCT**)TTTGAAGTCGCGCGGAGCGG	**[12]**
rrnA P2b	CCGGGCCAGAGCGAC(**TTGACA**)AGCCAGCCGAGATCGTAC(**TAAGCT**)GGCGAGGTTGCCTCAGACCG	**[12]**
rrnA P3	CAGGATGATGCAACT(**TGACTT**)GCCGGCAAGATTCGAATT(**AAGCTG**)GCGGGGTTGCCCCAAAACAG	**[12]**
	M. phlei	

rrnA PCL1	ACTGGGGACGAGGTC(**TTGACG**)CCCCTGATCAGATCGGTA(**TAGACT**)GGCAGG[G]TTGCCCGAAA	**[12]**
rrnA P1	GAGAACCTCCGCAGT(**CTCGGC**)GCCGAGATCGAGAGGG(**TCGCCT**)GAAACATGCCGTTTACCTGC	**[12]**
rrnA P2	AGGGGACCCCCCTTT(**TTGACT**)CCGCTCAGACGTGGGC(**TATTCT**)TCTAACCACAAGCCCAACGC	**[12]**
rrnA P3	CTGGGGACGAGGTCT(**TGACGC**)CCCTGATCAGATCGGTAT(**AGACTG**)GCAGGGTTGCCCGAAAGCAA	**[12]**
	Mycobacteriophage I3	
pKGR25	CCTGTACACCCTCGC(**TGCACT**)CGCCGAGGACAAG(**CACTAT**)CGCCCCGACGTCCCGGCCTGG	**[42]**
pKGR9	ACCACGAGCACCCGG(**TCGTCA**)GGACTGCGACACTCGA(**TGTTGT**)AGACGCACTGGTGCAGCATG	**[42]**
pKGR38	ATCTGGTCGACCTGC(**TCGACG**)AGGTCGATCATCTTCT(**TCATCT**)CGCCGAACGGGATGCCCTGG	**[42]**
ORF1	ACCTCATGGAGCACT(**TCGAGG**)TCACTGAGCACGCCCA(**CGAACT**)ACGAGAGGCCGTGGGACTGG	**[42]**
ORF2	TACTTTTTGTACCGT(**TCGACA**)CCAGCGGTTTCCGCTTCCT*TGC*(**CAATCT**)CCTGCAAACAAACCACAATG	**[42]**
pKGR1	ACACAGACCAGGAGC(**TCGACA**)TGACCGCCACCGCCCCCTACAGCG(**TCATCT**)GGTTCGAAGGCACCCCGGAT	**[42]**
	Mycobacteriophage L5	
71 P2	TACCTGTCACAAGGT(**TTGCTA**)CCGAGTGGGGCAGGCCGC(**TACATT**)TACGACC[G]CGTAACGCCA	**[43]**
71 P_{left}	TTTGCGATTAGGGC(**TTGACA**)GCCACCCGGCCAGTAGTG(**CATTCT**)TGTGTC[A]CCGCAGCAGC	**[43]**
71 P1	ACAACTGAATATGGT(**TCCGCA**)GACGCAACTAAATTAGGGG(**TATCCT**)TGACA[G]GCACCAACAT	**[43]**
	M. avium	

Avi-3	GCCGGCGATCGTGGG(**CTGATA**)AGTCTTATCGGGCATAC(**TATAAG**)TGTAGTGGGAAATATCA CCT	**[44]**
pLR7	AGCCTTGTTGGCGGC(**CAACTG**)CCGGACGATCGCGGCGGC(**CATCGT**)CCTCGAGCTCGGCCCCGTGC	**[45]**
	M. neoaurum	
rrnA PCL1	GCGAGACAGAGAAGC(**TTGACT**)CGCCAGACAAGATAGTT(**TAAGCT**)GGCAGG[G]TTGCCCCGAA	**[12]**
rrnA P1	GAAAACCTGGTCAGC(**TTGGGC**)GCCGGGATCGAGCGAG(**TACACT**)CGTAAGAGA CCGGTCGAGTG	**[12]**
rrnA P3	GCGAGACAGAGAAGC(**TTGACT**)CGCCAGACAAGATAGTT(**TAAGCT**)GGCAGGGTTGCCCCGAAACG	**[12]**
rrnA P2	CTCTGACCAGCGGAT(**TTGACT**)TCCGAAGGCACAAAGTTC(**TAATCT**)TTTGAAGTCGCCGCGGGGAG	**[12]**
	M. abscessus	
rrnA P4	GCCAAAACCGGGAAT(**TTGACT**)CAGGTTCACGAACT *TGA*(**TACGGT**)TTCCGA[G]CGCCCGAAAG	**[12]**
rrnA P1	GGCGGGTCTAGTGGC(**GGACGG**)CGTCACAGAGGTATACGA(**TGTGTT**)TCATATCG[A]CCGCGGTTAC	**[12]**
rrnA PCL1	GCCCCCGACCCGAAG(**TTGACT**)CAAGTTCATTGGACT *TGG*(**TACAGT**)GGTCGG[G]TTGCCCTGAA	**[12]**
rrnA P2	GCCAAAACCGGGAAT(**TTGACT**)CAAGTTCACCGAACT*TGA*(**TACGGT**)TTCC[A]AGTCGCTCGG	**[12]**
rrnA P3	GCCAAAACCGGGAAT(**TTGACT**)CAAGTTCACCGAACT*TGA*(**TACGGT**)TTCCAA[G]TCGCTCGGAA	**[12]**
	M. chelonae	
rrnA P2	CCAAAACCCGGAGTT(**TGACTC**)AAGTTCACCGAACT *TGA*(**TCGGTT**)CCCGG[G]CCGCTTACAA	**[12]**
rrnA P1	GGCGGGGTTAGTGGC(**GGATGG**)CGTCACCGAGGTATACGA(**TGTGTT**)TCATATC[G]ACCGCGGTTA	**[12]**
rrnA PCL1	CCCCAGAACCCGAAG(**TTGACT**)CAAGTTCATTGGACT*TGG*(**TACAGT**)GGTCGG[G]TTGCCCTGAA	**[12]**

rrnA P3	GCCAAAACCGGGAAT(**TTGACT**)CAAGTTCACCGAACT*TGA*(**TCGGTT**)TCCCA[G]CCGCCCGAAA	**[12]**
rrnA P4	GCCAAAACCGGGAAT(**TTGACT**)CAAGTTCACCGAACT*TGA*(**TACGGT**)TTCCGA[G]CCGCCCGAAA	**[12]**

Table II: Analysis of different features of each mycobacterial promoter

Gene	bp	-35	spacer	TG	-10	length	TSS	% (A+T)	% (G+C)
	M. tuberculosis								
T3	51	-	-	-	TAACAT	7	G	35.3	64.7
T6	51	-	-	-	TAGGCT	6	G	41.18	58.82
T26	51	-	-	-	GATGTT	5	A	39.22	60.78
T80	50	-	-	=	CAGGAT	6	G	40.0	60.0
T101	51	-	-	TG	TAGTGT	7	A	41.18	58.82
T119	51	-	-	-	TACCAT	7	T	33.34	66.67
T125	51	-	-	-	TATTAT	7	T	52.94	47.06
T129	51	-	-	TG	TTCAAT	8	A	41.18	58.82
T130	51	-	-	-	TATGCT	7	G	41.18	58.82
T150	51	-	-	-	TATCAT	7	C	49.02	50.98
recA	56	TTGTCA	9	-	TCTAGT	9	A	41.07	58.93
rrnA P1	61	CTCGGT	16	-	TATGCT	7	G	40.98	59.02
gyrA	62	CGACGC	17	-	CCCGCA	7	G	35.48	64.51
cpn60	62	TGCTCA	17	-	GGCGGC	7	A	30.64	69.35
gyrB P1	62	GCGCGG	17	-	AGGCCA	7	G	37.09	62.91
gyrB P3	62	TATTGC	17	-	ACACCT	7	A	37.10	62.91
85A	62	TACACG	17	-	CGCCTG	7	A	41.94	58.06
	67	TTGACT	22	-	CGCCTG	7	A	38.8	61.2
gyrB P2	62	TGTGGT	18	-	CGGGCC	6	A	37.1	62.9
rrnA PCL1	62	TTGACT	18	-	TAGACT	6	G	48.39	51.61
16S rRNA	62	TTGACT	18	-	TAGACT	6	G	46.77	53.23
glnA	66	TTACGA	18	-	CACAAT	10	A	33.33	66.67
	58	TTGCCG	10	-	CACAAT	10	A	32.76	67.24
katG P_A	65	TCTATG	19	-	GACACT	8	A	50.77	49.23

	61	TGTCCT	15	-	GACACT	8	A	52.46	47.54
katG P $_B$	64	TACTGG	20	-	GATATC	6	T	46.87	53.12
	66	TCTACT	22	-	GATATC	6	T	46.97	53.03
katG P $_C$	67	TCCGAC	22	-	CACAGC	7	C	49.25	50.75
	59	TTCGCG	14	-	CACAGC	7	C	49.15	50.85
purL	67	CACGCG	23		TAGACT	6	G	32.84	67.16
purC	68	TACCAG	24		TAGGCT	6	A	33.83	66.17
groE	68	CCCTTG	19	TG	ATAGAG	-	-	45.59	54.41
groE	68	CGCCCTTG	11		TCTCATGT -ATAGAG	-	-	45.59	54.41
ahpC	63	TTGCCT	16	-	TACGAT	-	-	53.96	46.03
32 KDa	63	TTGAGA	16	-	AAGAAT	-	-	50.79	49.21
10 KDa	64	TTGAGT	17	-	TAGAGT	-	-	48.44	51.57
	62	TTGAGT	15	-	TATAGA	-	-	46.77	53.23
	55	TTGAGT	8	-	TCTCAT	-	-	47.27	52.73
65 KDa	64	TTGCCG	17	-	TTTCAT	-	-	35.94	64.07
mpt 64	64	TCGTCG	17	-	TATGTT	-	-	43.75	56.25
metA	65	TTGGCG	18	-	TAGCCT	-	-	33.85	66.15
rpsL	65	TTGACC	18	-	TATTGT	-	-	30.77	69.23
38 KDa	67	-	-	-	GAAATT	-	-	40.3	59.7
ppgK	61	-	-	-	-	-	-	39.34	60.66
M. bovis BCG									
hsp60 P2	51	-	-	TG	TAAGAA	7	G	43.14	58.86
rRNA	62	TTGACT	15	-	TATTAG	-	-	46.77	53.23
ahpC	63	TTGCCT	16	-	TACGAT	-	-	52.38	47.62
23 K	64	TCGTCG	17	-	TATGTT	-	-	43.75	56.25
mpb 64	64	TCGTCG	17	-	TATGTT	-	-	43.75	56.25
18 K	65	TTGAGG	18	-	TACAGG	-	-	43.08	56.92
64 K	65	TTGCCG	18	-	TTTCAT	-	-	35.08	64.62

rpsL	65	TTGACC	18	-	TATTGT	-	-	30.77	69.23
mpb70	65	TTGAGG	18	-	TACAGG	-	-	43.08	56.92
alpha	65	TCGACA	18	-	TATGTT	-	-	41.54	58.46
	M. leprae								
16S rRNA	61	TTGACT	16	TG	ATTAAT	7	G	47.54	52.46
18 KDa	63	TTGCAT	17	-	CTATAT	8	T	63.5	36.51
	63	TTGCAT	18	TG	TATATC	7	T	63.5	36.51
28 KDa	62	TGGTCA	15	-	TACCAT	-	-	53.22	46.78
groE1	64	TTGAGT	17	-	TAGAGT	-	-	37.5	62.5
65 KD	60	TTGCAC	17	TG	TAAAAA	-	-	48.33	51.66
36 K	44	TTTCCT	17	-	TACGTT	-	-	36.36	63.64
SOD	40	TGGGCG	18	-	GATTAT	-	-	40.0	60.0
rpsL	65	TTGACC	18	-	TATTGT	-	-	38.46	61.54
	M. smegmatis								
alrA	39	-	-	-	GACAAT	7	G	38.46	61.54
S4	51	-	-	-	TATAAG	4	G	52.94	47.06
S5	51	-	-	TG	TAAGCT	6	G	50.98	49.02
S6	34	-	-	TG	TAGAGT	7	A	76.48	25.53
S12	51	-	-	-	TATACA	8	T	60.79	39.22
S14	51	-	-	-	TACCAT	7	T	54.90	45.10
S16	51	-	-	TG	TATAAT	6	G	41.18	58.82
S18	41	-	-	-	TAAAGT	7	G	65.85	34.14
S19	51	-	-	TG	TAGTAT	6	C	50.98	49.02
S21	51	-	-	TG	TATGGT	8	A	60.78	39.22
S30	43	-	-	-	TATATG	5	G	62.79	37.21
S33	45	-	-	-	TATGGT	4	G	57.78	42.22
S35	51	-	-	-	TATGAT	7	C	62.74	37.26
S65	51	-	-	-	TCCAGT	8	T	52.94	47.06

S69	51	-	-	-	TACAAT	6	A	54.9	45.1
S119	37	-	-	TG	TAAACG	8	G	67.57	32.43
gyr B	52	CGCTAT	7	-	TAGACT	7	G	40.38	59.61
recA	56	TTGTCG	9	TG	TCTAAC	9	A	41.07	58.93
ask	44	TTGCCC	12	-	CACGAT	7	A	25.0	75.0
acetamidase	60	TTGACT	16	-	TATATT	6	G	55.0	45.0
rrn B	62	TTGACT	17	-	TAACTT	7	G	48.39	51.61
rrnA P1	59	CTCGGA	18	-	TAAGCT	5	G	44.26	55.74
rrnA P2	62	TTGCAA	18	-	TATCTT	6	G	48.39	51.61
rrnA P3	62	TTGA CA	18	-	TAAGCT	6	G	38.71	61.29
rrnA PCL1	62	TTGACA	18	-	TAAGCT	6	G	45.16	54.84
rpsL	64	TTGTTT	18	TG	TATTGT	8	G	34.37	65.63
	64	TTCGTC	17	-	TGTGGT	5	G	39.06	65.63
ahpC	63	TTGCCT	16	-	CACGAT	-	-	50.79	49.2
M. paratuberculosis									
pAJB303	60	TGGCGT	16	-	CGGCAC	7	T	28.34	71.66
pAJB86	61	TGACGT	17	-	CGGTCC	6	T	39.34	60.66
pAJB125	62	TTAAAG	17	-	CATGTC	7	C	37.1	59.67
pAJB300	62	TGACGC	17	-	CAGCCG	7	A	27.42	72.58
pAJB305	62	TGTTGG	17	-	TGGTTG	7	T	38.71	61.29
pAJB304	62	AAGGAC	17	-	CCAACG	7	G	35.48	64.52
P_{AN}	62	TCGACA	17	-	TACACT	7	A	45.16	54.84
pAJB73	58	TGCCGC	20	-	CTCCAG	7	T	31.04	68.97
pAJB301	66	TCCAGT	20	-	CTGGCC	-	-	27.27	72.73
M. fortuitum									
repA	51	-	-	TG	TCTACT	6	C	47.06	52.94
rrnA PCL1	62	TTGACT	18	-	TAAGCT	6	G	46.78	53.22
rrnA P1	65	CTCGGA	16	-	TAGGGT	-	-	43.07	56.93

rrnA P2a	65	TTGACC	18	-	TAATCT	-	-	38.46	61.64
rrnA P2b	65	TTGACA	18	-	TAAGCT	-	-	36.93	63.08
rrnA P3	65	TGACTT	18	-	AAGCTG	-	-	47.96	52.31
M. phlei									
rrnA PCL1	62	TTGACG	18	-	TAGACT	6	G	41.93	58.06
rrnA P1	65	CTCGGC	16	-	TCGCCT	-	-	38.46	61.54
rrnA P2	63	TTGACT	16	-	TATTCT	-	-	42.85	57.14
rrnA P3	65	TGACGC	18	-	AGACTG			41.54	58.47
Mycobacteriophage I3									
pKGR25	61	TGCACT	13	-	CACTAT	-	-	31.14	68.85
pKGR9	63	TCGTCA	16	-	TGTTGT	-	-	39.68	60.32
pKGR38	63	TCGACG	16	-	TCATCT	-	-	41.27	58.73
ORF1	63	TCGAGG	16	-	CGAACT	-	-	38.1	61.91
ORF2	69	TCGACA	22	TG	CAATCT	-	-	52.18	47.82
pKGR1	71	TCGACA	24	-	TCATCT	-	-	36.62	63.38
Mycobacteriophage L5									
71 P2	63	TTGCTA	18	-	TACATT	7	G	42.85	57.14
71 P_{left}	61	TTGACA	18	-	CATTCT	6	A	42.62	57.38
71 P1	62	TCCGCA	19	-	TATCCT	5	G	54.84	45.16
M. avium									
avi-3	64	CTGATA	17	-	TATAAG	-	-	51.56	48.44
PLR7	65	CAACTG	18	-	CATCGT	-	-	27.69	72.31
M. neoaurum									
rrnA PCL1	61	TTGACT	17	-	TAAGCT	6	G	45.9	54.1
rrnA P1	65	TTGGGC	16	-	TACACT	-	-	40.0	60.0
rrnA P3	64	TTGACT	17	-	TAAGCT	-	-	45.31	54.69
rrnA P2	65	TTGACT	18	-	TAATCT	-	-	47.70	52.31
M. abscessus									

rrnA P4	61	TTGACT	17	TG	TACGGT	6	G	47.54	52.46
rrnA P1	64	GGACGG	18	-	TGTGTT	8	A	42.19	57.82
rrnA PCL1	62	TTGACT	18	TG	TACAGT	6	G	43.54	56.45
rrnA P2	60	TTGACT	18	TG	TACGGT	4	A	50.0	50.0
rrnA P3	62	TTGACT	18	TG	TACGGT	6	G	51.61	48.39
	M. chelonae								
rrnA P2	60	TGACTC	17	TG	TCGGTT	5	G	46.67	53.34
rrnA P1	63	GGATGG	18	-	TGTGTT	7	G	42.86	57.15
rrnA PCL1	62	TTGACT	18	TG	TACAGT	6	G	46.77	53.23
rrnA P3	61	TTGACT	18	TG	TCGGTT	5	G	47.54	52.46
rrnA P4	62	TTGACT	18	TG	TACGGT	6	G	48.38	51.61

