HUMAN PROMOTER RECOGNITION BASED ON PRINCIPAL COMPONENT ANALYSIS

XIAOMENG LI

A thesis submitted in fulfillment

of the requirements for the degree of

Master by Research

School of Electrical and Information Engineering

The University of Sydney

March 2008

Acknowledgement

This thesis would not have been possible without the assistance and support of many people. Firstly, I would like to thank my supervisor Professor Hong Yan for offering me this thesis topic and then supporting and guiding me throughout my research. I would also like to thank my colleagues in the Image Processing Laboratory at the University of Sydney and in the Signal Processing Laboratory at City University of Hong Kong: Inge Rogers, Miew Keen Choong, Qinzhi Zhang, Jia Zeng and Hongya Zhao for their friendship and assistance.

I would also like to acknowledge the invaluable support and encouragement supplied by my family and friends. In particular I would like to thank my parents for their financial support during my studies in Australia. And Zuoliang Ning, my husband, who is always concerned about my life and gives me technical support for my studies. I great appreciate your support. I would also like to thank Meixing Liu, Fengyao Yu, Honglan Xu, Sue Chen, Peter Wang, He'ai Ding and Qingshui Wang, for their friendship and thoughtful concern for my daily life in Australia.

Abstract

This thesis presents an innovative human promoter recognition model HPR-PCA. Principal component analysis (PCA) is applied on context feature selection DNA sequences and the prediction network is built with the artificial neural network (ANN). A thorough literature review of all the relevant topics in the promoter prediction field is also provided.

As the main technique of HPR-PCA, the application of PCA on feature selection is firstly developed. In order to find informative and discriminative features for effective classification, PCA is applied on the different n-mer promoter and exon combined frequency matrices, and principal components (PCs) of each matrix are generated to construct the new feature space. ANN built classifiers are used to test the discriminability of each feature space. Finally, the 3 and 5-mer feature matrix is selected as the context feature in this model.

Two proposed schemes of HPR-PCA model are discussed and the implementations of sub-modules in each scheme are introduced. The context features selected by PCA are

used to build three promoter and non-promoter classifiers. CpG-island modules are embedded into models in different ways. In the comparison, Scheme I obtains better prediction results on two test sets so it is adopted as the model for HPR-PCA for further evaluation. Three existing promoter prediction systems are used to compare to HPR-PCA on three test sets including the chromosome 22 sequence. The performance of HPR-PCA is outstanding compared to the other four systems.

Table of Contents

Table of Figures

Table of Tables

Chapter 1 Introduction

One of the most important goals of Human Genome Project [Lander, Linton et al. 2001] is to provide a complete list of annotated genes. Although large scale sequencing projects of human complete mRNAs have been undertaken, there are still many low copy genes which evade sequencing [Sonnenburg, Zien et al. 2006]. Bioinformatics, which is a combination of biology, biochemistry, mathematics and computer science [Chen(Ed.) 2005], makes it possible to identify those genes using computer-based technologies. As the core content of this thesis, detecting promoter regions which are close to the transcription start sites (TSS), is one of the most important aspects of DNA sequence analysis. Promoter regions have significant value for the human genome project, because once a promoter is found, a gene start can be annotated.

In this chapter, we firstly introduce the basic concepts of transcription and eukaryotic promoters. Secondly, the significance of promoter prediction is stated. Finally, the outline of the thesis is presented. The contribution of the thesis is towards finding effective features to improve the performance of the promoter recognition system.

1.1 Transcription and Eukaryotic Promoter

Transcription is the process in which a DNA sequence is converted to a corresponding RNA sequence. A DNA sequence contains four types of nucleotides: adenine (abbreviated A), cytosine (C), guanine (G) and thymine (T). DNA usually occurs in double strands and the nucleotides in one strand are complementary to the ones in the other strand, i.e., A bonding only to T and C bonding only to G. The transcription start site (TSS) is the position where protein synthesis begins [Attwood and Parry-Smith 1999]. When transcription starts, RNA is produced by copying the genetic information from the DNA sequence. If the RNA carries coding information to the protein synthesis sites, we call it mRNA. In this process, the nucleotides A, C, G, and T of the DNA sequence are transcribed into U (uracil, which replaces T), G, C and A of the mRNA sequence. Eventually, translation starts and protein is synthesized corresponding to mRNA [Chen(Ed.) 2005].

Promoter regions are located upstream of the TSS of genes. They contain binding sites that can be recognized by transcription factors that are a kind of protein. Acting as a "switch", promoter sequences specify the times and the places of the transcription occurring in genes. They attract transcription factors that give RNA polymerase access to gene and permit RNA transcription. Other activating and repressing proteins also bond to promoter regions and affect gene expression depending on different cell conditions [Deonier, Tavare et al. 2005]. Figure.1.1 shows the structure of a eukaryotic gene sequence and the location of the promoter in this DNA gene sequence

is marked by a black box.

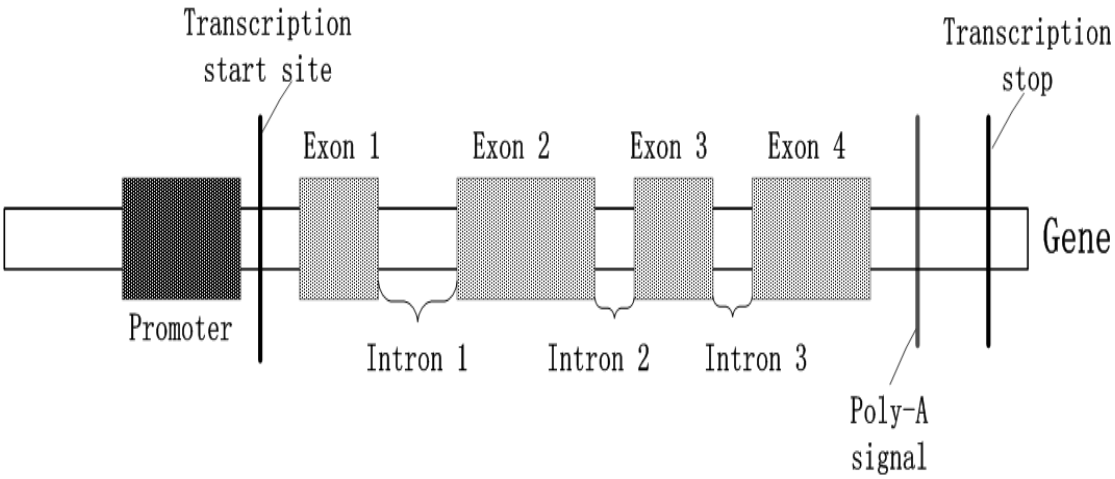

Figure 1.1 The structure of a eukaryotic gene sequence

The black box indicates the promoter, which is close to the transcription start site. Grey boxes indicate exons, which are separated by introns. In the transcription of eukaryotic gene systems, exons form a part of the final protein-coding sequence, but introns are cut out of the transcript. The transcription start site and the transcription stop are indicated by solid black lines. The lighter black solid line before the transcription stop corresponds to the Poly-A signal site, where the transcript is cleaved [Raychaudhuri 2006].