Table III: Percentage conserved homology of –35 and –10 regions for different mycobacterial species

	% (A+T)	% (G+C)	-35 region and % conserved homology						-10 region and % conserved homology					
M. tuberculosis	41.72	58.27	T 84.	T 52.	G 48.	A 36.	C 52.	G 39.	T 52.	A 71.	Y 31.	A 40.	C 33.	T 71.
M. bovis BCG	42.29	57.71	T 100.	T 67.	G 100.	A 56.	C 78.	G 56.	T 100.	A 90.	T 60.	G 50.	A 40.	T 60.
M. leprae	47.94	52.07	T 100	T 78.	G 89.	C 44.	C 56.	T 56.	T 67.	A 78.	T 44.	A 44.	A 56.	T 78.
M. smegmatis	49.90	50.11	T 83.	T 92.	G 75.	A 33.	C 58.	T 42.	T 89.	A 89.	T 44.	A 46.	A 37.	T 82.
M. paratuberculosis	34.42	65.22	T 89.	T 56.	R 33.	C 44.	G 67.	Y 33.	C 78.	R 33.	G 56.	C 33.	C 56.	C/G 44.
M. fortuitum	43.16	56.83	T 80.	T 80.	G 60.	A 60.	C 60.	T/A 40.	T 83.	A 83.	A 50.	G 50.	C 67.	T 83.
M. phlei	40.73	58.82	T 75.	T 75.	G 50.	A 50.	C/G 50.	C 50.	T 75.	A 50.	G 50.	C 50.	C 75.	T 75.
Mycobacteriophage I3	40.0	60.0	T 100.	C 83.	G 83.	A 83.	C 83.	A 50.	T/A 50.	A/C /G 33.	A 67.	T 83.	C 67.	T 100.
Mycobacteriophage L5	46.77	53.23	T 100.	T 67.	G 67.	A/C /G 33.	C 67.	A 100.	T 67.	A 100.	T 67.	T/A/ C 33.	C 67.	T 100.
M. avium	39.54	60.46	C 100.	T/A 50.	R 50.	A/C 50.	T 100.	R 50.	Y 50.	A 100.	T 100.	A/C 50.	R 50.	T/G 50.
M. neoaurum	44.70	55.19	T 100.	T 100.	G 100.	A 75.	C 75.	T 75.	T 100.	A 100.	A 75.	G 50.	C 100.	T 100.
M. abscessus	46.92	53.08	T 80.	T 80.	G 80.	A 80.	C 80.	T 80.	T 100.	A 80.	C 80.	G 80.	G 80.	T 100.
M. chelonae	46.43	53.57	T 80.	T 60.	G 60.	A 60.	C 60.	T 60.	T 100.	A/C 40.	G/C 40.	G 80.	T 60.	T 100.
Overall	43.70	56.28	T 87.	T 60.	G 65.	A 46.	C 56.	T 39.	T 70.	A 74.	T 34.	A 35. G 33.	C 33. A 27.	T 74.
E. coli	59.54	40.46	T 82.	T 84.	G 78.	A 65.	C 54.	A 45.	T 80.	A 95.	T 45.	A 60.	A 50.	T 96.

Table IV: Position of curvature maxima lying between region –50 and +10 for curved mycobacterial promoters

–50 to -41	–40 to -31	–30 to -21	–20 to –11	-10 to –1	+1 to +10
-	*M. tuberculosis* 85A, rrnA PCL1, 16S rRNA, 65 kDa, rpsL, 38 kDa	*M. tuberculosis* T125	*M. tuberculosis* T101, gyrB P1	*M. tuberculosis* T150, mpt64, metA	*M. tuberculosis* 32 kDa
-	*M. bovis BCG* 64 K, rpsL	*M. bovis BCG* rRNA, alpha	-	*M. bovis BCG* 23 K, mpb64, 18 K, mpb70	-
-	*M. leprae* rpsL	-	-	*M. leprae* 18 KDa, 28Kda, 65 Kda	-
-	*M. smegmatis* S4, S19, S21, rrnB, rrnA P1	*M. smegmatis* S12, S14, S16, S35, S69	*M. smegmatis* S30, S18, S6	-	-
-	*M. paratuberculosis* pAJB 305	-	-	-	-
-	*M. fortuitum* rrnA PCL1, rrnA P3	-	*M. fortuitum* rrnA P1	*M. fortuitum* rrnA P2a	-
-	*M. phlei* rrnA PCL1, rrnA P2, rrnA P3	-	-	-	-

Table IV continued:

–50 to –41	–40 to -31	–30 to -21	–20 to -11	-10 to –1	+1 to +10
Mycobacteriophage I3 ORF2	-	-	-	-	-
-	*Mycobacteriophage L5* 71 P_{left}	-	*Mycobacteriophage L5* 71 P1	*Mycobacteriophage L5* 71 P2	-
M. neoaurum rrnA P1	-	-	*M. neoaurum* rrnA PCL1, rrnA P3	*M. neoaurum* rrnA P2	-
-	*M. abscessus* rrnA P4, rrnA PCL1, rrnA P2, rrnA P3	-	-	-	-
-	*M. chelonae* rrnA P2, rrnA PCL1, rrnA P3, rrnA P4	-	-	-	-

CHAPTER

7

Application of artificial neural networks for prediction of mycobacterial promoter sequences

A multilayered feed-forward artificial neural network (ANN) architecture trained using the error-back-propagation (EBP) algorithm has been developed for predicting whether a given nucleotide sequence is a mycobacterial promoter sequence. Owing to the excellent prediction capability (≅97%) of the developed network model, it has been further used in conjunction with the calliper randomization (CR) approach for determining the structurally/functionally important regions in the promoter sequences. The results obtained thereby indicate that: (i) upstream region of –35 box, (ii) –35 region, (iii) spacer region and, (iv) -10 box, are important for mycobacterial promoters. The CR approach also suggests that the –38 to –29 region plays a significant role in determining whether a given sequence is a mycobacterial promoter. In essence, the present study establishes ANNs as a tool for predicting mycobacterial promoter sequences and determining structurally/functionally important sub-regions therein.

7.1 INTRODUCTION

Mycobacteria while have a low transcription rate and a low RNA content per unit DNA [1], their genomes are rich in the G+C content. Since the G+C content of a genome affects the codon usage and the promoter recognition sites in an organism [2-3], it is expected that the transcription and translation signals in *Mycobacteria* may be different from those in other bacteria such as, *E. coli*. Understanding the factors responsible for the low level of transcription and the possible mechanisms of regulation of gene expression in *Mycobacteria* necessitates examination of the structure of mycobacterial promoters and their transcription machinery.

Mulder et al. [4], have listed –35 and –10 regions of a few mycobacterial promoters. Some promoters from their compilation contain –35 and –10 regions resembling *E. coli* σ^{70} type promoters. Although *Mycobacteriophage I3* [5] and *M. paratuberculosis* [6] promoters exhibit good sequence similarity with the *E. coli* promoters at the –35 consensus, they display significant variation in the –10 region. For promoters like *M. tuberculosis 85A* [7], sequences at the –35 position are essential for transcription although their exact location may not be critical. Some mycobacterial promoters, for instance, *M. paratuberculosis* [6], have a high GC content in their –10 region as compared to the AT rich –10 region of *E. coli* σ^{70} type. Possibly, promoters having a high GC content at –10 region are the true representatives of the mycobacterial type. An analysis of *M. smegmatis* and *M. tuberculosis* promoters by Bashyam et al. [8] showed that the respective –10 regions are highly similar to those of *E. coli* σ^{70} promoters; however their –35 regions exhibit greater sequence variability. The stated feature contrasting the one observed by Ramesh and Gopinathan [5] is however in agreement with that noticed by Kremer et al. [7] for mycobacterial promoters, and by Strohl [9] for *Streptomyces* promoters. *Streptomyces* promoters contain diverse sequences in their –35 regions and do not function in *E. coli* [9] For mycobacterial promoters, where apparent conservation in -35 region is absent, many of them possess TG dinucleotide in the immediate upstream of the –10 region, and thus they are termed "extended –10 promoters". The large variations among the mycobacterial promoters characterized thus far suggest that the consensus sequences are not representative of all mycobacterial promoters.

Consequently, a number of conflicting opinions regarding the presence and characteristics of consensus promoter sequences in the *Mycobacteria* have been aired in the literature [4].

An important objective in molecular biology is analyzing the DNA sequences for their structural and functional motifs. Macromolecular binding to specific sites of DNA involves recognition of a specific sequence pattern. In some cases, this pattern may be very distinct while in others it may be diffused. During examination of the molecular binding sites in a DNA, conventionally a consensus is derived by aligning an ensemble of sequences recognized by a common macromolecule. It is often found that the sequence pattern is never completely conserved. Efforts have also been made to develop statistical algorithms for the sequence analysis and motif prediction by searching for homologous regions or by comparing the sequence information with a consensus sequence [10]. This approach may fail or yield insufficiently accurate results when consensus sequences are difficult to define [10-11]. Wide variations existing within individual promoter sequences are primarily responsible for the unsatisfactory results yielded by the promoter-site-searching algorithms that in essence perform statistical analysis [12-13]. It can thus be inferred that recognition of mycobacterial promoter sequences and the important regions therein, require a powerful technique that is capable of unraveling those hidden pattern(s) in the promoter regions, which are difficult to identify manually. An artificial intelligence (AI) based modeling/classification paradigm known as 'artificial neural networks' (ANNs) possessing significant nonlinear pattern recognition and generalization capabilities has become available in the last decade. Accordingly, our objective in this chapter is to demonstrate: (i) the utility of ANNs for differentiating (classifying) mycobacterial promoter sequences from random (non-promoter) sequences, and (ii) an ANN-based calliper randomization (CR) approach [14-15] for determining the structurally and functionally important regions within the mycobacterial promoter sequences.

7.1.1 Overview of ANNs

ANNs are simplified counterparts of biological neural networks and based on the concept that a highly interconnected system of simple processing units (also called

"neurons" or "nodes") can learn and generalize complex inter-relationships existing between independent (ANN input) and dependent (ANN output) variables to an arbitrary degree of accuracy [16]. ANNs possess several unique characteristics and advantages as tools for molecular sequence analysis [17-18]. An important feature of ANNs is their adaptive nature where "learning by example" replaces the explicit "programming" approach conventionally followed in seeking solutions to modeling/classification problems. This feature makes ANNs very appealing in application domains where although the system to be modeled is only partly understood, there exists an example data set, which can be used in empirical (or "black-box") model development. In such instances, a network is made to capture (learn) the nonlinear interrelationships in the example input-output data via a procedure called "network training". In modeling applications, the network input-output may be representing a DNA sequence and its sequence-dependent feature (viz. DNA curvature, [19]), respectively, while in an ANN-based classification application, the network input-output is an appropriately coded DNA sequence and its class (viz. *E. coli* promoter prediction, [20]), respectively.

A typical ANN architecture used in modeling/classification tasks comprises multiple (usually three) layers housing a number of processing units in each layer. Units in the two successive layers are fully connected by means of "weighted" links. A commonly utilized multilayer network structure is the feedforward network wherein information flow occurs only in the forward direction i.e. from the input layer to the output layer. Such an ANN architecture is also amenable to parallel processing since the mathematical computations performed by a processing unit are independent of the computations done by other units in the same layer. A large number of interconnections comprised by an ANN makes it error-tolerant and thus can easily deal with even noise-corrupted data. A trained neural network encodes information about interrelationships existing between its inputs and outputs in a distributed fashion. That is, the captured information is spread over network's entire weight-space. This ANN feature makes it easy to optimize the network to deal with a large volume of data and to analyze its numerous input parameters.

Training of an ANN essentially consists of finding a set of connection weights such that the network accurately predicts the outputs corresponding to the input data in the example set. The error-back-propagation (EBP) method [21-22] currently represents the

most popular algorithm for training feedforward networks. Neural networks using the EBP training algorithm (hereafter referred to as EBPN) have been successfully used for various applications in biology involving nonlinear input-output modeling and classification (e.g., [23-25]). In fact, the overriding success of EBPNs in solving computational problems in biology and other sciences exceeds their biological significance [26]. In the present study, a three-layered feedforward network trained using the EBP algorithm has been developed for predicting the mycobacterial promoter sequences.

7.2 SYSTEM AND METHODS

The simulation programs for network training and promoter prediction were written in FORTRAN-77 and compiled using the Microsoft FORTRAN 5.0 compiler for the IBM PC and compatibles.

7.2.1 Data

The data for EBPN training was taken from our own compilation of the mycobacterial promoters (refer Table I, from chapter 6). The compiled promoter data set contains a total of 125 mycobacterial promoters out of which 80 have their transcription start site (TSS) mapped while the remaining 45 sequences are putative promoters. The promoters with the mapped TSS contain sequence stretches between –50 and +10 bp with respect to the TSS; the sequence stretch for the putative promoters lies between 15 bp upstream region of –35 box and 20 bp downstream of the –10 region. Length-wise, the compiled promoter sequences show variations owing to: (i) non-uniform availability of the nucleotide sequence upstream of the –35 region and downstream of the –10 region in the original reference, and (ii) variations in the spacer length. The shortest and the longest of the compiled sequences are 34 and 71 nucleotides long, respectively. In a few cases, two or more different sequence frames are considered for the same gene on the basis of alternate consensus probability. Thus, an overall set comprising 135 mycobacterial promoter sequences has been employed in this study.

7.2.2 Data representation for ANN-based classification

In ANN-based molecular sequence analyses, flexible sequence (network input) encoding schemes can be used for grasping the heterogeneous sequence features. Specifically, an individual nucleotide of a sequence can be represented using various coding strategies, such as *CODE-2*, *CODE-4* [27], *EIIP* code [14], and *wedge* and *twist* codes [28]. In classification studies by Nair et al. (1994) and Parbhane et al. (2000), it is observed that the CODE-4 strategy fares better than the other input coding approaches. In the CODE-4 scheme, each nucleotide is represented using a set of four binary digits as given by: C=0001; G=0010; A=0100; and T=1000. On the other hand, the above-stated other coding schemes utilize smaller number of bits or real numbers. For instance, mononucleotide representation schemes such as CODE-2 and EIIP respectively use two binary digits and a single electron ion interaction potential value for describing a nucleotide. Dinucleotide based wedge and twist codes use a single non-binary value to represent a nucleotide pair. Since CODE-4 requires maximum number (i.e., four) of input nodes to represent a single nucleotide, it produces a large-sized network as compared to other coding schemes. A large-sized network consequently increases the number of network weights (adjustable network parameters), which in turn helps in improving the classification accuracy of CODE-4 based EBPNs. Hence, in the present classification study, the CODE-4 scheme has been preferred for mononucleotide representation.

For *E. coli* promoter sequences, Mahadevan and Ghosh [20] employed a three-module approach with 98% classification accuracy. In their methodology, the first neural net module predicts the consensus boxes; the second module aligns the promoters to a length of 65 bases, and the third neural net module classifies the entire sequence of 65 bases while taking care of the possible interdependencies among the bases in the promoters. It is important to note that in the present study, the perfectly aligned promoter sequences are not being used as the network input. Consequently, the input sequence data do not require introduction of gaps for perfect alignment. The advantage of this approach is that it allows analysis of sequences where alignment is difficult or impossible to define.

7.2.3 Neural Network Simulation

The EBP network architecture used in this study is shown in Figure 7-1. As can be seen, a bias neuron each with the fixed output of +1 is added to network's input and hidden layers. Usage of bias neurons increases network's weight-space thus providing more adjustable parameters for performing the classification task. Analogues to other nodes in the same layer, the bias nodes are fully connected using weighted links to all the nodes in the next layer. Nodes in the input layer do not perform any numerical processing and thus act as "fan-out" units; all numerical processing is done by the hidden and output layer nodes and thus they are termed "active" nodes.

Training simulations for the network shown in Figure 7-1 were performed on a 486 AT equipped with the math co-processor. The EBPN training comprises: (i) presenting the network with an input pattern (sequence) from the example set, (ii) calculating the network output by propagating the input pattern through the hidden and output layers, (iii) computation of prediction error [difference between the desired (target) output and the actual network output], and (iv) utilization of the prediction error value to update the network weights with a view of minimizing the prespecified error function. Steps (i) and (ii) of this procedure are termed "forward pass" and steps (iii) and (iv) are termed the "reverse pass" through the network architecture. The error function to be minimized during network training is usually the *root-mean-squared-error* (RMSE). For details of RMSE function, please refer to chapter 2, section 2.2.3.

During training of an EBPN, the task of RMSE minimization is accomplished by adjusting the network weights using a gradient descent technique namely the *generalized delta rule* (GDR) [21]. In actual practice, however, it is not sufficient that the trained network accurately classifies sequences in the available example set. What is essential is that the network also correctly classifies new sequences, which are not part of the example set utilized for training the network. The network ability of correctly classifying new input patterns is known as "generalization ability" and the phenomenon, which adversely affects network's ability to generalize is known as "overfitting". Network overfitting occurs when: (i) network architecture contains more hidden nodes than necessary (known as "over-parameterization"), and (ii) network training continues over excessively large number of training epochs. If overfitting occurs, the network attempts to fit even the noise in the example data set at the cost of learning the smooth trends therein. In other words, an

overfitted network learns (memorizes) every minute detail thereby failing to capture the true information content within the example input-output data set. To prevent occurrence of overfitting, the available data is partitioned into two sets namely, the *training* set and the *test* set. While the former is used for training the network (i.e., for computing the prediction error and subsequent weight-updation), the latter (test set) is used to simultaneously evaluate network's generalization ability. For testing how well the network is generalizing, its classification performance is checked at the end of each training epoch by computing the RMSE with respect to the test set; the network weights that result into smallest RMSE for the test set are taken to be optimal since such a weight set exhibits best classification performance. Since 'more-than-necessary' hidden neurons also result in overtraining, the above-described training procedure is repeated by assuming varying number of hidden nodes in the network architecture. The optimal network architecture is the one, which houses just adequate number of hidden neurons and whose weight set (termed "optimal weight set") results in the least RMSE magnitude for the test set. The detailed description of obtaining an optimal network structure and associated weight set can be found, e.g., in Freeman and Skapura [29], and Tambe et al. [30]. For training an EBPN, the GDR algorithm for weight-updation makes use of two adjustable parameters namely, the learning rate (η) and momentum coefficient (α). Addition of the momentum term in the weight updation expression helps in accelerating the weight convergence and avoiding local minima on the error surface. In practice, values of both the GDR parameters are selected heuristically so as to obtain a network possessing good generalization ability.

Towards developing an optimal EBPN, the compiled promoter data set (135 sequences) was partitioned into training and test sets comprising 95 and 40 sequences, respectively. In order that the EBPN differentiates promoter sequences from the non-promoter ones, the training and test sets must also include non-promoter sequences. Accordingly, non-promoter sequences of length equal to 71 nucleotides were randomly generated wherein probability of occurrence of either A, T, G or C was equal to 0.25. The random sequences thus created were added to the promoter sequences in the training and test sets in 1:3 ratio. Thus the training and test sets comprised 380 and 160 sequences, respectively. For network training, the input data vectors (fragments coded in CODE-4) need to be of same size. Thus, the shorter fragments (i.e., < 71 bp) were uniformly padded

with 0.01 till each fragment was 284 (=71×4) elements long. The resulting training and test sets can be viewed as matrices of size (380×284) and (160×284), respectively.

The EBPN architecture (Figure 7-1) used for classifying the promoter sequences consists of 284 nodes in the input layer, and a single node in the output layer for representing whether the input sequence is a mycobacterial promoter. Accordingly, the target output for a promoter sequence was chosen to be unity and for a non-promoter, the target output was zero. For a given input sequence if the network output lies between 0.5 and 1.0 then the sequence is assumed to be a promoter, otherwise (i.e., network output < 0.5) it is a non-promoter.

7.3 RESULTS AND DISCUSIION

The training and test sets each comprising promoter and random sequences were utilized for obtaining an optimal network architecture - and the optimized weight set thereof - by following the network optimization procedure described earlier. The optimal network so developed, contains a single neuron in its hidden layer; increasing the number of hidden neurons beyond one did not increase the classification accuracy of the trained network. The RMSE profiles corresponding to the training and test sets for the optimized network are shown in Figure 7-2. It was observed that the weights at the 318th training epoch (η=0.6, α=0.4) correspond to the minimum RMSE (highest classification accuracy) with respect to the test set; thus these weights were taken as optimal. The optimal EBPN could correctly classify all the 380 sequences in the training set (100% classification accuracy). That is, the network could indeed differentiate between 95 promoter sequences and 285 random sequences. Moreover, the network correctly classified 155 sequences in the test set comprising 160 sequences (96.9% classification accuracy). It was also witnessed that the network did not predict any false positive i.e., none of the random sequences in the training/test sets were classified as mycobacterial promoter sequences.

7.3.1 Analysis using Calliper Randomization strategy

The above-described classification results in essence indicate that the optimized EBPN model possesses excellent capability of differentiating between a mycobacterial promoter sequence and a random sequence. In other words, the network model has

satisfactorily captured the hidden features that impart mycobacterial promoter characteristic to a given nucleotide sequence. It can be inferred further that the network model could now be utilized to identify important sub-regions in a promoter sequence. Towards this goal, we employ the caliper randomization (CR) approach wherein a mycobacterial promoter sequence is randomized in parts and applied to the trained network to examine whether the sequence still retains its promoter characteristic. If the network classifies the partly randomized sequence to be a non-promoter, then it can be concluded that the randomized region of the original promoter sequence governs its promoter functioning. For testing this hypothesis, the trained network was presented with mycobacterial promoter sequences randomized at fixed calliper lengths. Specifically, a fixed-sized calliper window of 10 nucleotides (approximately one turn of the helix) is chosen for randomization, which is moved from one end of the sequence to the other, in an overlapping fashion (refer Figure 7-3). Thus from a promoter sequence of 71 nucleotides, 62 sequences each containing a different randomized sub-region (window) could be formed. Upon randomizing all the promoter sequences in the training and test sets in this manner, the resulting sequences were applied to the optimized EBPN for predicting whether they maintain their promoter characteristic. In here, we present results pertaining only to the mycobacterial promoters whose TSS is mapped experimentally. The other type of compiled promoter sequences, namely "putative" promoters are called so since they comprise possible consensus boxes. However, the fact that their TSS is not mapped experimentally may lead to erroneous conclusions about mycobacterial transcription machinery. For this reason, the putative promoters are excluded from analysis via CR approach.

The classification results in respect of the partially randomized mycobacterial promoter sequences - whose TSS is known - are portrayed in Figure 7-4. In the figure, it is observed that depending upon the starting location of the randomized window, the resulting sequences are classified as non-promoters to varying extent. It can thus be opined that the starting location of the randomized window plays an important role while classifying a randomized promoter sequence. More importantly, it is noticed that when the starting position for the randomized calliper window lies in the –42 to –35 region, then the resulting sequences are predominantly classified as non-promoters. This observation suggests that the nucleotide content and its arrangement in the callipers located in the –42 to –35 region

are critical for mycobacterial promoters. When the calliper windows covering the spacer region and the –10 box are randomized, the original mycobacterial promoter sequences loose their promoter features. However, in this case the percentage of randomized sequences classified as non-promoters is not as high as that when calliper windows located in the –42 to –35 region are randomized. Thus, it is possible to infer that: (i) the –35 box and its upstream region play a critical role in mycobacterial promoter functioning, (ii) –10 box and spacer region also contribute towards mycobacterial promoter characteristics, and (iii) for promoter recognition the –10 region is not as important as –35 region.

In Figure 7-4, it is clearly noticed that the calliper window starting at location –38, when randomized, results in the highest percentage (i.e., 37%) for non-promoters. This observation suggests that the –38 to –29 region is most influential in determining whether a given compiled sequence is a mycobacterial promoter or not. For an in-depth scrutiny of the –38 to –29 region, it was divided into two sub-regions viz., -38 to –34 and –33 to –29, following which each of the two sub-regions was separately randomized. Upon randomizing all the promoter sequences in this manner, they were subjected to classification using the optimal EBPN. The results of such an analysis show that 57% of the sequences require randomization of the entire –38 to –29 region to alter their classification from promoters to non-promoters. It was also noticed that randomization of the –38 to –34 region and –33 to –29 region changes 36% and 7% of the original promoter sequences, respectively, to non-promoters.

Since the –38 to –29 region of the mycobacterial promoter sequences seems more influential in imparting them the promoter characteristics, it is of interest to study the nature of consensus nucleotide pattern for this sequence stretch. Towards this objective, all the mycobacterial promoters from the compilation (refer Table I, from chapter 6), were aligned with respect to their TSS and examined carefully to identify the consensus pattern in the –38 to –29 region. Thus, the consensus nucleotide pattern observed in the –38 to –29 region is: A_{31} C_{30} T_{43} T_{49} G_{44} G_{27} C_{34} C_{37} T_{37} C_{40}. Here it is seen that while the –38 to –34 region comprises a single ‘A’ and two ‘T’s, the –33 to –29 region is GC-rich. This observation suggests that the comparatively higher AT content in the –38 to –34 region assumes special significance for mycobacterial promoters. The –38 to –29 region is also

analyzed for purine/pyrimidine consensus pattern. Thus, purine (R) and pyrimidine (Y) consensus for –38 to –29 region is: R_{57} Y_{54} Y_{65} Y_{65} R_{52} R_{51} Y_{52} Y_{62} Y_{56} Y_{58}.

Using the results of the CR analysis, it is possible to get an insight into the sub-regions of the promoter sequences, which upon randomization were classified as non-promoters. Accordingly, a detailed examination of the randomized promoters was undertaken. It revealed that the mycobacterial promoters that were subjected to randomization (and were subsequently classified as non-promoters) in: (i) upstream region of –35 box, (ii) –35 region, (iii) spacer region, and (iv) –10 region, show resemblance to *E. coli* σ^{70} type promoters. More specifically, it is noticed that 32 mycobacterial promoters are sensitive to randomization within –38 to –29 region. Among these, 20 (64%) promoters exhibit resemblance to typical *E. coli* σ^{70} type; the remaining 12 (36%) belong to a typical mycobacterial type (GC rich –10 region) for their consensus sequence pattern.

7.4 CONCLUSION

To conclude, the results presented in this study suggest that ANNs can be gainfully employed for mycobacterial promoter sequence prediction. In view of the excellent performance of the optimized ANN in capturing the local and global features in the promoter sequences, it is possible to use them as feature detectors for locating the functionally important regions. The results of the CR strategy indicate that the network is indeed capable of acquiring the knowledge of regions that are structurally and functionally important. Additionally, the CR analysis results show that the method can be exploited in deriving consensus for other functionally important regions wherein weak consensus sequence pattern is observed.

7.5 REFERENCES

1. Harshey, R. M. and Ramkrishnan, T. (1977) *J. Bacteriol.*, **129**, 616-622.
2. Nakayama, M., Fujita N., Ohama, T., Osawa, S., Ishihama, A. (1989) *Mol. Gen. Genet.*, **218**, 384-389.
3. Ohama, T., Yamao, F., Muto, A., Osawa, S. (1987) *J. Bacteriol.*, **169**, 4770-4777.
4. Mulder, M.A., Zappe, H., Steyn, L.M. (1997) *Tuber. Lung Dis.*, **78**, 211-223.
5. Ramesh, G., Gopinathan, K. P. (1995) *Indian J. Biochem. Biophys.*, **32**, 361-367.
6. Bannantine, J.P., Barletta, R.G., Thoen, C.O., Andrews, R. E. Jr. (1997) *Microbiology*, **143**, 921-928.
7. Kremer, L., Baulard, A., Estaquier, J., Content, J., Capron, A., Locht, C. (1995) *J. Bacteriol.*, **177**, 642-653.
8. Bashyam, M.D., Kaushal, D., Das Gupta, S. K., Tyagi, A.K. (1996) *J. Bacteriol.*, **178**, 4847-4853.
9. Strohl, W.R. (1992) *Nucl. Acids Res.*, **20**, 961-974.
10. O'Neill, M.C. (1989a) *J. Biol. Chem.*, **264**, 5522-5530.
11. O'Neill, M.C. and Chiafari, F. (1989b) *J. Biol. Chem.*, **264**, 5531-5534.
12. Mulligan, M.E., Hawley, D.K., Entriken, R. and McClure, W.R. (1984) *Nucl. Acids Res.*, **12**, 789-800.
13. Mulligan, M. and McClure, W.R. (1986) *Nucl. Acids Res.*, **14**, 109-126.
14. Nair, T.M., Tambe, S.S. and Kulkarni, B.D. (1994) *FEBS Lett.*, **346**, 273-277.
15. Nair, T.M. (1997) *J. Biomol. Struct. & Dyn.*, **15**, 611-617.
16. Poggio, T. and Girosi, F. (1990) *Science*, **247**, 978-990.
17. Wu, C. H. (1997) *Comput. & Chem.*, **21**, 237-256.
18. Schneider, G and Wrede, P. (1998) *Prog. Biophys. Mol. Biol.*, **70**, 175-222.
19. Parbhane R. V., Tambe, S.S. and Kulkarni, B.D. (1998) *Bioinformatics*, **14**, 131-138.
20. Mahadevan, I. and Ghosh, I. (1994) *Nucl. Acids Res.*, **22**, 2158-2165.
21. Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1986) *Nature*, **323**, 533-536.
22. Rumelhart, D.E. and McClelland, J.L. (1986) *Parallel and Distributed Processing: Explorations in the Microstructure of Cognition*. MIT Press, Cambridge, MA.
23. Tolstrup, N., Engelbrecht, T.J. and Brunak, S. (1994) *J. Mol. Biol.*, **243**, 816-820.

24. Bisant, D. and Maizel, J. (1995) *Nucl. Acids Res.*, **23**, 1632-1639.

25. Uberbacher, E. C., Xu, Y. and Mural, R. J. (1996) *Method Enzymol.*, **266**, 259-281.

26. Zupan, J. and Gasteger, J. (1993) *Angew. Chem. Int. Ed. Engl.* **32**, 503-527.

27. Demeler, B. and Zhou, G. (1991) *Nucl. Acids Res.*, **19**, 1593-1599.

28. Parbhane R. V., Tambe, S.S. and Kulkarni, B.D. (2000) *Comput. & Chem.*, **24**, 699-711.

29. Freeman, J.A. and Skapura, D.M. (1992) Neural Networks Algorithms, Applications, and Programming Techniques. Addison-Wesley, Reading (MA).

30. Tambe, S.S., Kulkarni, B.D. and Deshpande, P.B. (1996) *Elements of Artificial Neural Networks With Selected Applications in Chemical Engineering and Chemical and Biological Sciences*, Simulation and Advanced Controls, Inc., Louisville.

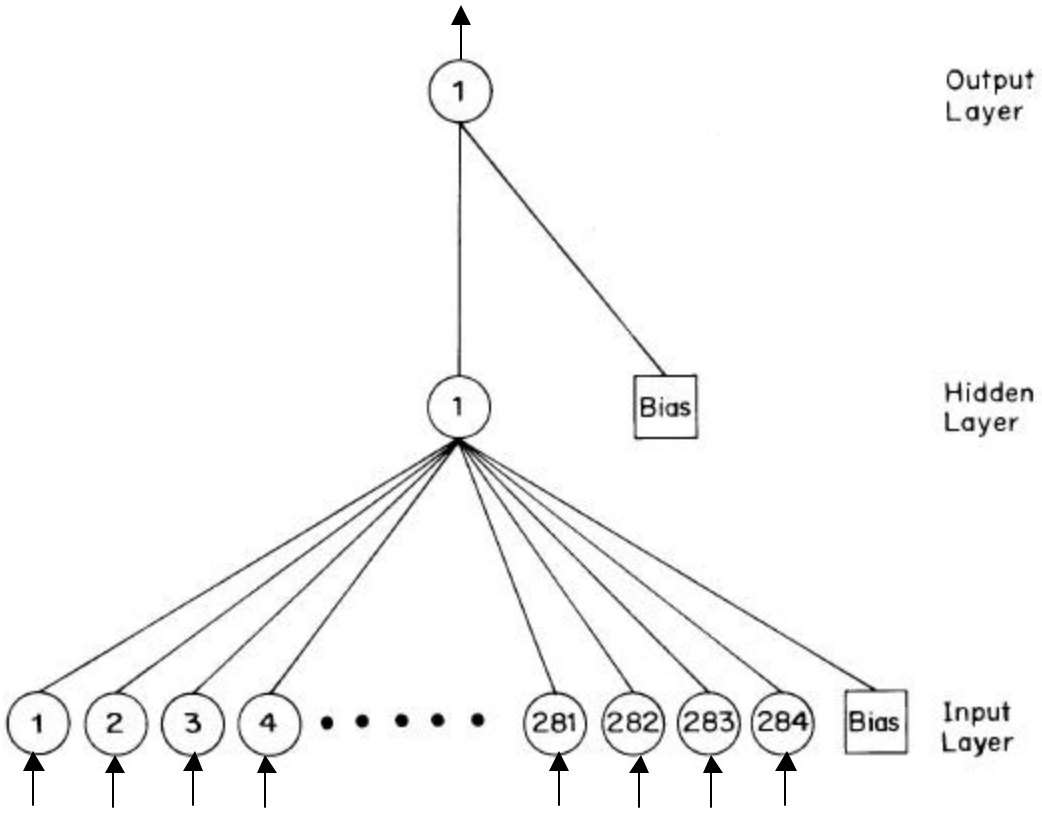


Figure 7-1: Schematic of the optimized EBP neural network used in the study (containing 284 neurons in the input layer and a single neuron each in the hidden and output layers).