1.2 Significance of Promoter Prediction

The protein coding gene is one of the most important features of the genomic DNA sequence. However, compared to non-coding DNA sequences, coding regions only occupy a tiny part of whole DNA sequences. So for a further understanding of how the

coding regions are regulated, the first and necessary step is to extract the small islands of coding regions from the sea of non-coding sequence. Only by using biochemical methodology, can researchers locate the exact position of TSS, where the transcription starts. However, the processes are complicated and the time spent on the massive sequence database cannot be estimated. Therefore it is proposed that computational analysis of eukaryotic promoters be first employed to improve gene identification and prediction efficiency. Promoter regions contain the binding sites for transcription factors and other proteins which control the transcription. By predicting the promoter regions, we can approximate the location of TSS, which is located in, or immediately downstream, of promoter regions, and then find the gene start.

Owing to the development of the human genome project, almost entire coding sequences of human beings have been mapped. However, there are still many genes which have not been sequenced, meaning that these genes and their promoter regions may be ignored. Promoter identification in other species also needs to be accomplished, but unlike promoter recognition on the human genome, the experiment is without the support of massive sequencing [Sonnenburg, Zien et al. 2006]. Therefore, to develop an effective computational promoter system with high specificity and sensitivity is crucial for solving these problems.

Last but not least, promoter regions are elaborate mechanisms that control specific genes in a highly defined manner both spatially and temporally. Cell types are then determined in multicellular organisms by specifically turning on or off the

transcription of sets of genes. Moreover, human health is also influenced by transcription control, since improper regulation of genes in cell growth may cause diseases such as cancer [Pedersen, Baldi et al. 1999]. Therefore, promoter prediction is definitely interesting of its own function.

1.3 Outline of this Thesis

In this thesis, a human promoter prediction system — HPR-PCA (human promoter recognition system base on principal component analysis) is explored. A new proposal regarding promoter feature selection is developed and a principal component analysis algorithm is adopted in the process. Most discriminative features are selected to form feature matrices. The classification and promoter prediction modules are implemented by artificial neural networks (ANN).

The contributions of the thesis include:

Codons and pentamers of DNA sequences are firstly proposed to be combined as candidates for feature selection. Comparison results testify to the efficiency of the combination.

PCA is applied in feature selection process. Rather than specific pentamers such as CGGCG, GCGCG etc. mentioned in PromoterExplore [Xie, Wu et al. 2006], more complex hybrid feature vectors are built, based on simple features which are extracted directly from training promoter sequences.

ANN is used in both training and test processes. Optimal network parameters are

received by the comparative experiments and the holdout validation training approach. The modeling network is computationally efficient.

Both Specificity and Sensitivity are improved compared to three of the most popular promoter prediction systems: DragonGSF [Bajic and Seah 2003] [Bajic, Tan et al. 2004], FirstEF [Davuluri, Grosse et al. 2001] and Eponine [Down and Hubbard 2002]. The better performance testifies to the effectiveness of the new feature selection methodology based on PCA.

The thesis is organized as follows:

In Chapter 2, the concepts and methodologies involved in the promoter prediction approach are reviewed. Important features of promoters are classified into context features, and the corresponding algorithms of feature extraction and feature selection are introduced. Several state-of-the-art modeling and classification technologies are stated. Summaries of sequence features and modeling approaches used in existing promoter prediction systems are listed.

In Chapter 3, PCA is taken as the feature selection algorithm in HPR-PCA, so the theory and application of PCA are presented first. As the implementation tool, the theory of backpropagation (BP) ANN is also introduced. The steps contained in the feature selection process of HPC-PCA are stated specifically and the implementation of PCA on feature selection is emphasized in this section. Finally a codon and pentamer combined frequency matrix is selected as original input matrices of the PCA

feature selection algorithm.

In Chapter 4, two promoter prediction schemes are proposed, whose structures and training methods are different. The sub-modules' implementations of each scheme are the same and stated in this section. The experiment for building classifiers including parameter optimization is emphasized. Finally, promoter and non-promoter sequences and three human DNA sequences are used for testing the performance of the two proposed schemes. In this comparison, the scheme that achieves better results is adopted as the HPR-PCA model.

In Chapter 5, three test sets are formed to evaluate the performance of HPR-PCA and different evaluation criterions are introduced. Three other state-of-the-art promoter prediction models— DragonGSF, FirstEF and Eponine are chosen for comparison. The characteristics of the four systems are analyzed according to test results.

In Chapter 6, the advantages and shortages of HPR-PCA are pointed out. Finally, future work for improving the performance of HPR-PCA is proposed.

Chapter 2 Literature Review

This chapter presents a review of the literature in the promoter prediction area. In particular the following topics are reviewed: important promoter features, promoter feature extraction and algorithms and promoter prediction modeling techniques. In order to obtain a clear understanding of the feature extraction and selection processes and the whole prediction systems, we firstly introduce important promoter features, which are the basis of building classifiers for the promoter recognition system. Two distinct types of promoter features: signal features and context features are discussed. A summary of features used in promoter prediction systems is listed. Next, the theoretical background of several typical feature extraction and selection algorithms is introduced. Fianlly, several state-of-the-art modeling and classification methodologies are presented with a summary of the specific models used in some well-known existing promoter prediction systems.

2.1 Important Promoter Features

To accurately predict promoter regions, finding discriminative and informative

features is the first and key step. As far as feature choice is concerned, there are two distinct types of features used in the area of promoter prediction: signal and context structure features. All of the promoter prediction systems adopt one or more kinds of features as these informative features are important for building a powerful classifier. Therefore, it is necessary to present the definition and characteristics of these promoters.

2.1.1 Sequence Signal Features

Signal features are biological signals in promoter regions. The most important signal features include CpG islands, transcription factor binding sites (TFBSs) such as TATA-box and CAAT-box, and initiator (Inr). These are the most intuitive features and they all have clear definition. In this section, the concept of CpG islands and TATA-box are discussed as they are the most frequently used features in promoter prediction systems.

CpG islands

CpG islands are genomic regions that contain a high frequency of CG dinucleotides and relate to 56% of human genes [Antequera and Bird 1993]. The letter "p" in CpG notation refers to the phosphodiester bond between the cytosine and the guanine. We use two features to identify whether the sequence (>200 base pair (bp)) is CpG islands related:

(1) GC percentage (GCp)

$$GCp = P(C) + P(G), \tag{2.1}$$

(2) Observed/expected CpG ratio (o/e)

$$o/e = \frac{P(CG)}{P(C) \times P(G)}, \tag{2.2}$$

where $$P(C) = \frac{number\ of\ Cs}{length} \tag{2.3}$$

$$P(G) = \frac{number\ of\ Gs}{length} \tag{2.4}$$

$$P(CG) = \frac{number\ of\ CGs}{length} \tag{2.5}$$

If $GCp > 0.5$, and $o/e > 0.6$, the sequence is CpG islands related, otherwise it is non-CpG islands related [Gardiner-Garden and Frommer 1987]. Many promoter prediction systems, such as CpGProd [Ponger and Mouchiroud 2002], FirstEF, PromoterExplore, use CpG islands as a global signal feature.