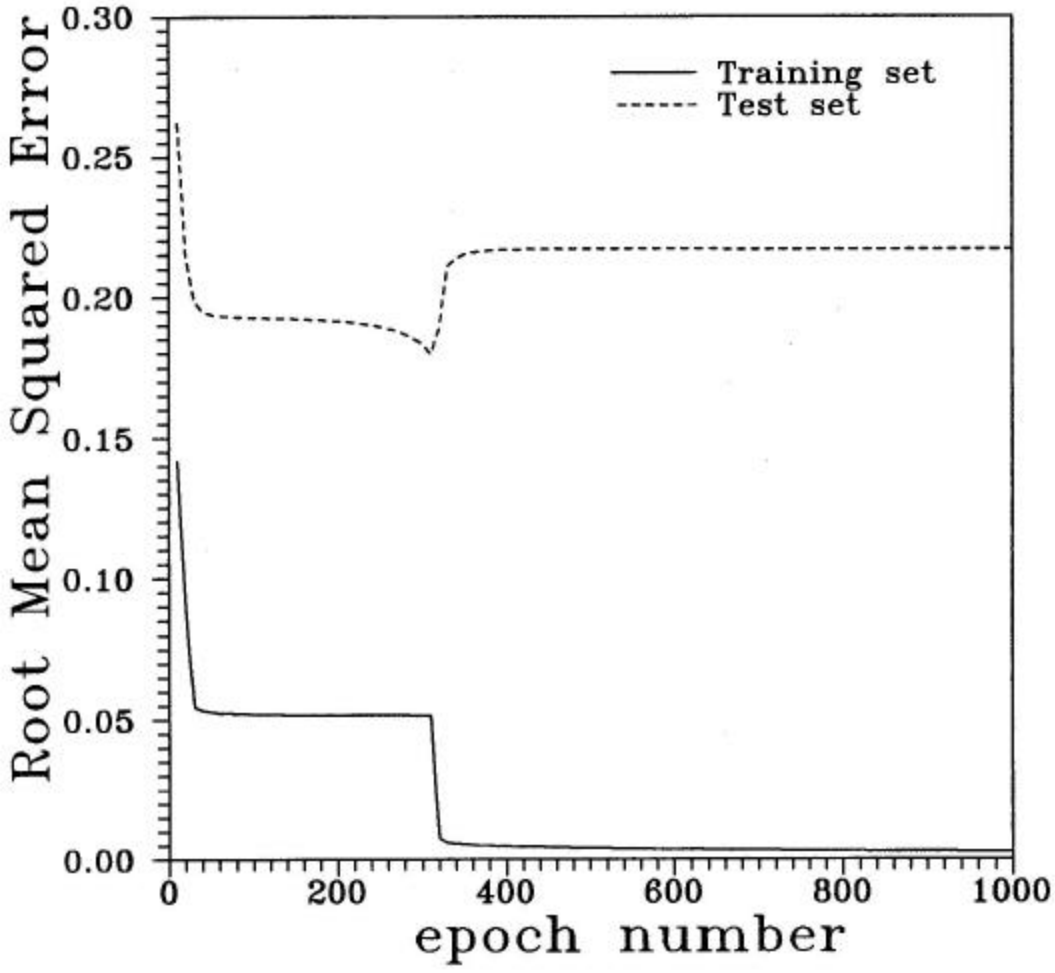


Figure 7-2: RMSE profiles corresponding to the training and test data sets.

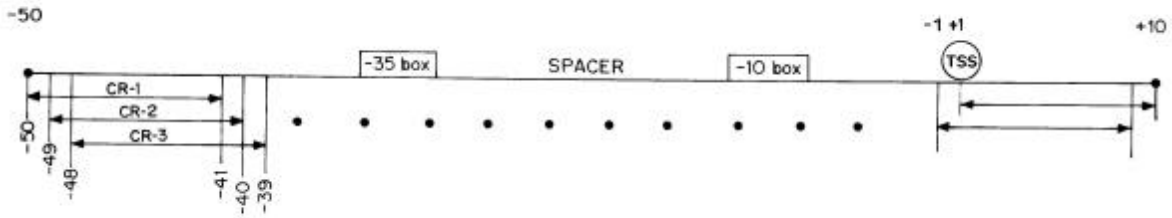


Figure 7-3: Calliper randomization scheme; CR-i refers to i^{th} calliper window.

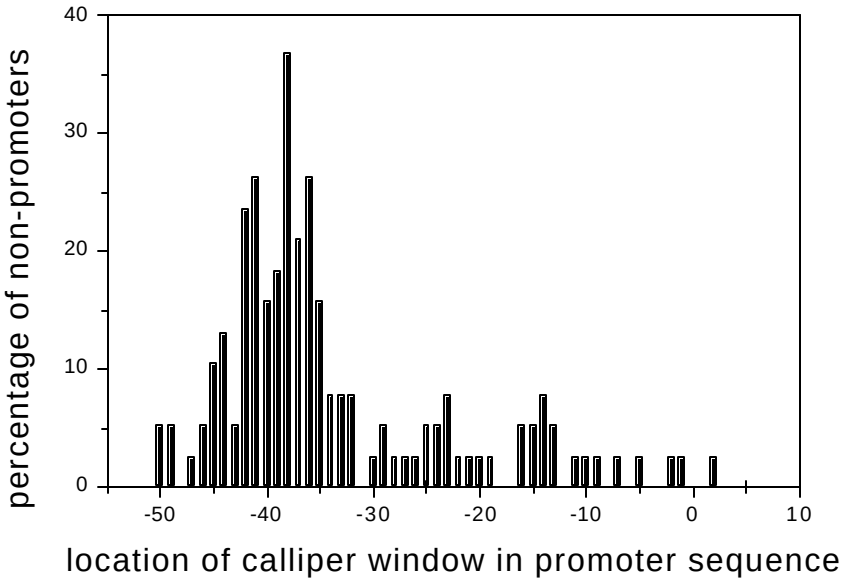


Figure 7-4: Classification results in respect of partially randomized mycobacterial promoter sequences. The X-axis refers to the location of 10 nucleotide-sized calliper window and Y-axis refers to the percentage of randomized promoters classified as non-promoters.

CHAPTER

8

Analysis of DNA curvature distribution in mycobacterial promoters using theoretical models

In this chapter, 125 mycobacterial promoters are analyzed for their DNA curvature distribution using several di- and tri-nucleotide dependent models of DNA curvature. Different models give similar behavior and therefore qualitative validation of the results. Mycobacterial promoters resembling to *E. coli* σ^{70} type have nearly 81% (85%) sequences having medium and high curvature profiles using dinucleotide dependent models. *Non-E. coli* σ^{70} type mycobacterial promoters have comparatively higher percent of low curvature profiles. Very few of extended -10 promoters have low curvature profiles. Mycobacterial promoters having A_nT_m ($n+m \geq 3$) tract in the upstream region of -35 box and repeated in phase with each other have high curvature profiles. *M. smegmatis* promoters have high curvature profiles compared to *M. tuberculosis* promoters.

8.1 INTRODUCTION

Transcription process in *Mycobacteria* may differ from *E. coli* and many other bacteria as mycobacterial genome has high G+C content which affects codon usage and promoter recognition sites in an organism. Mycobacterial promoters like *M. tuberculosis* 65 kDa [1], *M. bovis BCG* 64 kDa [2], and *M. leprae* 65 kD [3] are known to function in *E. coli*. However, mycobacterial promoters like *M. tuberculosis* 85A [4], recA [5] are known to be non-functional in *E. coli*. Thus depending on the choice of expression host, mycobacterial promoters are classified as *E. coli* type and Non-*E. coli* type promoters. *M. smegmatis* and *M. tuberculosis* promoter analysis by Bashyam et al. [6] showed that occurrence of TG motif near –10 region is functionally significant for those having nonfunctional –35 region. These promoters form a different class of promoters known as 'Extended –10 promoters'. The type of expression host, and the variation of the nucleotide sequence composition at –35 and –10 region of mycobacterial promoters [7] indicates that there exists immense variation in transcription initiation mechanism of mycobacterial promoters.

Transcription initiation is a multi-step, sequential process involving: a) binding of RNA polymerase to the promoter leading to formation of a relatively weak closed initiation complex; b) its isomerization to the more stable open complex that is accompanied by the separation of the DNA strands upstream and around the start site of the transcription; and c) RNA polymerase escapes from the promoter after cycles of abortive initiation forming the stable elongation complex [8]. Promoter DNA undergoes drastic conformational changes during initiation of transcription. The necessary condition for open complex formation is that RNA polymerase must bind and bend the promoter DNA. This bending and subsequent torquing is responsible for melting the DNA and the formation of open complex [9-10].

The role of DNA curvature has been studied extensively in *E. coli* [11-17]. The conformation of DNA is a function of its nucleotide sequence [18-19]. The three dimensional structure of DNA is the effect caused largely by interactions between neighboring base-pairs [20-29]. Generally, periodic repetitions of curved DNA in phase with the helical pitch cause the DNA to assume a macroscopically curved structure. Several theoretical models for estimating DNA curvature from di- or trinucleotides have been devised, and require various types of experimental data [23,

25, 28-31]. It is to be noted, however, that these models are being debated for their generality [32]. The importance of DNA conformation in transcription initiation is, however, clear and it would be interesting to study the DNA curvature distribution within the mycobacterial promoters especially in view of the large variation in their transcription mechanism. The objective of this chapter is to use six different di- and trinucleotide dependent models of curvature prediction for analysis of mycobacterial promoters.

8.2 SYSTEM AND METHODS

8.2.1 Data

The data for curvature analysis was taken from compilation of mycobacterial promoters (refer Table I, chapter 6). This data set contain 125 different mycobacterial promoters, out of which 80 promoters have their transcription start site (TSS) mapped while the other 45 are the putative promoters. In the listed compilation, we have considered the sequence stretches between –50 and +10 bp with respect to the TSS for the promoters whose TSS is mapped. For the putative promoters, we have documented the sequence stretch between 15-bp upstream region of –35 box and 20 bp downstream of the -10 box. The promoter sequence length varies from 34 to 71 nucleotides based on the availability of the nucleotide sequence upstream of the –35 region and downstream of the –10 region. In few cases, for the same gene two or more different sequence frames are considered based on the alternate consensus probability. Thus, 135 mycobacterial promoter sequences are used in this study.

8.2.2 Curvature Analysis

For the purpose of analyzing curvature distribution within mycobacterial promoter sequences, we have used the following dinucleotide models based on i) experimentally determined wedge angles [25]; ii) energy minimized values of roll and tilt angles [31, 33]; iii) X-ray crystallography of DNA oligomers [30]; and iv) Calladine-Dickerson rules [34-35]. The trinucleotide models used include: i) the model based on tabulation of preferred sequence locations on nucleosomes [23, 28]; and ii) DNase I cutting frequencies [29].

1. CURVATURE [36]: To obtain curvature map of each mycobacterial promoter, a window size of 21 bp nucleotide sequence is given as an input to the program and the curvature is obtained as an output. The results of this study are listed in Table I for each mycobacterial promoter. Various sub-groups of mycobacterial promoters are analyzed for nature of curvature profile and results are listed in Table II.
2. P. De Santis [33]: The curvature vector C (n,v) representing, in the complex plane (in modulus and phase), the directional change of the double helix axis between sequence number n and n+v is calculated for each mycobacterial promoter sequence in the compilation. For this calculation, roll and tilt angle values (in degrees) for the sixteen different dinucleotide steps in DNA are taken from Anselmi et al., [31]. In our analysis, we have used integration step value as 31 (~ three turns of B-DNA) in order to minimize the signal to noise ratio. The results of this study are also listed in Table I for each mycobacterial promoter. Various sub-groups of mycobacterial promoters are analyzed for the nature of curvature profile and these results are presented in Table II. Curvature dispersion σ^2 quantifies the central dispersion of the local helical axes with respect to the average direction of the double helix. The σ^2 plot of cyclically permuted DNA sequence allows an easy alternative to experimental permutation assay for DNA tracts up to 700 bp long. Hence, σ^2 plots of cyclically permuted mycobacterial promoters are prepared to see exact position of molecular bend locus. For simplicity, of analysis mycobacterial promoter region is divided into the following five sub-regions: i) region above -35 box, ii) -35 region, iii) spacer region, iv) -10 region, and v) region below -10 box. The position of molecular bend locus, for each mycobacterial promoter, with respect to the sub-regions specified above, is mentioned in Table III.
3. Calladine-Dickerson Rule [34-35]: Calladine proposed four rules to understand the sequence-dependent departures from classical B-DNA due to simple steric hindrance of nearest neighbor purines on opposite strands. He suggested that the DNA chains may overcome these steric clashes in four possible ways: i) the helix twist angle may be reduced, ii) the base pairs can rotate along their long axes, iii) the DNA backbone can shift sideways towards the pyrimidines, and iv) the propeller twist can be suppressed. Dickerson quantified this by constructing four

sum functions (Σ_1 to Σ_4), by means of which the base sequence can be used to calculate the expected local variation in helix twist (Σ_1), base plane roll (Σ_2), torsion angle difference at the two ends of the base pair (Σ_3), and flattening of propeller twist (Σ_4). DNA helical structure variation at the molecular bend locus is studied here for mycobacterial promoters using Calladine-Dickerson rules. For this analysis, we have taken 11-bp long sequence stretch obtained by taking five nucleotides on either side of the molecular bend locus of each mycobacterial promoter. For brevity, only Σ_1 function plots for the promoters whose TSS is mapped are shown in Figure 8-1.

4. Propeller Twist [30]: It is known that different types of dinucleotide step have different levels of conformational flexibility, which is very closely related to the Propeller-Twist. Propeller Twist values are obtained from X-ray crystallography of DNA oligomers. Dinucleotides with a large propeller-twist have a tendency to be more rigid than dinucleotides with low propeller twist. Higher (less negative) values correspond to higher flexibility. Flexibility profile was plotted using the propeller twist values from X-ray crystallography of DNA oligomers for overlapping dinucleotides.
5. DNase I derived bendability parameters [29]: The productive binding of Bovine pancreatic deoxyribonuclease I (DNase I) requires DNA to be bent toward the major groove (positive roll). Base sequences that are flexible or inherently bent towards the major groove should therefore be more accessible to DNase I cleavage. DNase I cutting frequencies on naked DNA can be used as a quantitative measure of anisotropic bendability (major groove compressibility). Bendability profile was calculated using DNase I derived bendability parameters for overlapping trinucleotides of each mycobacterial promoter sequence.
6. Location Preference [23]: From experimental investigations of the positioning of DNA in nucleosomes, it has been found that certain trinucleotides have strong preference for having minor grooves facing either towards or away from the nucleosome core. Based on the premise that flexible sequences can occupy any rotational position on nucleosomal DNA, while rigid sequences will be restricted in rotational location. We have calculated DNA flexibility profile using these location preference values for mycobacterial promoters at each position considering overlapping trinucleotides.

8.3 RESULTS AND DISCUSSION

The curvature distribution for various mycobacterial promoters as calculated using different models show similar trends. In order to aid the analysis the results obtained using: i) experimentally determined wedge angles and ii) energy minimized values of roll and tilt angles, have been compared. The extent of curvature obtained using these models has been classified in terms of low, medium or high curvature and the results of the two models corroborate each other for most of the promoters barring a few promoter entries (e.g. *M. tuberculosis* T3, T6, T101, T129, T130, recA, rrnA P1, gyrA, cpn60, rrnA PCL1, 16S rRNA, metA, rpsL etc.) where the prediction of the two models differ.

In order to obtain a better insight for the results obtained by these two models, mycobacterial promoters are sub-divided into various groups. These groups are as follows: i) Class I: mycobacterial promoters resembling to *E. coli* σ^{70} type promoters, ii) Class II: mycobacterial promoters which are different from *E. coli* σ^{70} type promoters, and constituting a class known as typical mycobacterial promoters, iii) Class III: Extended −10 promoters, iv) mycobacterial promoters having optimum (17±1 bp) spacer length, v) mycobacterial promoters having high (≥50%) AT content, vi) mycobacterial promoters having A_nT_m (n+m≥3) tract repeated in phase with each other and present at the upstream of −35 box, vii) *M. tuberculosis* promoters, viii) *M. smegmatis* promoters, and ix) entire mycobacterial promoter compilation. The curvature analysis of promoters classified in these groups is listed in Table II. From Table II, it can be seen that *E. coli* σ^{70} type mycobacterial promoters have 15% (19%), 60% (67%), and 25% (14%) of low, medium, and high curvature profiles using curvature models of Shpigelman et al., [36] (P.De Santis et al., [33]). This distribution indicates that mycobacterial promoters resembling to *E. coli* σ^{70} type (Class I) have nearly 81% (85%) sequences having medium and high curvature profiles. Very few i.e., 19% (15%) promoter sequences are having low curvature profiles. Considering percent distribution of curvature existing among *E. coli* σ^{70} type mycobacterial promoters, we can say that these promoters might be having good promoter activity. The analysis also indicates that the Non-*E. coli* σ^{70} type mycobacterial (Class II) promoters have 22% (27%), 56% (54%), and 22% (19%) of low, medium, and high curvature profiles (using both curvature models). This group of mycobacterial promoters has comparatively higher percent of low

curvature profiles indicating that Non - *E. coli* σ^{70} type mycobacterial promoters might be expressed poorly compared to *E. coli* σ^{70} type mycobacterial promoters. The curvature models applied to the extended −10 (Class III) promoters show 17% (4%), 25%(58%), and 58% (38%) of low, medium, and high curvature profiles. The percent distribution of these promoters indicates that very few of these promoters have low curvature profiles. Extended −10 promoters might therefore have reasonably high promoter activity. *M. tuberculosis* T101, *M. smegmatis* S6, S16, and S19 promoters are extended −10 promoters, which are strongly curved. For such mycobacterial promoters sequence of the −35 region seems to be less important due to presence of extended TG motif in the immediate neighborhood of −10 box along with the high curvature existing within it. Mycobacterial promoters lacking consensus sequence at −35 and are curved are *M. tuberculosis* T150, *M. smegmatis* S12, S14, S30, and S35. Here curvature along with −10 region might be useful for promoter activity although they do not possess TG motif in the immediate neighborhood of −10 box. The mycobacterial promoters having optimum (17±1 bp) spacer length have 9% (11%) of sequences having low curvature profiles by both the models. Majority of sequences from this class has curved structure. The favorable flexibility and/or curvature of DNA may compensate somewhat for a sub optimal spacing of 16 or 18 base pairs between −35 and −10 regions during transcription initiation. The mycobacterial promoters with high % of AT have 12% (15%), 54% (58%) and 35% (27%) of sequences possess low, medium and high curvature profiles, respectively. The occurrence of curvature is obvious for majority of sequences from this class due to their high percentage of AT content. Among mycobacterial promoters with A_nT_m $(n+m \geq 3)$ tract repeated in phase with each other and present at the upstream of −35 box, 58% (50%) of sequences have high curvature trends. These promoters having upstream sequences, which can be expected to produce curvature in the DNA helical axis might be transcriptionally active promoters. *M. tuberculosis* promoters have 14% (9%), and *M. smegmatis* promoters have 29% (25%) of high curvature profiles. Such a percent distribution may be one of the causative factors for *M. smegmatis* to express better than *M. tuberculosis.* For the analysis performed in the Table II, it is important to realize that the % value of curvature predictions by both the models sometimes differ significantly due to different conditions defined for low, medium, and high curvature

profiles; and in few cases predictions by two models lie on the boundary conditions of low and medium, or medium and high curvature profiles. The sample size considered in this analysis is also small, and can affect large difference in the predictions by two models. Results listed in Table II should therefore be used to see only qualitative and semi-quantitative trends.

According to CURVATURE software, curvature maxima for *M. tuberculosis* gyrB P1, *M. bovis BCG* alpha, *M. fortuitum* rrnA P1, *Mycobacteriophage L5* 71P1, *M. neoaurum* rrnA PCL1, and rrnA P3 promoters lies above 0.3 DNA curvature units and it is present between −35 and −10 regions. It will be interesting to study the transcription initiation mechanism in these promoters because in *E. coli* it is shown that curvature between −35 and −10 regions seems to correlate significantly with promoter activity. In such cases the curved structure of promoter DNA enhances the binding of *E. coli* RNA polymerase to the promoter, when the curve is oriented correctly relative to the potential −10 and −35 regions, and it also facilitate unwinding of the −10 region by thermal motion, as the DNA vibrates back and forth in solution between twisted and curved forms [11].

σ^2 plots of cyclically permuted mycobacterial promoters should allow an alternative to the experimental permutation assay for determining molecular bend locus of a mycobacterial promoter sequence. The model has been successful in predicting the experimental results for other systems [33, 38-39], while promoters analyzed here have not been subjected to any such experimental investigations and hence the theoretical predictions could not be tested. In Table III, we have evaluated the percent occurrence of position of molecular bend locus in i) region above −35 box, ii) -35 region, iii) spacer region, iv) -10 region, and v) region below −10 box. For this analysis, we have separated entire promoter compilation into two groups i) promoters whose TSS is mapped (true promoters) and ii) putative promoters. According to percent distribution for true promoters, molecular bend locus lies predominantly in the spacer region and region below −10 box. The 16%, 16%, 30%, 6% and 32% of true mycobacterial promoter sequences show that their molecular bend locus lies in the region above −35 box, -35 region, spacer region, -10 region, and region below −10 box, respectively. For putative promoters 8%, 23%, 15%, 6% and 48% of sequences show their molecular bend locus in region above −35 box, -35 region, spacer region, -10 region, and region below −10 box, respectively. Thus, for

true as well as putative mycobacterial promoters spacer region and region below −10 box seems to be of frequent occurrence for the location of molecular bend locus. Similar studies by Nair and Kulkarni [40] on *E. coli* promoter sequences showed that 60% of these promoters have their minima (molecular bend locus) lying in the spacer region. However, for mycobacterial promoters position of molecular bend locus can occur with varying percent distribution at region above −35 box, -35 region, spacer region and region below −10 box. Thus, mycobacterial promoters have variation towards position of molecular bend locus compared to *E. coli* promoters.

Calladine and Dickerson rule (Σ_1-Σ_4) gives a way of revealing possible structural homology between regions of DNA, when the similarity is not obvious by direct comparison of sequence alone. The helical structure variation at the molecular bend locus for the true mycobacterial promoters is sub-grouped according to the position of molecular bend locus. Thus, Figure 8-1 is subdivided into five plots. The helical structure variation obtained using Σ_1 function at the molecular bend locus lying in i) region above -35 box, ii) −35 region, iii) spacer region, iv) −10 box, and v) region below −10 box shows that each sub-group has structural similarity within that particular sub-group. The other sum functions also uphold the structural similarities (results not shown). The analysis of the sequence at the minima reveals that there exists homology among these sequences irrespective of the exact position of minima. The regions that are localized for mycobacterial promoters show significant commonality in structure, which is evident from the Σ_1 function plot. There seems to exist some structural commonalties among the each sub-group of mycobacterial promoters. We can therefore group the promoters based on the common structural features and advocate the notion of "consensus structure" suggesting their common biological significance. The variation from these consensus structures can account for varying strength of the promoters. Such an analysis might help us in designing experiments to define the exact location and function of a promoter.

Although the entire mycobacterial promoter compilation has been analyzed using other curvature models [23, 29, 30], the results obtained using only three models are presented.

Mycobacterial promoters that are strongly curved are *M. tuberculosis* T150, and gyrB P1; *M. Leprae* 65KD; *M. smegmatis* S6, S12, S14, S30, S35, and rrnB; *M. Phlei* rrnA P2; *M. abscessus* rrnA P4, rrnA P2, and rrnA P3; *M. chelonae* rrnA P2,

rrnA P3, and rrnA P4. Figure 8-2 shows the curvature map expressed in DNA curvature units of these promoters using CURVATURE software. The curvature maxima of these curvature maps correspond to region having more curved structure. Figure 8-3- a & b presents the curvature analysis using energy-minimized values of roll and tilt angles. The curvature vector is a complex function of the sequence with the modulus representing the deviation and the phase indicating the relative direction. The curvature diagrams for these mycobacterial promoters clearly show a DNA tract characterized by both a high curvature modulus (see Figure 8-3-a) and a constant phase (Figure 8-3-b). Figure 8-4 shows flexibility profiles based on propeller twist values from X-ray crystallography of DNA oligomers. Dinucleotides with a large propeller-twist have a tendency to be more rigid than dinucleotides with low propeller-twist. Thus, sequence positions corresponding to higher (less negative) values represent regions of higher flexibility for mycobacterial promoter. Figure 8-5 presents flexibility profile calculated using trinucleotide model based on preferred sequence location on nucleosomes. Sequence positions corresponding to lower values of location preference represent more flexible region of mycobacterial promoter, which have less preference for being positioned specifically. Figure 8-6 shows bendability profile in the mycobacterial promoters calculated using DNase I derived bendability parameters. Sequence position corresponding to higher bendability parameters represent to higher propensity for major groove compressibility of mycobacterial promoter. Essentially all the models predict similar behavior for these promoters. Thus nucleotide sequence position corresponding to high (low) curvature is showing high (low) curvature trend with all the other models. Mycobacterial promoters *M. abscessus* rrnA P4, rrnA P2, and rrnA P3; *M. chelonae* rrnA P2, rrnA P3, and rrnA P4 have similarity in their curvature trends as their nucleotide sequence shows maximum homology with each other. The similar curvature trends suggest common mechanism of transcription initiation.

Regions with high DNA curvature would be expected to exhibit anomalous mobility by the gel electrophoresis assay. It will be of interest to examine fragments containing these regions for the structural feature of DNA curvature, and the corresponding functional feature of transcriptional activation. Plasmids containing stiff, flexible or curved DNA structure near the cleavage site of commonly used restriction enzymes can be helpful for studying the role of DNA structure in transcription mechanism of mycobacterial promoters.

Thus, analysis of DNA curvature distribution for mycobacterial promoters reveals the following important features. i) The curvature distribution for various mycobacterial promoters calculated using different models show similar trends. ii) Mycobacterial promoters resembling to *E. coli* σ^{70} type have nearly 81% (85%) sequences having medium and high curvature profiles. iii) Non- *E. coli* σ^{70} type mycobacterial promoters have comparatively higher percent of low curvature profiles. iv) Very few of extended -10 promoters have low curvature profiles. v) Mycobacterial promoters having A_nT_m $(n+m \geq 3)$ tract in the upstream region of -35 box and repeated in phase with each other have high curvature profiles. vi) *M. smegmatis* promoters have high curvature profiles compared to *M. tuberculosis* promoters.

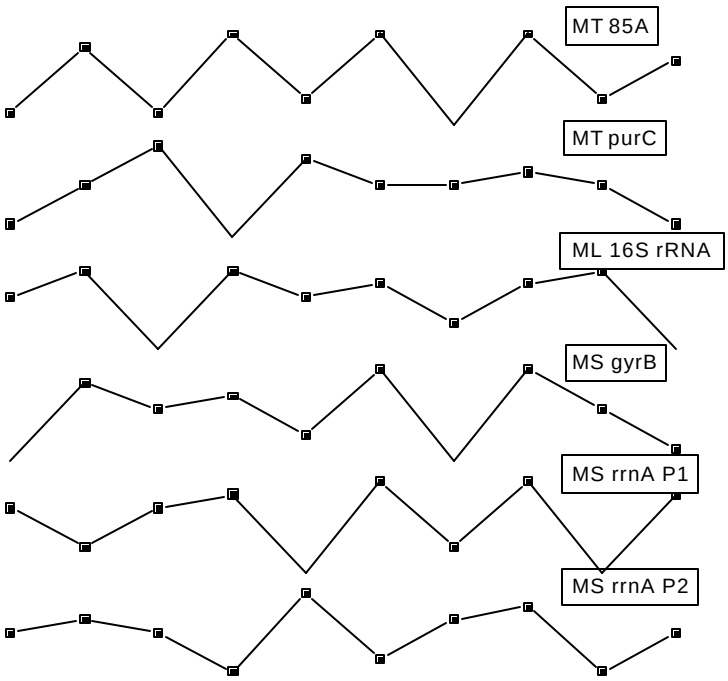
Experimental studies based on curvature distribution and its role in transcription mechanism for particular mycobacterial promoter(s) or representative examples from various groups of mycobacterial promoters showing some distinct features will throw light on our understanding of transcription mechanism of *Mycobacteria*.

8.4 REFERENCES

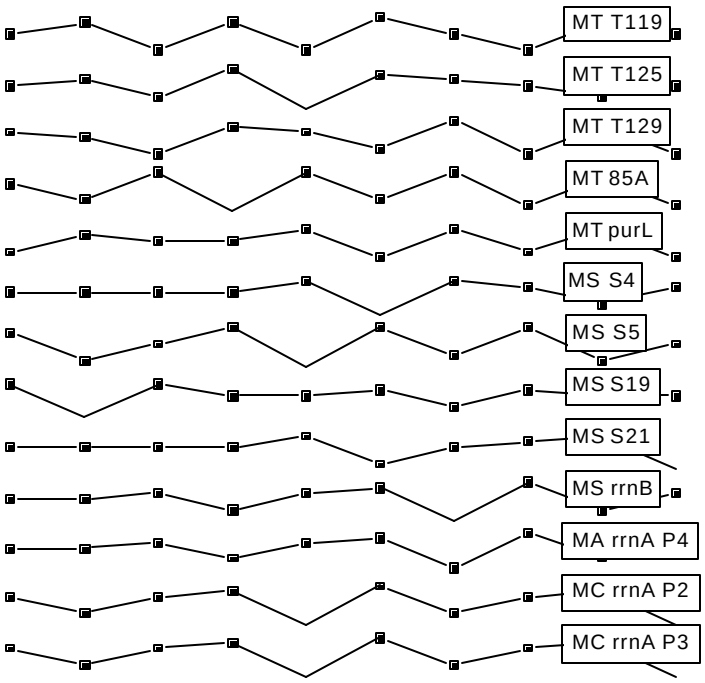
1. Shinnick, T. M. (1987) *J. Bacteriol.*, **169**, 1080-1088.
2. Thole, J. E., Dauwerse, H. G., Das, P. K., Groothuis, D. G., Schouls, L. M. and van, E. J. D. (1985) *Infect. Immun.*, **50**, 800-806.
3. Mehra, V., Sweetser, D. and Young, R. A. (1986) *Proc. Natl Acad. Sci. USA.* **83**, 7013-7017.
4. Kremer, L., Baulard, A., Estaquier, J., Content, J., Capr on, A. and Locht, C. (1995) *J. Bacteriol.*, **177**, 642-653.
5. Movahedzadeh, F., Colston, M. J. and Davis, E. O. (1997) *J. Bacteriol.*, **179**, 3509-3518.
6. Bashyam, M. D., Kaushal, D., Das Gupta, S. K. and Tyagi, A. K. (1996) *J. Bacteriol.*, **178**, 4847-4853.
7. Mulder, M.A., Zappe, H., Steyn, L.M. (1997) *Tuber. Lung Dis.*, **78**, 211-223.
8. Hoopes C. B. and McClure W. R. (1987) Strategies in regulation of transcription initiation; in Escherichia coli and Salmonella typhimurium (ed.) F C Neidhart (Washington DC: American Society of Microbiology) pp 1231-1239.
9. Ramstein, J. and Lavery, R. (1988) *Proc. Natl. Acad. Sci. USA*, **85**, 7231-7235.
10. Parvin, J. D., McCormick, R. J., Sharp, P.A., Fisher, D.E. (1995) *Nature*, **373**, 724-727.
11. Collis C. M., Molly, P. L., Both, G. W. and Drew, H.R. (1989) *Nucl. Acids Res.*, **17**, 9447-9468.
12. Plaskon, R.R. and Martell, R.M. (1987) *Nucl. Acids Res.*, **15**, 785-796.
13. Bracco, L., Kotlarz, D., Kolb, A., Diekmann, S. and Buc, H. (1989) *EMBO J.*, **8**, 4289-4296.
14. Ohyama, T., Nagumo, M., Hirota, Y. and Sakuma, S. (1992) *Nucl. Acids Res.*, **20**, 1617-1622.
15. Gaal, T., Rao, L., Estrem, S., Yang, J., Wartell, R. and Gourse, R. (1994) *Nucl. Acids Res.*, **22**, 2344-2350.
16. Engelhorn, M. and Geiselmann, J. (1998) *Mol. Microbiol.*, **30**, 431-441.
17. Aiyar, S.E., Gourse, R.L. and Ross, W. (1998) *Proc. Natl. Acad. Sci. USA*, **95**, 14652-14657.
18. Trifonov, E.N. (1985) *CRC Crit. Rev. Biochem.*, **19**, 89-106.
19. Hagerman, P.J. (1990) *Annu. Rev. Biochem.*, **59**, 755-781.