TATA-box

The TATA-box is a DNA sequence that has a consensus TATA(A/T)A(A/T) sequence. TATA-box is usually located at 25bp upstream to the transcription site [Fickett and Hatzigeorgiou 1997] and it is normally bound by the TATA Binding Protein: TFIID/ TFIIA, TFIIB, RNA polymerase II/TFIIF, TFIIE and TFIIH [Smale and Kadonaga 2003]. TATA-box is used in many systems, i.e. PWMs [Bucher 1990], NNPP [Reese 2001], Promoterscan [Prestridge 1995] as a promoter sequence signal like

CpG-islands. However, compared to a long genome sequence, the TFBSs like TATA-boxes are too short, which means that similar elements may be found by chance anywhere [Pedersen, Pierre Baldi et al. 1998]. Furthermore, statistical analysis also indicates that the density of TFBSs does not contain information to effectively classify promoters and non-promoters [Zhang 1998a]. Therefore, we need to combine other features with TFBSs like TATA-boxes for promoter and non-promoter classification.

Initiator (Inr)

Inr elements are usually characterized by the consensus sequence PyPyAN(T/A)PyPy [Kaufmann, Verrijzer et al. 1996], where Py is a pyrimidine(C or T), and N is any nucleotide [Bucher 1990]. The first A is located at the transcription start site and the pyrimidine just upstream of this A is often cytosine. It is found that in some TATA-box less promoters, Inr may control the transcriptional start point instead of TATA-box. There are also promoters that have both TATA-box and Inr elements, and promoters without either [Pedersen, Baldi et al. 1999]. Therefore, Inr is usually combined with the TATA-box as the signal feature in earlier promoter recognition methods.

2.1.2 Sequence Context Features

Unlike signal features, context features are basically extracted from training genomic sequences and analyzed by statistical methods. The word "n-mers" can cover all of the

context features. Codons (3-mer) are used as the genetic code and they can be translated into 20 distinct amino acids (refer to Table 2.1). A coding region always begins with a start codon and ends with a stop codon. ATG is usually regarded as an initial codon, and TGA, TAA or TAG are usually known as stop codons. It is found that codon patterns in coding and non-coding regions are different [Attwood and Parry-Smith 1999], thus it is supposed that codon-usage statistics can be used to analyze context features of promoters. Context features have been used in many well-known promoter prediction models recently: PromoteExplore [Xie, Wu et al. 2006] selects informative pentamers (5-mer) as context features; DragonGSF [Bajic and Seah 2003] generates the positional weight matrix (PWM) of pentamers to calculate scores of fixed length sequence. The Kullback-Leibler divergence [Wu, Xie et al. 2007] based classifier chooses the hexamers (6-mer) as promoter features to balance the discriminant power of classifiers and the computational speed. Additionally, it has also been shown that 8-mers have distinct pattern relative to TSS [FitzGeralad, Shlyakhtenko et al. 2004]. The great power of sequence context features is testified to by the improved performance of these new promoter models.

Table 2.1 The genetic code of codons [Attwood and Parry-Smith 1999]

	T		C		A		G		
T	TTT TTC	Phe	TCT TCC	Ser	TAT TAC	Try	TGT TGC	Cys	T C
	TTA	Leu	TCA		TAA	Stop	TGA	Stop	A

	TTG		TCG		TAG		TGG	Trp	G
C	CCT	Leu	CCT	Pro	CAT	His	CGT	Arg	T
	CTC		CCC		CAC		CGC		C
	CTA		CCA		CAA	Gln	CGA		A
	CTG		CCG		CAG		CGG		G
A	ATT	Ile	ACT	Thr	AAT	Asn	AGT	Ser	T
	ATC		ACC		AAC		AGC		C
	ATA		ACA		AAA	Lys	AGA	Arg	A
	ATG	Met	ACG		AAG		AGG		G
G	GTT	Val	GCT	Ala	GAT	Asp	GGT	Gly	T
	GTC		GCC		GAC		GGC		C
	CTA		GCA		GAA	Glu	GGA		A
	GTG		GCG		GAG		GGG		G

2.1.3 Summary of Features Used in Existing Promoter Prediction Models

Sequence context features theoretically include signal features, because signal features are also the combination of nucleotides. For example, in the evaluation equation of CpG islands, the frequency of "CG" (a typical 2-mer) is counted, and TATA sequence is defined as a short consensus sequence TATA(A/T)A(A/T). Many promoter prediction methods locate the promoter regions by purely using those signal

features as biological signals are the most reliable features of promoter sequences. However, the statistical characteristics of DNA sequences are also important because it is possible to dig out the relationship among the massive nucleotides sequence datasets. Therefore, it is advisable to combine signal features and context features to construct the classifier for promoter recognition. A summary of features that have been used in existing promoter prediction methods is shown in Table 2.2.

Table 2.2 Summary of features used in existing promoter prediction systems

System	Signal Feature				Context Feature
	TATA-box (Y /N)	Inr (Y/ N)	GC-box or CpG Island (Y or N)	Other	
PWMs	Y	Y	Y	Cap signal CAAT-box	N/A
Promoter-Scan	Y	N	N	TEs	N/A
PromFD	Y	Y	N	N	5-mer–10-mer
Promoter-Inspector	N	N	N	N	IUPAC words
NNPP	Y	Y	N	N	N/A

FirstEF	N	N	Y	first splice donor site	5-mer 6-mer
DFP	N	N	Y	N/A	5-mer
DragonGSF	N	N	Y	N/A	5-mer
CpGProD	N	N	Y	N/A	N/A
Eponine	Y	N	Y	N/A	N/A
McPromer	Y	Y	N	N/A	motifs
Prom-Predictor	N	N	Y	N/A	5-mer
Promoter-Explorer	N	N	Y	N	5-mer
KLD	N	N	N	N	5-mer–7-mer

2.2 Review of Promoter Feature Extraction and Selection Algorithms

The selection of discriminative input features is the crucial step in building a powerful classifier. Varieties of algorithms for signal processing and pattern recognition have been used in promoter feature extraction and selection processes, as DNA sequences also can be regarded as digital signals. In this section, several classical and popular feature extraction and selection algorithms are reviewed.

2.2.1 Sequence Feature Extraction Algorithms

The positional weight matrix (PWM) and the hidden Markov model (HMM) are two of the most classical and useful algorithms in sequence feature extraction. These are always essential to sequence signal features. Therefore, the theoretical background of these two algorithms and their application in sequence feature extraction are presented in the following section

Positional weight matrix

A weight matrix is a simple generative model for a short, ungapped sequence motif [Down and Hubbard 2002]. PWM is used extensively in signal feature extraction processing, as it can create a profile that represents the common feature across the training sequence. This profile can be used to scan new sequences and make a decision as to whether these sequences are related to the training group [Raychaudhuri 2006]. The simple process of generating a PWM is stated as follows:

Firstly, given a group of fixed length sequences, we total up the number of each nucleotide in each position.

Secondly, we calculate the probability of each nucleotide at each position by the following equation:

$$P_{n,i} = \frac{N_{n,i}}{N} \tag{2.6}$$

where $P_{n,i}$ is the probability of a specific nucleotide n (A, C, T, G) in position i of a sequence, and N_i is the number of times the nucleotide occurs at the position

among the whole observed sequence. N is the total number of sequences. Sometimes a bias factor might be added to this equation to avoid zero probability and the equation can be changed to:

$$P_{n,i} = \frac{N_{n,i} + q_n}{N + 1} \tag{2.7}$$

Here, q_n is the background frequency of a nucleotide and a pseudo-count of one is used here.

Next, we calculate the values of the weight matrix as follows:

$$W_{n,i} = \log\left(\frac{P_{n,i}}{q_n}\right) \tag{2.8}$$

where $W_{n,i}$ is the value of each nucleotide in position i. At last, the score of a new sequence is derived by finding the weight values in the matrix that correspond to the nucleotide at a specific position and their totals:

$$S = \sum_{i=1}^{N} W_{n,i} \tag{2.9}$$

The above process is a simplified version. PWM is also used to generate position matrices of n-mers. In order to satisfy different models, some promoter prediction systems that use PWM to extract the signal features modify the equation by adding some parameters and conditions. PWMs [Bucher 1990] derives four weight matrices of TATA-box, cap signal, CCAAT-box and GC-box respectively. PromoterScan [Prestridge 1995] uses a weight matix to score TATA-box. Eponine combines weight matrices with associated discrete probability distributions relative to TSS to generate more complex weight matrices of TATA-box and CpG-island enrichment. DPF [Bajic, Chong et al. 2002] [Bajic, Seah et al. 2003] and PromoterExplorer [Xie, Wu et al.

2006] derive their position weight matrices from selected pentamers.

Hidden Markov Model (HMM)

Hidden Markov Model (HMM) [Krogh and Brown 1994] is a more sophisticated technology for feature extraction from sequences compared to PWM. A sequence of coin flips is always used to illustrate a hidden Markov model (Figure 2.1). We can assume that a fair coin can derive flips of head (H) and tail (T) with equal probability of 0.5, and a biased coin derives probability of H and T for p and $1-p$ respectively. The arrows indicate state transitions, by which the fair coin can switch to the biased one with probability q and the biased coin can switch to the fair one with probability q'. There are two states in this model. $S1$ represents one state in which the flips are generated by the fair coin and $S2$ is the other state in which flips are generated by the biased coin [Raychaudhuri 2006].

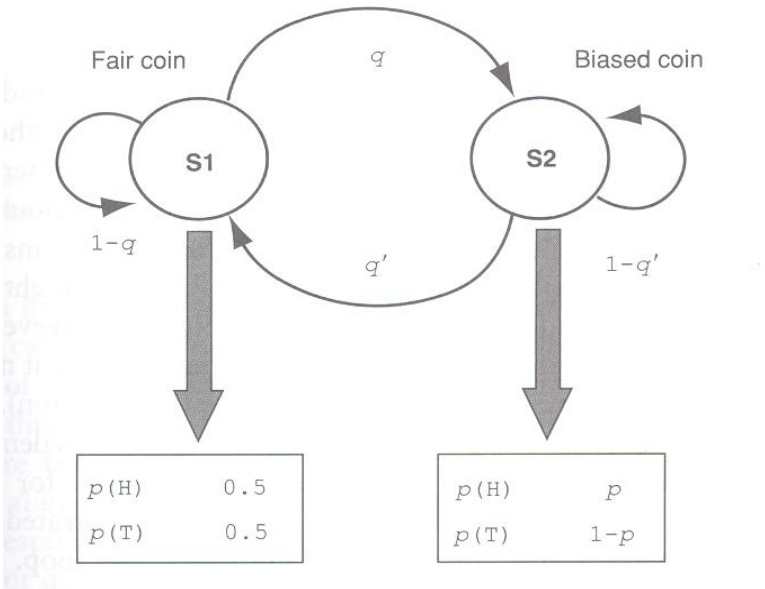

Figure 2.1 A simple hidden Markov model with two states [Raychaudhuri 2006]

When applying the HMM on sequence analysis, we assume that observations (nucleotides in DNA) are accounted for by hidden states and the probability of the current state is only dependent on the prior state of the same system. HMM can represent spacer-included motifs [Murakami, Ohta et al. 2000] of a sequence family. There are a number of algorithms that can be used for training HMM and a simple example to train the HMM by Baum-Welsh algorithm is as follows:

Firstly, the random parameters are assigned to the model.

Secondly, the probability of each state for each position of each sequence is calculated.

Lastly, the parameters are updated by the training algorithm and then the model with optimized parameters can describe this family of sequences [Raychaudhuri 2006].

The classical model has been utilized by different systems. Generalized Hidden Markov Model (GHMM) [Stormo and Haussler 1994] is used for generating multi-symbol strings in gene finding systems [Kulp, Haussler et al. 1996]. The Pol II promoter prediction program [Murakami, Ohta et al. 2000] is built based on PromFD [Chen, Hertz et al. 1997] and utilizes HMM to acquire additional motifs. McPromoter is developed based on GenScan [Burge and Karlin 1997], and uses stochastic segment models (SSMs) [Ostendorf, Digalakis et al. 1995] which is a generalization of HMM to represent six segments of the promoter sequence from -250 to +50bp: upstream 1

and 2, TATA box, spacer, initiator and downstream.

2.2.2 Sequence Feature Selection Algorithms

Compared to sequence signal features, context features are more complex. From 3-mer (codon) to 8-mer, the number of variables increases exponentially. Sometimes, a better promoter prediction result relies on finding more discriminative features rather than improving model building methodology. Therefore, many systems start to focus on how to select the most effective and informative features among the massive feature-candidate pool. Here I simply present four methodologies in context feature selection process used by different systems.

DPF [Bajic, Chong et al. 2002] selects 256 from 1024 pentamers p_j ($j = 1, 2, ...1024$) for the classification model using the relevance function:

$$J = \left(\mu_p - \mu_n\right) / \left(\lambda_p + \lambda_n + 1\right) \tag{2.10}$$

where μ_p and μ_n are the percentages of promoters and non-promoters in which the pentamer p_j appears respectively. The number of λ_p and λ_n represents the p_j's average number of occurrences in sequences where p_j appears in promoter and non-promoter sequence training sets. Finally, 256 pentamers with the highest values received from the relevance function are selected for the classification model.