20. Klug, A., Jack, A., Viswamitra, M.A., Kennard, O., Shakked, A. and Steitz, T.A. (1979) *J. Mol. Biol.*, **131**, 669-680.
21. Dickerson, R.E. and Drew, H.R. (1981) *J. Mol. Biol.*, **149**, 761-786.
22. Hagerman, P.J. (1984) *Proc. Natl. Acad. Sci. USA*, **81**, 4632-4636.
23. Satchwell, S.C., Drew, H.R. and Travers, A.A. (1986) *J. Mol. Biol.*, **191**, 659-675.
24. Calladine, C.R., Drew, H.R. and McCall, M.J. (1988) *J. Mol. Biol.*, **201**, 127-137.
25. Bolshoy, A., McNamara, P., Harrington, R.E. and Trifonov, E.N. (1991) *Proc. Natl. Acad. Sci. USA*, **88**, 2312-2316.
26. Hunter, C.A. (1993) *J. Mol. Biol.*, **230**, 1025-1054.
27. Hunter, C.A. (1996) *Bioessays*, **18**, 157-162.
28. Goodsell, D.S. and Dickerson, R.E. (1994) *Nucl. Acids Res.*, **22**, 5497-5503.
29. Brukner, I. Sanchez, R., Suck, D. and Pongor, S. (1995) *EMBO J.*, **14**, 1812-1818.
30. Hassan M. A. EI and Calladine, C. R. (1996) *J. Mol. Biol.*, **259**, 95-103.
31. Anselmi, C., Bocchinfuso, G., Santis, P. De, Savino, M. (1999) *J. Mol. Biol.*, **286**, 1293-1301.
32. Dlakic, M. and Harrington, R.E. (1996) *Proc. Natl. Acad. Sci. USA*, **93**, 3847-3852.
33. Santis, P. De, Palleschi, A., Savino, M., Scipioni, A. (1988) *Biophys. Chem.*, **32**, 305-317.
34. Calladine, C. R. (1982) *J. Mol. Biol.*, **161**, 343-352.
35. Dickerson, R. E. (1983) *J. Mol. Biol.*, **166**, 419-441.
36. Shpigelman E S, Trifonov E N and Bolshoy A. (1993) *Comp. Appl. Biosci.*, **9**, 435-440.
37. Trifonov E N and Ulanovsky L E (1987) Inherently curved DNA and its structural elements. In Wells, R D and Harvey, S C (eds) Unusual DNA Structures, Springer-Verlag, Berlin, pp 173-187.
38. Santis, P. De, Palleschi, A., Savino, M. and Scipioni, A. (1990) *Biochemistry*, **29**, 9269-9273.
39. Santis, P. De, Palleschi, A., Savino, M. and Scipioni, A. (1992) *Biophys. Chem.*, **42**, 147-152.
40. Nair, T. M. and Kulkarni, B.D. (1994), *Biophys. Chem.*, **48**, 383-393.

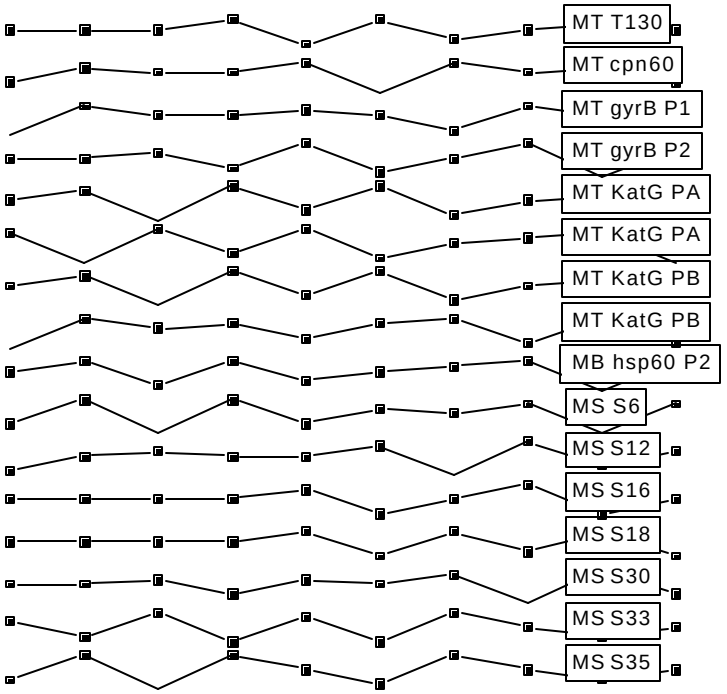
Region 1:



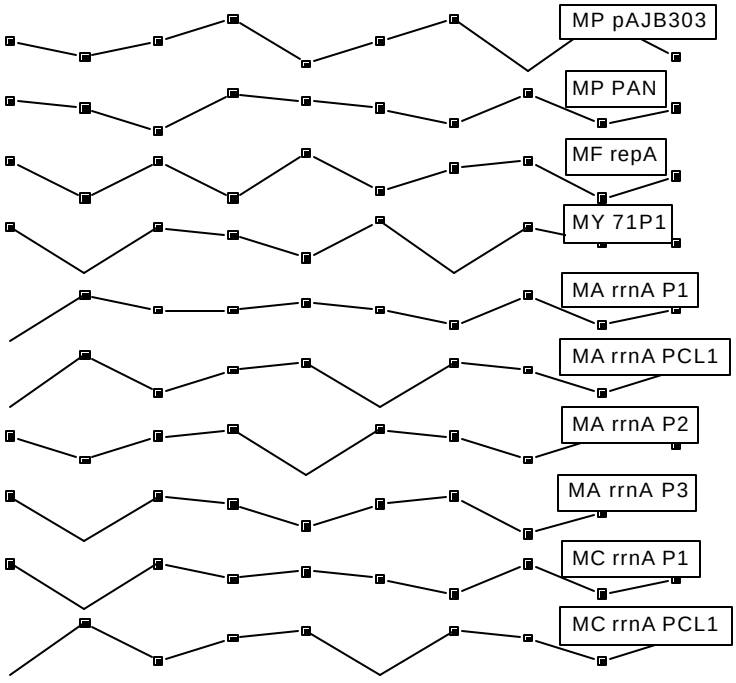
Region 2:



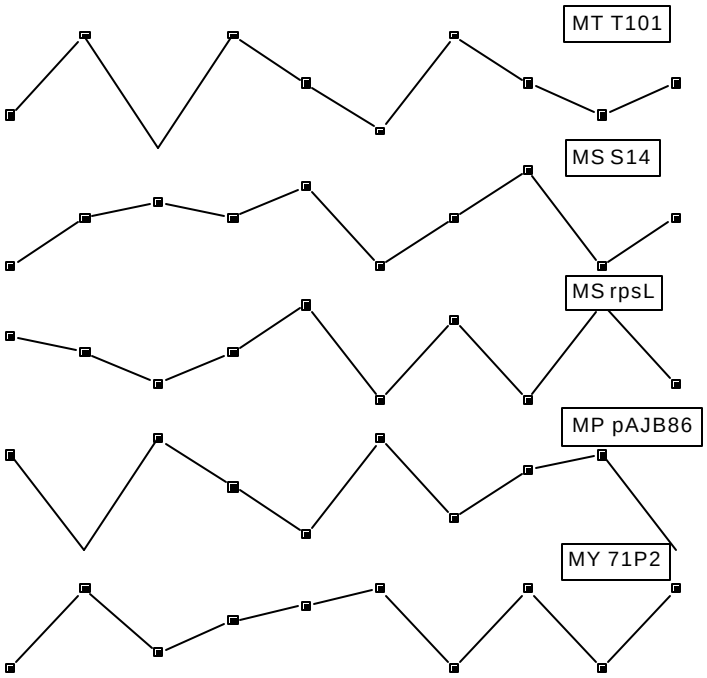
Region 3:



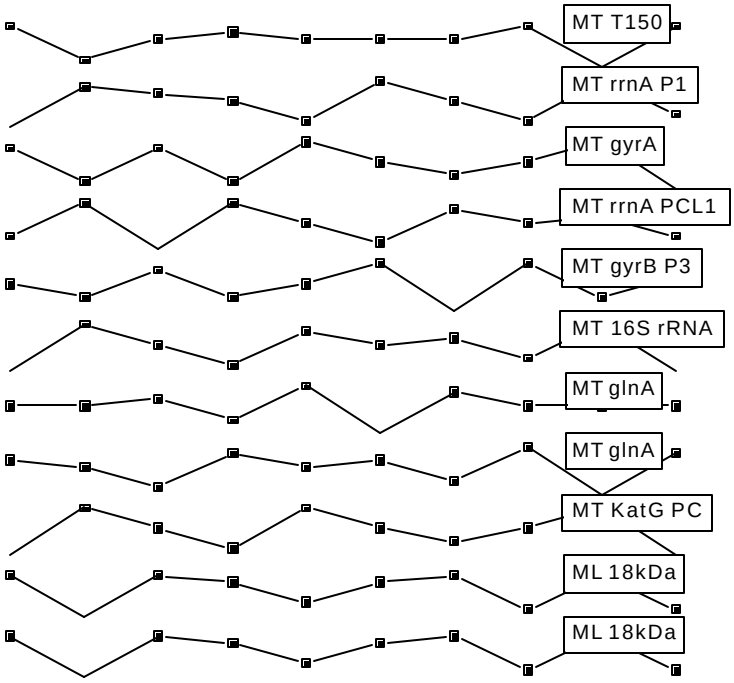
Region 3:



Region 4:



Region 5:



Region 5:

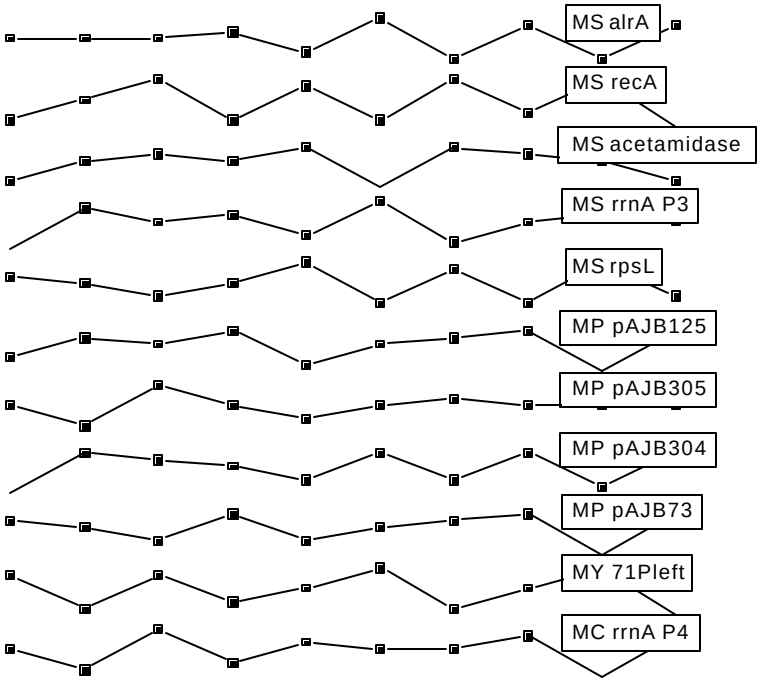


Figure 8-1: Σ_1 function plots for the true mycobacterial promoters, sub-grouped depending upon the location of the molecular bend locus.

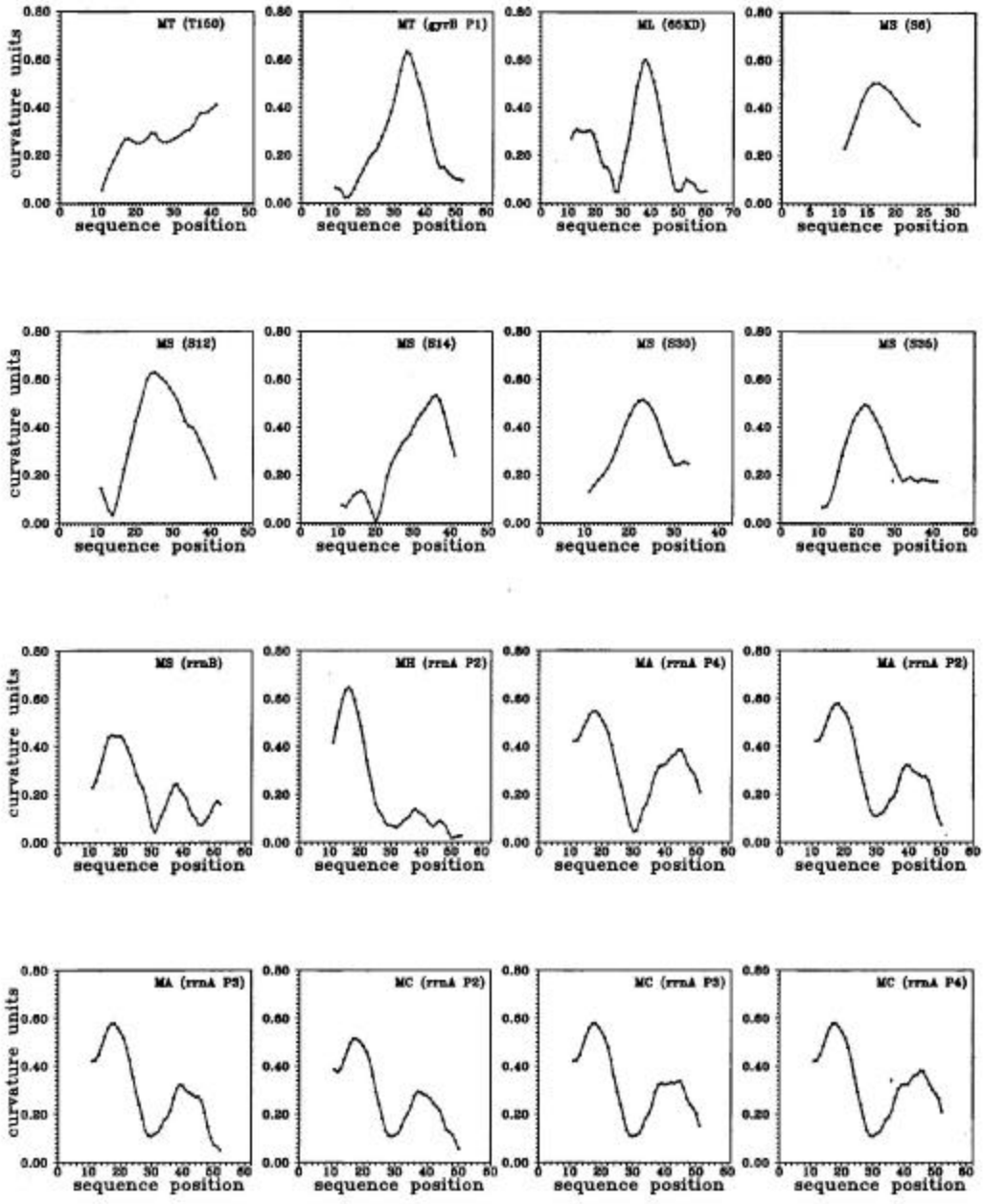


Figure 8-2: Curvature map obtained using experimentally determined wedge angles for mycobacterial promoters. Curvature is expressed in DNA curvature units [37] where one curvature unit corresponds to the mean DNA curvature in the crystalline nucleosome (1⁄42.8 $\overset{o}{A}$).

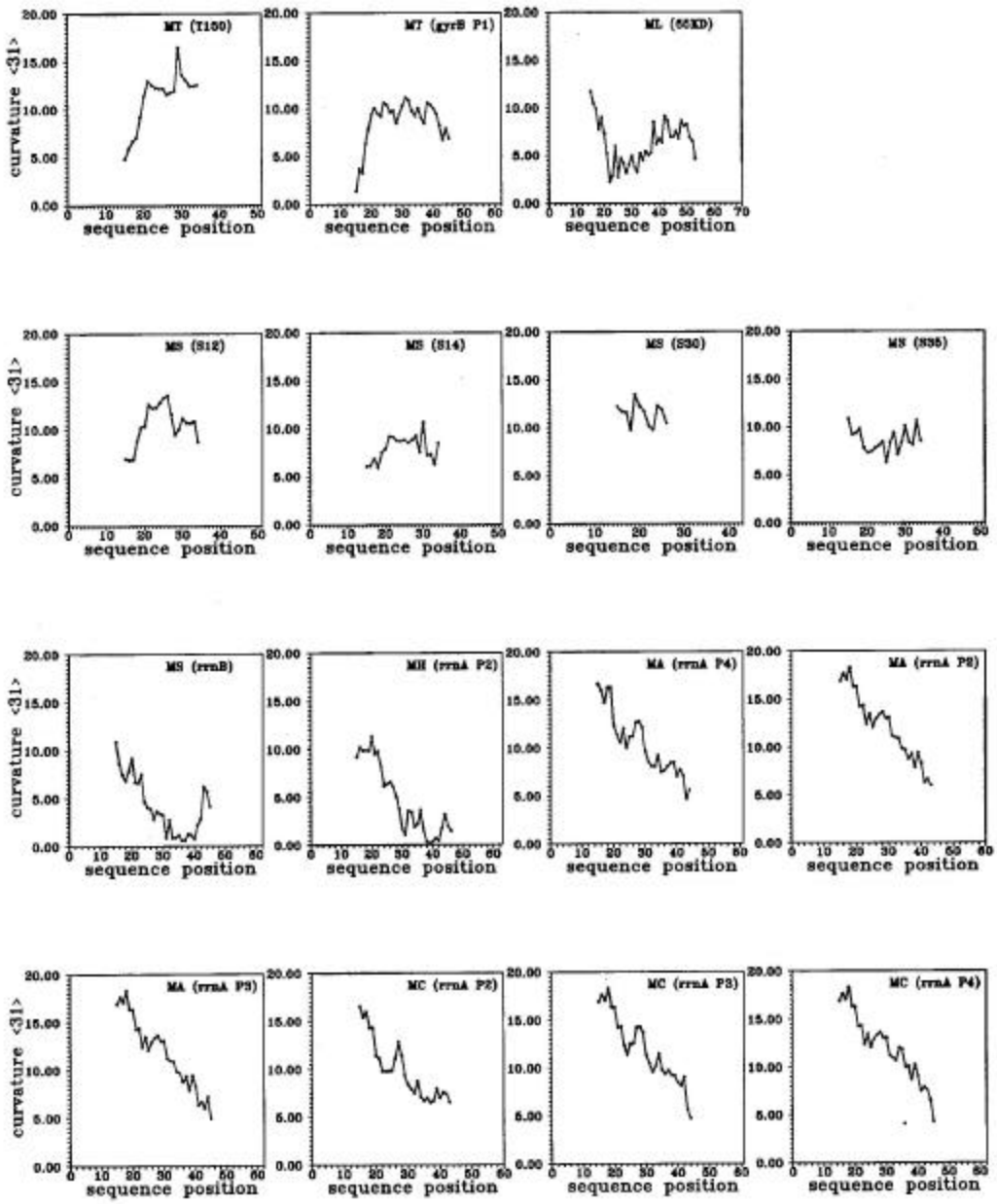


Figure 8-3a: Curvature profiles obtained using energy minimized values of roll and tilt angles for mycobacterial promoters. The curvature is reported as $|C|$, the curvature modulus averaged over 31 bp (*M. smegmatis* S6 is excluded from this plot as grid size used for it is 21 bp).