The Pentamer selection method used by PromPredictor [Chen and Li 2005] is also

based on calculating the relevance of the same features between different data sets. It refers to the distance function [Solovyev and Makarova 1993]:

$$D(X) = \frac{(m_1 - m_0)^2}{d_1^2 + d_0^2} \tag{2.11}$$

where m_1 and m_0 are the mean value of the feature X in the promoter and non-promoter sequence data sets respectively. d_1 and d_0 are the standard deviations of feature X in positive and negative training sequence data sets separately. They can be calculated by the following function:

$$m = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{2.12}$$

$$d = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - m)^2} \tag{2.13}$$

where x_i is the value of feature X appearing in sequence S_i and N is the total number of sample sequences in a data set. Subsequently, the top N_a pentamers ($N_a < 1024$) with the largest values of $D(X)$ are selected.

PromoterExplorer involves posterior probability in context feature selection. A function for selecting most informative pentamers is defined as:

$$\eta = \frac{P(I=1 \mid a_i)}{P(I=0 \mid a_i)}, \qquad i = 1, 2, \cdots 1024 \tag{2.14}$$

where $P(I=1 \mid a_i)$ is the posterior probability of I given a pentamer a_i. $I=1$ representing the input sequence is the promoter, otherwise $I=0$. The pentamers are ranked according to their η values and then 250 pentamers with the highest values are selected.

More recently, the promoter prediction system has adopted the concept of relative entropy in the feature selection process [Wu, Xie et al. 2007]. Kullback-Leibler divergence is used to calculate the distance, which is defined as follows:

$$\delta(p^k_{promtoer}, p^k_{non-promoter}) = \sum_{i=1}^{4^k} p^k_{promoter}(i) \ln \frac{p^k_{promoter}(i)}{p^k_{non-promoter}(i)} \tag{2.15}$$

where $p^k_{promoter}$ and $p^k_{non-promoter}$ represent the probability density functions of words (the combination of A, C, T and G) in promoter and non-promoter sequences. k ($k = 4, 5, 6, 7$) indicates the fixed word length and the total number of words is 4^k. One subgroup of the most effective words can be obtained by maximizing the following criterion function:

$$\begin{aligned} S &= \arg_{\{i|i\in\{1,2,\ldots 4^k\}\}} \left\{\max \delta(p^k_{promoter}, p^k_{non-promoter})\right\} \\ &= \left\{i \mid p^k_{promoter}(i) > p^k_{non-promoter}(i)\right\} \end{aligned} \tag{2.16}$$

where S represents the set of subscripts of all the words in the subgroup that are selected. The desirable number of words within a subgroup can be selected by sorting $\{p^k_{promtoer}(i) \ln \frac{p^k_{promtoer}(i)}{p^k_{non-promtoer}(i)}, i \in S\}$ in descending order.

In conclusion, the concepts of statistics and probability are widely used in sequence feature selection processing, and the prediction performances of the promoter recognition models are greatly improved with the introduction of these methodologies.

2.3 Review of Modeling and Classification Methodology

To construct a model with effective classifiers is the final step of promoter recognition systems. Pattern recognition approaches are widely used in the promoter prediction area. Pattern recognition models have the ability to determine if an unknown sequence belongs to some group it has ″seen″ before, because the characteristic of a certain sequence group can be remembered by the model [Attwood and Parry-Smith 1999]. Several methodologies of building classifiers and models that are used in promoter recognition program are discussed here.

2.3.1 Introduction of modeling and classification methodology

The process of building models and classifiers is always implemented with machine learning technologies. Machine learning is an adaptive process that enables computers to learn from experience [Chen(Ed.) 2005]. Several machine learning approaches for building models and classifiers are introduced. As some of these methods take PWM scores as input signals, the classification function of PWM is presented first.

PWM

Apart from sequence feature extraction, PWM is also used for classification in promoter recognition. The scores that are needed in a decision process can be

obtained by calculating the log likelihood that it belongs to the same class using the weight matrix. As we mentioned in section 2.2.1; usually, two weight matrices of the same feature are generated from both positive and negative training sequence sets. Given an unknown sequence, two scores will be obtained according to these two matrices. A decision then can be made by simply comparing the two scores, or it may be based on more complex rules carried out by a neural network or learning machine.

Artificial neural network (ANN)

ANN is one of the most popular modeling and classification tools because of its powerful intelligent learning ability. ANN can take various input signals; for example, the scores generated by the PWM, or directly embed sequence features into the framework. As the promoter recognition system I built is developed using ANN, the theory and application will be discussed in Chapter 3.

Support vector machine (SVM) and Relevance vector machine (RVM)

SVM is always used as an effective classifier for separable and non-separable data. This is so because the good performance can be generated without prior knowledge of the problem and better performance can be achieved by incorporating prior knowledge into SVM [Burges 1998]. The problem of the slow speed in the test phase can be resolved by combining other algorithms with SVM [Scholkopf, Smola et al. 1998]. RVM [Tipping 2001] is also a sparse training algorithm like SVM. It can select the most helpful functions for classification and discard useless basis functions from

provided functions [Down and Hubbard 2002]. At the same time parameters of the learning machine can be trained by these probabilistic and distance functions. Both SVM and RVM are widely used in promoter recognition systems.

2.3.2 Summary of modeling and classification methodology used in existing promoter prediction models

Pattern recognition approaches are widely used in the promoter prediction area and it is seldom found that different promoter prediction systems construct exactly the same models or classifiers. Each model has its own advantages and disadvantages. A summary of classification and modeling methodology used in promoter prediction systems is listed in Table 2.3. The selected systems are the same as the ones in Table 2.2.

Table 2.3 Summary of classification and modeling methodology used in promoter prediction systems

System	Classification and Modeling Methodology
PWMs	Probabilistic model based on PWM scoring block
PromoterScan	Probabilistic model based on PWM scoring block
PromFD	Scoring model based on Information matrix database
PromoterInspector	Prediction model based on IUPCA matched rules
NNPP	Time-delay neural networks for feature extraction, detection and functional regions prediction

FirstEF	Probabilistic models based on QDF
DFP	Classifiers built by ANN with input scores from feature weight matrices
DragonGSF	Classifiers built by ANN with input scores from feature weight matrices
CpGProD	Linear model based on CpG island score
Eponine	RVM built model initialized by PWM and random Gaussian position distribution
McPromoter	ANN incorporated with hidden Markov chain
PromPredictor	Classifier built by ANN
PromoterExplorer	Classifier trained by AdaBoost
KLD	decision model based on PWM scoring block

Chapter 3 Application of Principal Component Analysis and Artificial Neural Network to Promoter Feature Selection

This section presents the promoter feature selection process, which is the most important part of building the classification model HPR-PCA. In this human promoter recognition model, principal component analysis (PCA) is adopted to reduce the dimensionality and select the most discriminative context features from promoter and non-promoter sequences. The theory and application of PCA are stated, and as the implementation tool, back propagation (BP), artificial neural network (ANN), and a holdout validation training method are also introduced. Finally, the application of PCA for feature selection of the model HPR-PCA is presented in detail, and the conclusions are made through a series of comparative experiments.

3.1 Principal Component Analysis

Principal Component Analysis (PCA) is one of the best known techniques of

multivariate analysis [Jolliffe 1986] and it has wide applications in areas from neuroscience to computer graphics. The central idea of PCA is to transform the original variables to a new set of variables which are projected into a new space. By reducing the dimensionality of a high dimensional dataset, the first few principal components are retained according to their order, which can represent the most variation in the original data set. The theory and the application of PCA are stated here.