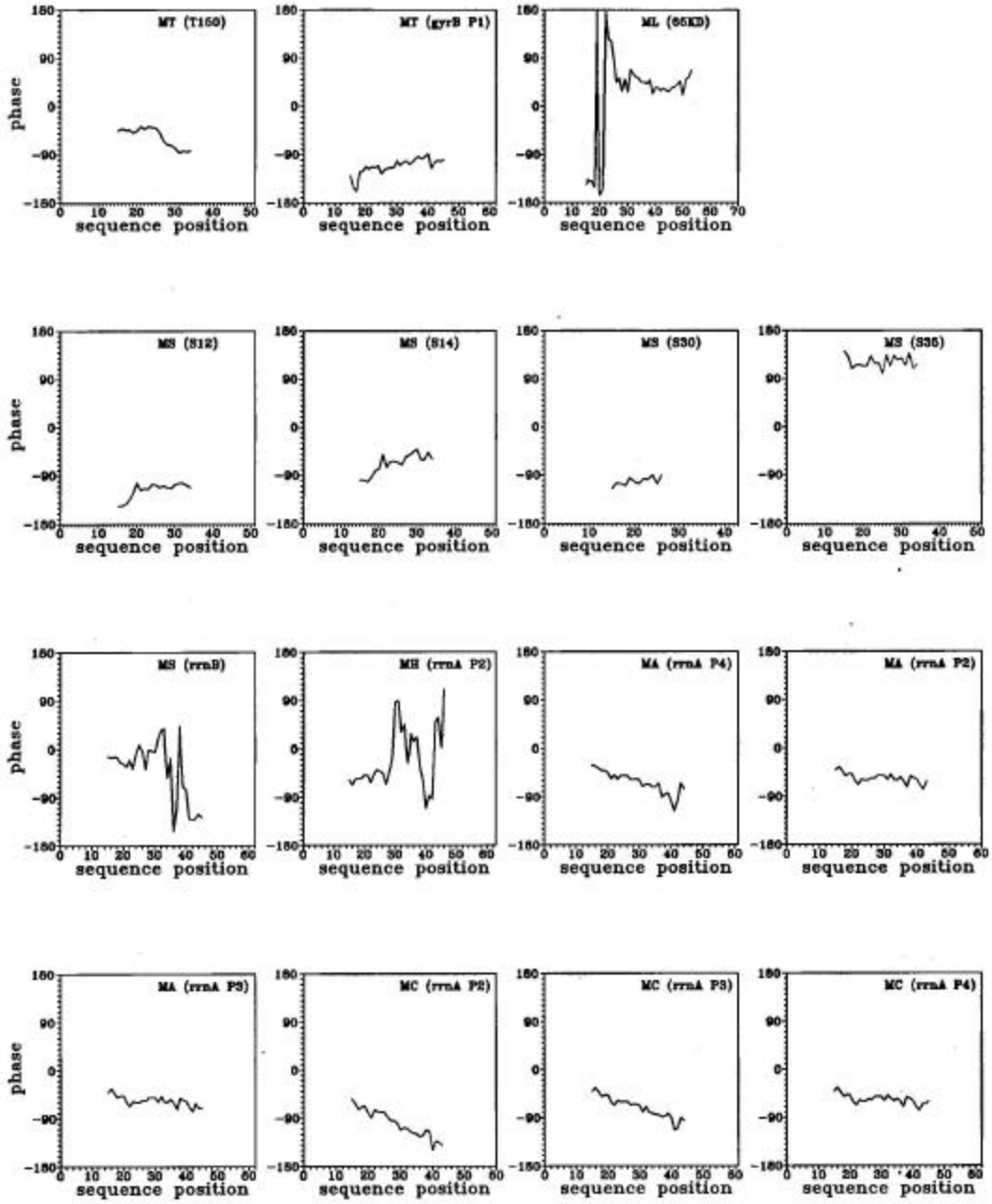


Figure 8-3b: Relative phase profiles of the mycobacterial promoters.

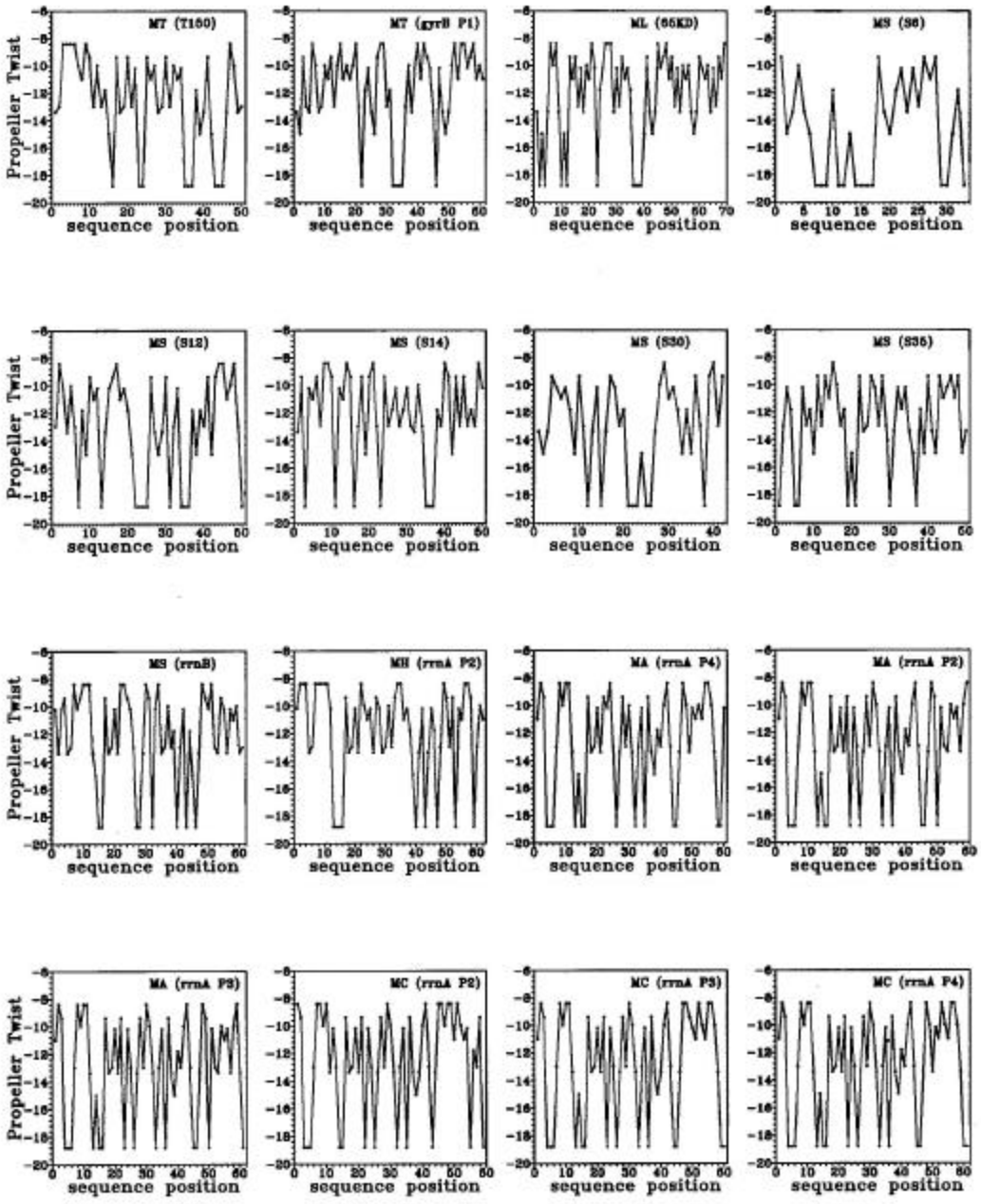


Figure 8-4: Flexibility profile calculated using propeller-twist values obtained from X-ray crystallography of DNA oligomers for mycobacterial promoters.

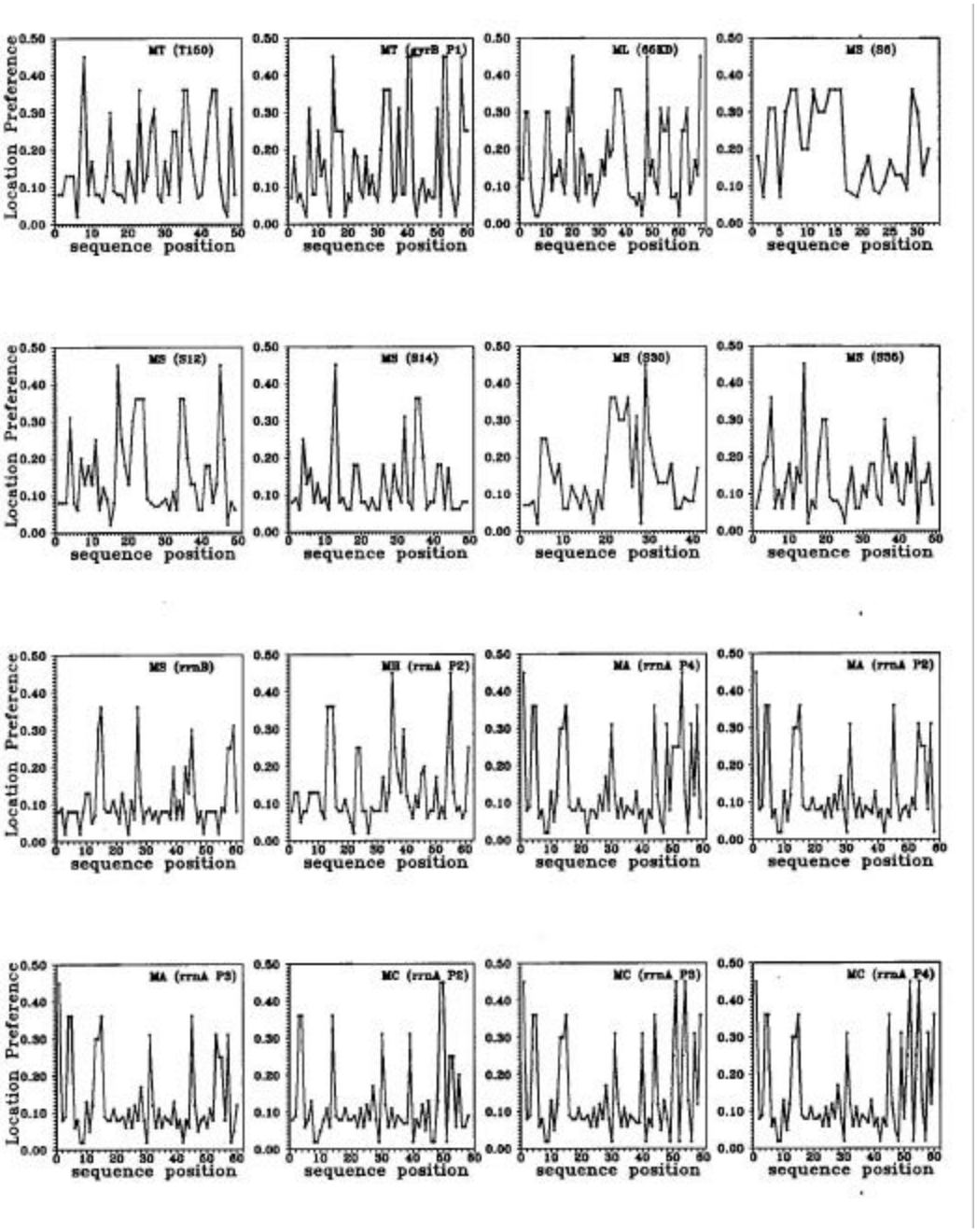


Figure 8-5: Flexibility profile calculated using trinucleotide model based on preferred sequence location on nucleosomes for mycobacterial promoters.

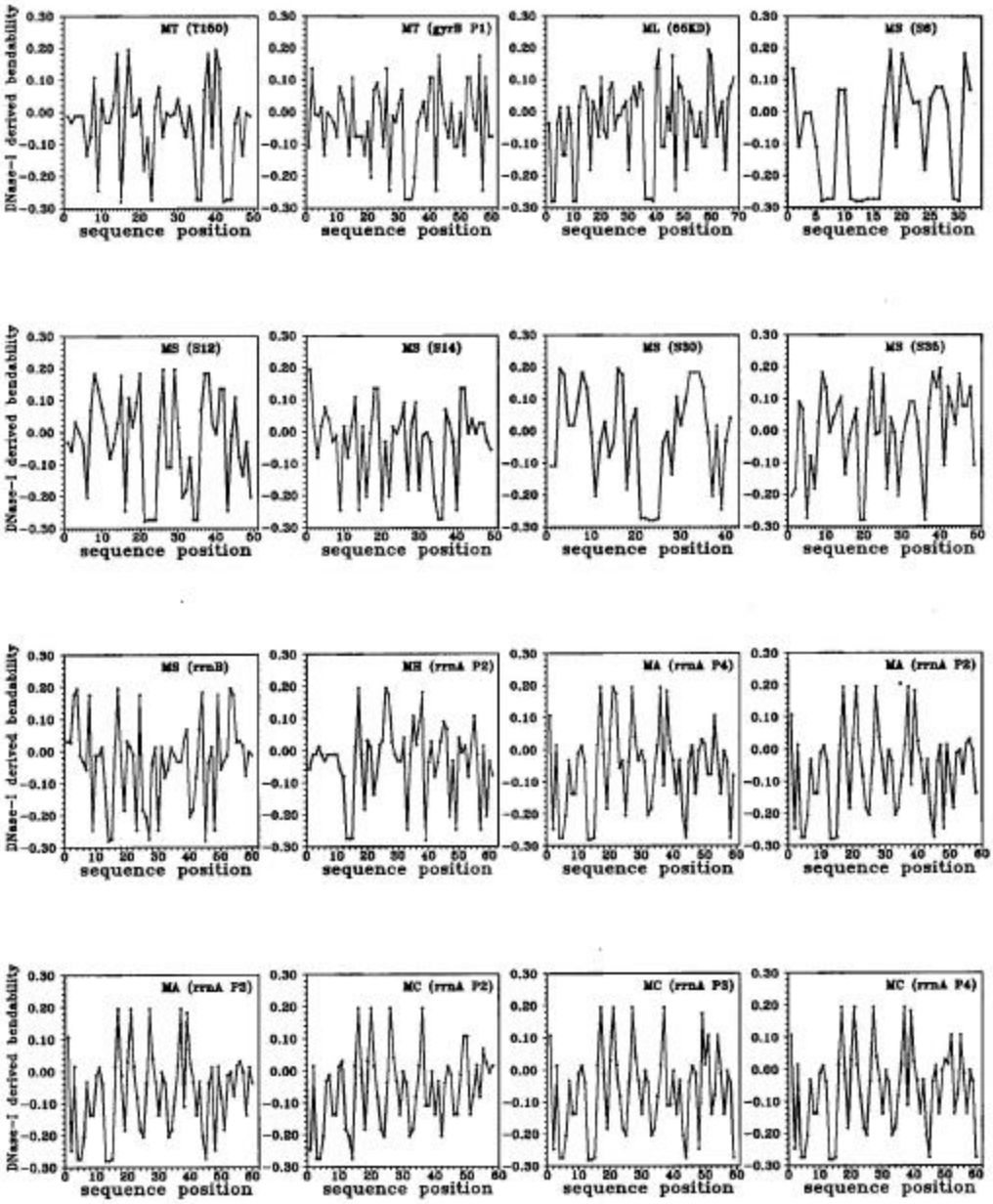


Figure 8-6: Bendability profile calculated using DNase I derived bendability parameters for the mycobacterial promoters.