3.1.1 Theoretical background

PCA is the application of linear algebra, and there are some important concepts in linear algebra which are useful for understanding PCA.

Variance is a measure of the spread of data in a data set. Variance of data set a is defined as:

$$s_a^2 = \frac{\sum_{i=1}^{n} (a_i - \bar{a})^2}{n-1} \tag{3.1}$$

where a_i is the element of a. n is the total number of elements in data set a. $\bar{a}$ is the mean of a, which is calculated by:

$$\bar{a} = \frac{\sum_{i=1}^{n} a_i}{n} \tag{3.2}$$

Variance is purely operated on one dimensional data. When calculating how much the dimensions vary from the mean with respect to each other, we need to refer to covariance. Covariance is used to measure two or more dimensional data. Given another data set b, which has the same number of samples as data set a, the

covariance between b and a can be calculated as follows:

$$\mathrm{cov}(a,b) = \frac{\sum_{i=1}^{n}(a_i - \bar{a})(b_i - \bar{b})}{n-1} \tag{3.3}$$

If we calculate the covariance between a and itself, the result is variance[Smith 2002]. We can build a matrix :

$$C = \begin{bmatrix} c_1 \\ \vdots \\ c_m \end{bmatrix}, \tag{3.4}$$

and make row vectors:

$$c_1 \equiv a = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix}, \tag{3.5}$$

$$c_2 \equiv b = \begin{bmatrix} b_1 & b_2 & \cdots & b_n \end{bmatrix}, \tag{3.6}$$

$c_3 ... c_m$ are set by additional row vectors.

Let us define a covariance matrix:

$$C_c \equiv \frac{1}{n-1} CC^T \tag{3.7}$$

where C_c is an $m \times m$ covariance matrix. The diagonal elements of C_c are the variance data type, i.e., s_a^2, s_b^2, and the off-diagonal elements are the covariance data type, i.e., $\mathrm{cov}(a,b)$. Here large values of variance elements represent the signal that are of interest and small ones represent noise. Covariance is a measure of how much two random variables correlate to each other. A large diagonal element in the covariance matrix corresponds to high independency between two variables and small one corresponds to low independency. In PCA, we want to make new variables be independent to each other through linear transformation. Therefore, the ideal solution is to diagonalize the covariance matrix C_c, or to find another matrix related to C_c,

making the off-diagonal elements of the matrix become zero through linear transformation. In order to achieve this goal, let us refer to an orthonormal matrix P,

where $$P^{-1} = P^T \tag{3.8}$$

and a new matrix Y,

where $$Y = PC \tag{3.9}$$

Y is the projection of C based on new space P. We can make the following deduction:

$$\begin{aligned} C_Y &= \frac{1}{n-1} YY^T \\ &= \frac{1}{n-1} (PC)(PC)^T \\ &= \frac{1}{n-1} PCC^T P^T \\ &= \frac{1}{n-1} PAP^T \end{aligned} \tag{3.10}$$

where $$A \equiv CC^T \tag{3.11}$$

As A is symmetric, we can find matrix E and D so that $A = EDE^T$, where D is a diagonal matrix and E is a matrix of eigenvectors of A arranged as columns. Thus, we can select P where each row P is an eigenvector of CC^T. Now, we can rewrite C_Y in terms of P and D.

$$\begin{aligned} C_Y &= \frac{1}{n-1} PAP^T \\ &= \frac{1}{n-1} P(P^T DP)P^T \\ &= \frac{1}{n-1} (PP^T)D(PP^T) \\ &= \frac{1}{n-1} (PP^{-1})D(PP^{-1}) \\ &= \frac{1}{n-1} D \end{aligned} \tag{3.12}$$

It is obvious that P is the matrix that can diagonalize C_Y. The eigenvalues of C_Y (diagonal values in D) are the variances of C, and the row vectors of P corresponding to the largest eigenvalues are the principal components (PCs) of C [Shlens 2005]. By linear transformation, the covariance matrix C_c is transformed to C_Y, so the problem becomes to digonalize C_Y. Following the above deduction, the problem is simplified to finding the eigenvectors and eigenvalues of the covariance matrix C_c.

After obtaining matrix D, we rank the values of the diagonalized elements. The larger values in matrix D associate with higher levels of energy. We need to use these values to estimate how many PCs we need to built a classifier. Let us define the matrix D:

$$D = \begin{pmatrix} d_1 & & 0 \\ & \ddots & \\ 0 & & d_N \end{pmatrix} \tag{3.13}$$

The idea is to select the smallest number of PCs that should contribute a certain percentage of total variation. The selection can be completed by the following function:

$$t_{n_0} = \sum_{j=1}^{n_0} d_j \Big/ \sum_{j=1}^{N} d_j \tag{3.14}$$

In this function, d_j is the jth diagonal value of D. In order to retain as much information as possible and reduce the dimensionality at the same time, we chose a cut-off value $t_M = 0.7$, where n_0 is the smallest integer, for which $t_{n_0} > t_M$ [Jolliffe 1986].

Therefore, n_0 eigenvectors from matrix P which correspond to the first n_0 eigenvalues are selected. The new matrix constructed by these orthonormal eigenvectors can be regarded as the basis of a new space, into which the original data set is projected.

At last, the solution of PCA can be concluded in five steps:

1. For a given data set, arrange it into a $m \times n$ matrix, where m is the number of measurement types and n is the number of samples. In our experiment, m specifically represents a number of sequence features and n is the sequence number in a training data set.

2. Calculate the mean for each measurement and generate a covariance matrix.

3. Calculate the eigenvectors and eigenvalues of the covariance matrix.

4. Rank the eigenvalues of the covariance matrix and determine how many PCs are needed for the classifier.

5. Select the eigenvectors according to step 4, and form a new space.

3.1.2 Bioinformatics applications of PCA

A genomic data set is probably the largest data set with high complexity and redundancy, so clustering algorithms or dimensional reduction algorithms [Raychaudhuri, Stuart et al. 2000] are important for genetic analysis. With wide application, PCA is used for pattern recognition in microarray data sets

[Raychaudhuri, Stuart et al. 2000], species-specific codon usage analysis [Medigue, Rouxel et al. 1991] [Kanaya, Yamada et al. 1999], clustering gene expression data [Yeung and Ruzzo 2001] and so on. In these applications, PCA performs well on extracting the useful signals while reducing the noise in genetic datasets. It is shown that PCA can automatically detect the redundancies in a data set and define a smaller hybrid feature [Raychaudhuri 2006]. Those important features that account for most variables in an original data set are mostly selected as principal components for clustering or classification. As PCA works well on dimension reduction and feature extraction, it is selected as the feature extraction algorithm in HPR-PCA.