Table I: Nature of curvature profile for mycobacterial promoters using dinucleotide models based on- i) experimentally determined wedge angles, and ii) energy minimized values of roll and tilt angles

Promoter analyzed	Shipgelman et al.[36]♣	Santis et al. [33]♦	Promoter analyzed	Shipgelman et al. [36]	Santis et al. [33]
M. tuberculosis			*M. tuberculosis*		
T3	Medium	Low	glnA (sp=10)	Medium	Medium
T6	Medium	High	KatG P_A (sp=19)	Low	Low
T26	Low	Low	KatG P_A (sp=15)	Low	Low
T80	Medium	Medium	KatG P_B (sp=20)	Medium	Medium
T101	High	Medium	KatG P_B (sp=22)	Medium	Medium
T119	Low	Low	KatG P_C (sp=22)	Low	Low
T125	Medium	Medium	KatG P_C (sp=14)	Low	Low
T129	Low	Medium	purL	Medium	Medium
T130	Low	Medium	purC	Low	Low
T150	High	very high	groE (sp=19)	Medium	Medium
recA	Medium	Low	groE (sp=11)	Medium	Medium
rrnA P1	Medium	Low	ahpC	Medium	Medium
gyrA	Low	Medium	32 KDa	Medium	Medium
cpn60	Low	Medium	10 Kda (sp=17)	Medium	Medium
gyrB P1	very high	very high	10 Kda (sp=15)	Medium	Medium
gyrB P3	Medium	Medium	10 Kda (sp=8)	Medium	Medium
85A (sp•=17)	Medium	Medium	65 KDa	Medium	Medium
85A (sp=22)	Medium	Medium	mpt 64	Medium	Medium
gyrB P2	Medium	Medium	metA	Medium	High
rrnA PCL1	High	Medium	rpsL	High	Medium
16S rRNA	High	Medium	38 KDa	Medium	Medium
glnA (sp=18)	Medium	Medium	ppgK	Medium	Medium

Table I continued….

Promoter analyzed	Shipgelman et al. [36]	Santis et al. [33]	Promoter analyzed	Shipgelman et al. [36]	Santis et al. [33]
M. bovis BCG			*M. smegmatis*		
hsp60 P2	Medium	Low	S4	Medium	Medium
rRNA	High	Medium	S5	Low	Medium
ahpC	Medium	Medium	S6	High	High*
23 K	Medium	Medium	S12	very high	High
mpb 64	Medium	Medium	S14	High	High
18 K	High	Medium	S16	High	Medium
64 K	Medium	High	S18	Medium	Medium
rpsL	High	Medium	S19	High	Medium
mpb70	High	Medium	S21	Medium	Medium
alpha	High	Medium	S30	High	High
M. leprae			S33	Medium	Low
16S rRNA	Medium	Medium	S35	High	High
18 Kda (sp=17)	High	Medium	S65	Medium	Medium
18 Kda (sp=18)	High	Medium	S69	Medium	Low
28 KDa	Medium	Medium	S119	Low	Low
groE1	Low	Low	gyr B	Low	Low
65 KD	very high	High	recA	Medium	Medium
36 K	Medium	Low	ask	Low	Low
SOD	Medium	Low	acetamidase	Medium	Medium
rpsL	High	Medium	rrn B	High	High
M. smegmatis			rrnA P1	Medium	Medium
alrA	Low	Low	rrnA P2	Medium	Low

Table I continued…..

Promoter analyzed	Shipgelman et al. [36]	Santis et al. [33]	Promoter analyzed	Shipgelman et al. [36]	Santis et al. [33]
M. smegmatis			*M. phlei*		
rrnA P3	Low	Medium	rrnA PCL1	Medium	Medium
rrnA PCL1	Medium	High	rrnA P1	Medium	Medium
rpsL (sp=18)	Medium	Medium	rrnA P2	very high	High
rpsL (sp=17)	Medium	Medium	rrnA P3	Medium	Medium
ahpC	Medium	Medium	*Mycobacteriophage I3*		
M. paratuberculosis			pKGR25	Medium	Medium
pAJB303	Low	Low	pKGR9	Medium	Medium
pAJB86	Medium	Medium	pKGR38	Medium	Medium
pAJB125	Medium	Medium	ORF1	Medium	Low
pAJB300	Medium	Low	ORF2	Medium	High
pAJB305	Medium	Medium	pKGR1	Medium	Medium
pAJB304	Low	Medium	*Mycobacteriophage L5*		
P_{AN}	Low	Medium	71 P2	Medium	Medium
pAJB73	Low	Low	71 P_{left}	Medium	Medium
pAJB301	Medium	Low	71 P1	High	Medium
M. fortuitum			*M. avium*		
repA	Low	Medium	avi-3	Medium	Medium
rrnA PCL1	Medium	Medium	pLR7	Medium	Low
rrnA P1	Medium	Medium	*M. neoaurum*		
rrnA P2a	High	Medium	rrnA PCL1	Medium	Medium
rrnA P2b	Medium	Medium	rrnA P1	Medium	Medium
rrnA P3	Medium	Medium	rrnA P3	Medium	Medium

Table I continued…..

Promoter analyzed	Shipgelman et al. [36]	Santis et al. [33]	Promoter analyzed	Shipgelman et al. [36]	Santis et al. [33]
M. neoaurum			*M. chelonae*		
rrnA P2	High	Medium	rrnA P2	High	very high
M. abscessus			rrnA P1	Medium	Medium
rrnA P4	High	very high	rrnA PCL1	High	Medium
rrnA P1	Medium	Medium	rrnA P3	High	very high
rrnA PCL1	High	Medium	rrnA P4	High	very high
rrnA P2	High	very high	-	-	-
rrnA P3	High	very high	-	-	-

♣ Curvature maxima lying in the range [0.0-0.2], [0.2,0.4], [0.4,0.6]; and [0.6 and above] DNA curvature units is referred to as low, medium, high and very high curvature map, respectively.

♦ Curvature maxima lying in the range [0-5], [5-10], [10-15]; and [15 and above] unit is refereed to as low, medium, high and very high curvature profile, respectively.

• sp denotes spacer length in bp.

* For *M. smegmatis* S6 promoter grid value used is 21 bp while calculating curvature vector (in phase and modulus) by Santis et al. [33]

Table II: Percentage of low, medium and high curvature profiles for various sub-groups of mycobacterial promoters using: i) experimentally determined wedge angles [36]; and ii) energy minimized values of roll and tilt angles [33]

Mycobacterial promoters	Low		Medium		High	
	Shipgelman et al. [36]	P. De Santis et al. [33]	Shipgelman et al. [36]	P. De Santis et al. [33]	Shipgelman et al. [36]	P. De Santis et al. [33]
Class I: E. coli σ^{70} type (sample size=69)	15	19	60	67	25	14
Class II: Non-E. coli σ^{70} type (sample size=36)	22	27	56	54	22	19
Class II: Extended –10 type (sample size=24)	17	4	25	58	58	38
Having optimum (17±1 bp) spacer length (sample size=79)	9	11	61	72	30	17
With high (≥50%) AT content (sample siz=26)	12	15	54	58	35	27
Having A_nT_m (n+m ≥ 3) tract repeated in phase with each other and present at the upstream of –35 box (sample size=12)	17	17	25	33	58	50

Table II continued…

Mycobacterial promoters	Low		Medium		High	
	Shipgelman et al. [36]	P. De Santis et al. [33]	Shipgelman et al. [36]	P. De Santis et al. [33]	Shipgelman et al. [36]	P. De Santis et al. [33]
M. tuberculosis (sample size=44)	25	23	61	68	14	9
M. smegmatis (sample size=28)	21	25	50	50	29	25
Entire compilation (sample size =135)	17	20	57	64	26	16

Table III: Location of molecular bend locus with reference to following sub-regions in the mycobacterial promoter[§] sequence: i) region above –35 box, ii) –35 region, iii) spacer region, iv) –10 region; and v) region below –10 box

Region above –35 box	-35 region	Spacer region	-10 region	Region below –10 box
Promoters whose transcription start site is determined				
MT T180	MT T119	MT T130	MT T101	MT T3
MT recA	MT T125	MT cpn60	MS S14	MT T6
MT 85A ($sp^{\bullet}$=17)	MT T129	MT gyrB P1	MS rpsL (sp=17)	MT T26
MT KatG P_C (sp=22)	MT 85A (sp=22)	MT gyrB P2	MP pAJB86	MT T150
MT purC	MT purL	MT katG P_A (sp=19)	MY 71P2	MT rrnA P1
ML 16S rRNA	MS S4	MT katG P_A (sp=15)	-	MT gyrA
MS S69	MS S5	MT katG P_B (sp=20)	-	MT gyrB P3
MS gyrB	MS S19	MT katG P_B (sp=22)	-	MT rrnA PCL1
MS ask	MS S21	MB hsp60 P2	-	MT 16S rRNA
MS rrnA P1	MS S119	MS S6	-	MT glnA (sp=18)
MS rrnA P2	MS rrnB	MS S12	-	MT glnA (sp=10)
MP pAJB300	MA rrnA P4	MS S16	-	MT KatG P_C (sp=14)
MF rrnA PCL1	MC rrnA P2	MS S18	-	ML 18 kDa (sp=17)
MH rrnAPCL1	MC rrnA P3	MS S30	-	ML 18 kDa (sp=18)
-	-	MS S33	-	MS alrA
-	-	MS S35	-	MS S65
-	-	MP pAJB303	-	MS recA
-	-	MP P_{AN}	-	MS acetamidase
-	-	MF repA	-	MS rrnA P3
-	-	MY 71P1	-	MS rrnA PCL1
-	-	MA rrnA P1	-	MS rpsL (sp=18)

Table III continued…

Region above –35 box	-35 region	Spacer region	-10 region	Region below –10 box
-	-	MA rrnAPCL1	-	MP pAJB125
-	-	MA rrnA P2	-	MP pAJB305
-	-	MA rrnA P3	-	MP pAJB304
-	-	MC rrnA P1	-	MP pAJB73
-	-	MC rrnA PCL1	-	MY 71P_{left}
-	-	-	-	MN rrnAPCL1
-	-	-	-	MC rrnA P4
16%	**16%**	**30%**	**6%**	**32%**
Putative Promoters				
MT 32 kDa	MT ahpC	MT 10 kDa	ML SOD	MT groE
ML 28 kDa	MT metA	MT 38 kDa	MI pKGR25	MT groE
MF rrnA P1	MT rpsL	MT ppgK	MN rrnA P2	MT 10kDa
MN rrnA P1	MB ahpC	MB alpha	-	MT 10kDa
-	MB rpsL	MI pKGR38	-	MT 65kDa
-	ML 65 kDa	MI ORF2	-	MT mpt64
-	ML 36K	MV pLR7	-	MB rRNA
-	ML rpsL	-	-	MB 23K
-	MS ahpC	-	-	MB mpb64
-	MH rrnA P2	-	-	MB 18K
-	MI pKGR9	-	-	MB 64K
-	-	-	-	MB mpb70
-	-	-	-	ML groE1
-	-	-	-	MP pAJB301
-	-	-	-	MF rrnA P2a

Table III continued …

Region above –35 box	-35 region	Spacer region	-10 region	Region below –10 box
-	-	-	-	MF rrnA P2b
-	-	-	-	MF rrnA P3
-	-	-	-	MH rrnA P1
-	-	-	-	MH rrnA P3
-	-	-	-	MI ORF1
-	-	-	-	MI pKGR1
-	-	-	-	MV Avi-3
-	-	-	-	MN rrnA P3
8%	**23%**	**15%**	**6%**	**48%**

♣ MT: *M. tuberculosis*; MB: *M. Bovis BCG*; ML: *M. leprae*; MS: *M. smegmatis*; MP: *M. paratuberculosis*; MF: *M. fortuitum*; MH: *M. phlei*; MI: *Mycobacteriophage I3*; MY: *Mycobacteriophage L5*; MV: *M. avium*; MN: *M. neoaurum*; MA: *M. abscessus*; MC: *M. chelonae*

• sp denotes spacer length in bp.

Biological systems are complex in nature and several known and unknown factors govern their functioning. It is difficult most of the times to interpret underlying relationship(s) between several experimental conditions and corresponding system output(s). Phenomenological modeling of such systems is also difficult due to the inherent complexity of biological systems and inadequate information about them. Thus, it is important to develop and use alternate methods that can be applied to systems with inadequate information. Artificial Intelligence (AI) tools viz. ANN and GA can uncover the underlying relationship(s) of such biological systems.

Detailed understanding of the biosystems require carrying out experiments that are often costly and time consuming. Most of the experiments are also difficult to perform. Due to multilevel interactions, a small change in input parameter of the system may result in changes in large number of features of system. Thus, to have a predictive model that captures the cause and effect relationship is certainly a difficult task. AI tools like ANN and GA can help in building up predictive models and use qualitative and quantitative information about the system. Thus, such modeling can help us in having better understanding of intricate biosystems. Therefore, the primary objective of this thesis is: i) to built up quantitative predictive relationship between inputs and outputs of biosystems wherever possible, and ii) in instances where such predictive quantitative relationship can not be built due to gross inadequacy of input-output data, it is hoped that they would at least provide qualitative guidelines for narrowing the choice of experiments to be performed.

It is with this view that in chapter 2, we develop an ANN model to establish a correlation between a nucleotide sequence of DNA and its effective curvature, characterized in terms of retardation anomaly (R_L) value. An ANN capturing the role of phasing, increased helix flexibility, run of polyA tracts, and flanking base pair effects in determining the extent of curvature has been developed. The results suggest that ANN can be used as a model-free tool for studying DNA curvature. In chapter 3 for ANN – based modeling of DNA sequences, two new input coding strategies namely, the *wedge* and the

twist code have been suggested. The performance of the proposed strategies has been tested by performing various case studies. The proposed coding schemes have been shown to outperform the existing coding strategies especially in situations wherein limited data are available for building the ANN models. Chapter 4, presents a hybrid strategy involving an ANN and a GA for the optimization of a biologically important feature or property. This strategy is general and is illustrated using an example of optimization of DNA curvature. The ANN-GA technique is a useful tool to obtain, ahead of experimentation, sequences that yield high R_L values. Chapter 5 illustrates a hybrid non-linear strategy involving an ANN and GA for optimization of transcription efficiency in eukaryotic systems using β-globin gene as a case example. The study reveals that multiple base substitutions in the conserved as well as non-conserved regions can cause substantial enhancements in the RTL. We identify positions in the nucleotide sequences, which preferable should not be altered, as well as those positions where mutations can lead to increased RTL. The study helps to obtain an insight into the structural aspects of β-globin gene leading to high transcription efficiency.

Chapter 6 of the thesis provides a compilation of different mycobacterial promoters and analysis of their DNA sequences for various features. Further, the study suggests show a broad classification of these promoters into three main types viz., i) *E. coli* type, *Non-E. coli* type, and iii) Extended –10 promoters. In chapter 7, an ANN model is developed for classifying mycobacterial promoter sequences from non-promoter sequences. Calliper randomization approach has been suggested for determining structurally and functionally important regions within the mycobacterial promoter sequences. Chapter 8 presents theoretical analysis of DNA curvature for mycobacterial promoters using several di- and trinucleotide dependent models of DNA curvature. Various theoretical studies on mycobacterial promoters throw some light on the mycobacterial transcription machinery and structure of mycobacterial promotes. Such studies are an important step towards understanding low levels of transcription and the possible mechanisms of regulation of gene expression.

In essence, the thesis aims at building predictive relationships using AI tools for complex biological systems with a view to model and analyze DNA sequences for their properties and biological roles. This continues to be a poorly understood area and it is hoped that the approach adopted in the thesis takes a step forward in resolving the issues.

LIST OF RESEARCH PUBLICATIONS

1. **Rupali V. Parbhane**, S.S. Tambe, B.D. Kulkarni (1998) Analysis of DNA curvature using artificial neural networks. *Bioinformatics*, **14**, 131-138.
2. **Rupali V. Parbhane**, S.S. Tambe, B.D. Kulkarni (2000) ANN modeling DNA sequences: new strategies using DNA shape code. *Comput. & Chem.*, **24**, 699-711.
3. **Rupali V. Parbhane**, S. Unniraman, S.S. Tambe, V. Nagaraja, B.D. Kulkarni (2000) Optimum DNA Curvature Using a Hybrid Approach Involving an Artificial Neural Networks and Genetic Algorithm. *J Biomol Struct Dyn*, **17**, 665-672.
4. **Rupali V. Parbhane**, S.S. Tambe, B.D. Kulkarni (2000) Optimizing transcription efficiency in eukaryotic systems using a hybrid approach involving an Artificial Neural Networks and Genetic Algorithms: a case study of β-globin gene (submitted).
5. **Rupali V. Parbhane**, V. Nagaraja, B.D. Kulkarni (2000) Compilation and Analysis of Mycobacterial Promoters (in preparation).
6. **Rupali V. Parbhane**, B.D. Kulkarni (2000) Analysis of DNA curvature distribution within Mycobacterial Promoters (in preparation).
7. **Rupali V. Parbhane**, S.S. Tambe, B.D. Kulkarni (2000) Analysis of Mycobacterial Promoters Using Artificial Neural Networks: Calliper Randomization Approach in Determining Functionally Important Region (in preparation).