3.2 Artificial Neural Network and Training Method

Machine learning approaches gradually replace traditional computer science techniques and algorithms in bioinformatics [Chen(Ed.) 2005]. Artificial neural network (ANN) is one of the machine learning mechanisms which is widely used in genetic sequence analysis. The basic concept of ANN is introduced firstly and followed by a backpropagation (BP) learning algorithm which is used in HPR-PCA. HPR-PCA adopts the holdout validation training method in training and test processes, so it is also stated in this section.

3.2.1 Artificial Neural Network and Backpropagation Learning Method

ANN is a kind of adaptive machine learning algorithm that can learn from experience, and it is also intelligent as it has the ability to recognize new things. ANN usually has several layers and each layer contains several neurons. In a typical ANN, the first layer is the input layer, where the information enters the network. The middle layers are hidden layers, which contain neurons that combine the input features and compute more complex functions. The last layer is the output layer that is connected with the nodes in the hidden layer and combines them to decide on a final decision [Chen(Ed.) 2005]. In the classification process, ANN firstly imitates the human brain to capture the input-output relationship and update the inner network structure, and then classifies those new input data according to the stored knowledge. A simple illustration of ANN is shown Figure 3.1.

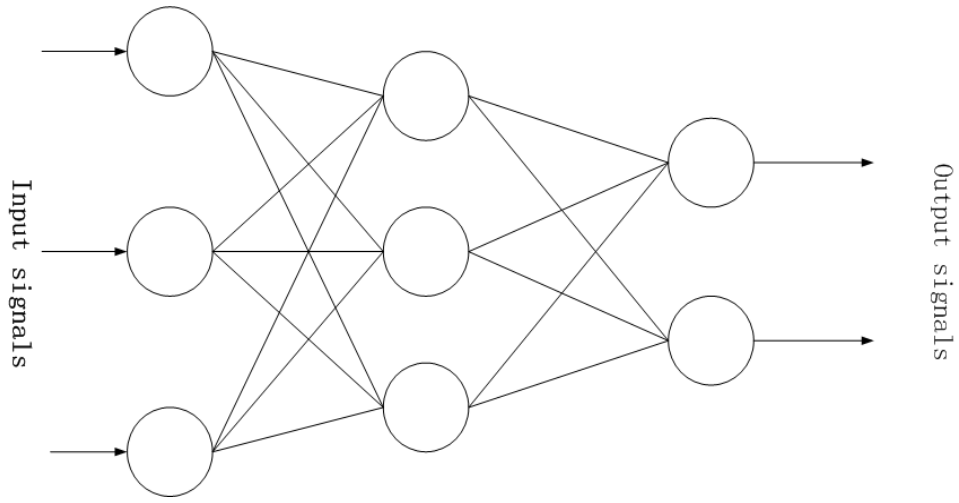

Figure 3.1 Schematic illustration of an ANN

A neuron is the basic element of ANN. In a neuron, the sum of weighted inputs and

bias are sent to the transfer function, and each neuron can apply different transfer functions to generate the output. The inputs of a neuron can be either signals from an outside network or outputs of other neurons, and the outputs can be either the final results of the network or inputs to other neurons. Figure 3.2 shows the model of a general neuron.

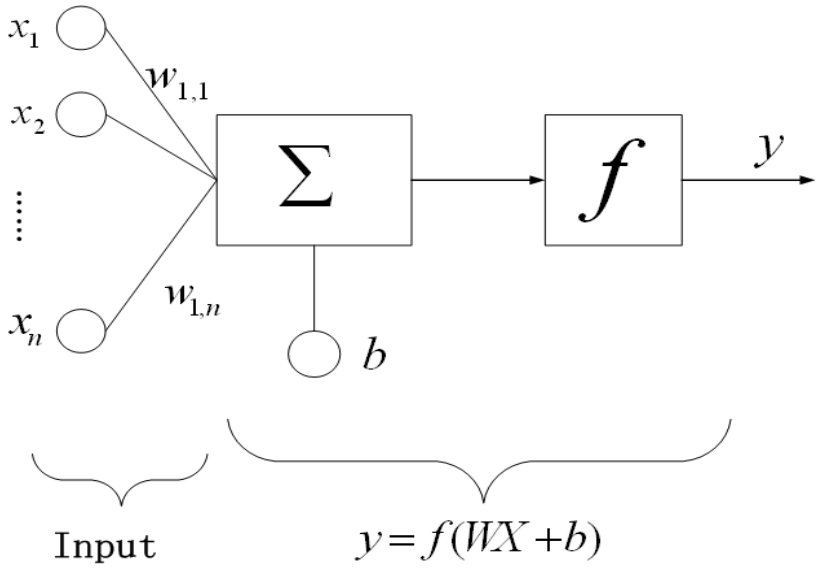

Figure 3.2 The structure of a general neuron.

X is the input vector and y is the output. W represents the weight vector and b represents the bias. f is the transfer function.

The ANN built classifier is used for classifying promoter regions and non-promoter regions, therefore, the output results are non-linear. Two non-linear transfer functions: log-sigmoid and tan-sigmoid are used in HPR-PCA. Figure 3.3 and Figure 3.4 are the graphs of log-sigmoid and tan-sigmoid transfer functions.

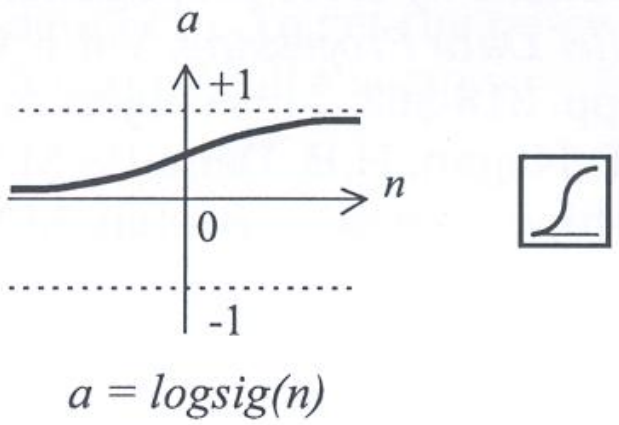

Figure 3.3 The figure of log-sigmoid transfer function

The log-sigmoid algorithm is:

$$\log\mathrm{sig}(n) = \frac{1}{(1+\exp(-n))} \tag{3.15}$$

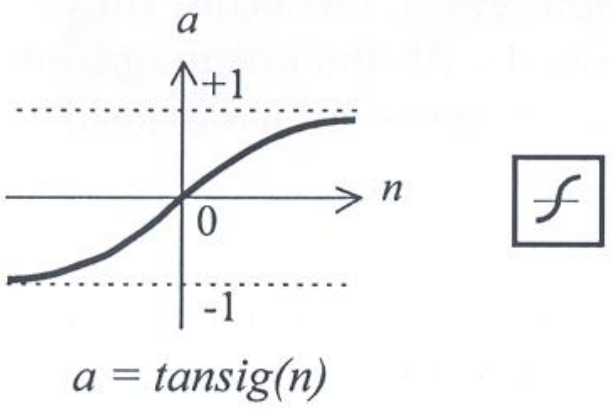

Figure 3.4 The figure of tan-sigmoid transfer function

The tan-sigmoid algorithm is:

$$\tan\mathrm{sig}(n) = \frac{2}{1+\left(\exp(-2\times n)\right)} - 1 \tag{3.16}$$

A feed-forward neural network is used in HPR-PCA. In a feed-forward neural network, each neuron in one layer connects with every neuron in the next layer but not to those in the same layer or a previous layer. BP is the most common algorithm of a feed-forward network. It minimizes the error function which is calculated by inputs and target outputs in a training set and updates the weights by using the method of gradient descent [Rojas 1996]. A simple three-layer BP learning process is introduced as follows:

1. Initialize the w_{ij} $i=1,2,\cdots,n$ $j=1,2,\cdots p$, v_{jt} $(j=1,2,\cdots,p$ $t=1,2,\cdots,q)$, which are the connection weights between input and hidden layers and the connection weights between hidden and output layers. The initiation values can be randomly selected from the range $[-1,\ 1]$. n represents the dimension of the input vector, p represents the number of neurons in the hidden layer and q is the neuron number of the output layer.

2. Calculate the input and output of the hidden layer:

$$s_{j=}\sum_{i=1}^{n} w_{ij}x_{i-}\ b \tag{3.17}$$

$$y_j = f(s_j) \tag{3.18}$$

where $P_k=[x_1,x_2,\cdots,x_n]$ is the input vector. b_j is the output bias of the hidden layer. $S_k=(s_1,s_2,\cdots,s_p,)$ is the input vector of the hidden layer. y_j is the output of each neuron in the hidden layer. f is the transfer function, and we only use log-sigmoid and tan-sigmoid transfer functions in the ANN training process of

3. Calculate the input and output of the output layer:

$$L_t = \sum_{j=1}^{p} v_{jt} y_j - \gamma_t \quad t = 1, 2, \cdots, q \tag{3.19}$$

$$C_t = f(L_t) \qquad t = 1, 2, \cdots, q \tag{3.20}$$

where γ_t is output bias of the output layer and L_t is the input vector of the output layer. C_t is the response of the output layer .

4. Calculate the gradient of the error function d_t^k of the output layer with target output $T_k = \left[t_1, t_{2,} \cdots, t_q \right]$ and C_t:

$$d_t^k = \left(t_t^k - C_t \right) \square C_t \square \left(1 - C_t \right) \qquad t = 1, 2, \cdots, q \tag{3.21}$$

5. Calculate the gradient of the error function e_j^k by v_{jt} and d_t^k:

$$e_j^k = \left[\sum_{t=1}^{q} d_t^k \square v_{jt} \right] \square y_j \square (1 - y_j) \tag{3.22}$$

6. Adjust the connection weights v_{jt} and the output bias γ_t by d_t^k and y_j:

$$v_{jt}(N+1) = v_{jt}(N) + \alpha \square d_t^k \square y_j \tag{3.23}$$

$$\gamma_t(N+1) = \gamma_t(N) + \alpha \square d_t^k \tag{3.24}$$

$t = 1, 2, \cdots q \quad j = 1, 2, \cdots p \quad , \quad 0 < \alpha <$

where α is the learning constant.

7. Adjust the connection weights w_{ij} and the output bias b_j by e_j^k and input x_i:

$$w_{ij}(N+1) = w_{ij}(N) + \beta e_j^k x_i \quad (3.25)$$

$$b_j(N+1) = b_j(N) + \beta e_j^k \quad (3.26)$$

$i = 1, 2, \cdots, n \quad j = 1, 2, \cdots, p, \quad , \beta <$

where β is the learning constant.

8. Randomly select the next training sample, sent it to the ANN and repeat steps 1 to 7

9. When all the samples in the training set are trained once, evaluate the error of the network. If the error reaches an acceptable value, it means that the ANN is trained successfully, otherwise the training is unsuccessful.

The BP algorithm uses the gradient descent method to minimize the error functions, so the continuity and differentiability of transfer functions need to be guaranteed [Rojas 1996]. The BP network performs well on classifying massive and complex data, therefore, it is ideal for classifying DNA sequence data.

3.2.2 Holdout Validation Training Method

Holdout validation training method is used in the HPR-PCA feature selection process. In all sequence datasets (promoter, exon, intron and 3' UTR), training data are randomly selected from the original data set and the remaining observations are

retained as validation data.

3.3 Features Selection Based on PCA

Finding effective features is essential for building powerful classifiers. As mentioned in Chapter 2, there are two types of features used in promoter recognition approaches: sequence signal features and context features. In HPR-PCA, PCA is used to select discriminative context features. Context features take an important position in our promoter prediction model, because nearly all of the signal features are included in context features, and they have more common patterns for genome-wide promoter regions prediction. The following section states the feature selection process in HPR-PCA step by step. The evaluation step is implemented with ANN.

3.3.1 Feature Matrix Generation

Feature matrices are extracted from human promoter and non-promoter sequences, which are accessible in public databases. Human promoter sequences used in this experiment are from Eukaryotic Promoter Databases (EPD), Release 86 [Schmid, Perier et al. 2006], and from the database of transcription start sites (DBTSS), version 5.2.0 [Suzuki, Yamashita et al. 2002]. Human non-promoter sequences used in this model are exon sequence, intron sequences and 3' UTR sequences. Human exon and intron sequences are extracted from the exon-intron database [Saxonov, Daizadeh et al. 2000], and the human 3' UTR sequences are from the UTR database [Pesole, Liuni

et al. 2001]. Among these sequences, only the promoter sequences from EPD have a fixed length of 1200bp, the length of sequences from other databases varies.

For training and testing purposes, promoter sequences from 250bp upstream to 50bp downstream of the transcription binding site are extracted from EPD and DBTSS promoter databases respectively. Non-promoter sequences including exon sequences, intron sequences and 3'UTR sequences whose length are over 1200bp (compared to the sequence length in EPD) are selected from those three non-promoter databases mentioned above and arranged into 300bp each. Those sequences with letter ″N″ are not included in the selected data sets.

A DNA sequence consists of four types of nucleotides, so with different combinations, there are $4^3 = 64$ codens (3-mers), $4^4 = 256$ 4-mers, and $4^5 = 1024$ pentamers (5-mers) in promoter and non-promoter sequences. The feature matrices are based on the n-mer frequencies and the steps of feature matrix generation are as follows:

1. Count the overlapping 3-mer, 4-mer and 5-mer frequencies of promoter and non-promoter sequences. For a sequence with length L bp, the counting window moving 1bp per step, there will be $(L-n+1)$ n-mers counted in each overlapping sequence matrix ($n = 3, 4, 5$). As the length of each sequence is fixed at 300bp, there are 298 3-mers, 297 4-mers and 296 5-mers involved in counting 3-mer, 4-mer and 5-mer frequency matrices respectively.

2. Generate three types of frequency feature matrices for each sequence group: 3-mer matrix, 4-mer matrix and 5-mer matrix with the dimension $64\times n_s$, $256\times n_s$ and $1024\times n_s$ respectively, where n_s is the number of sample sequences. A total of 12 feature matrices are extracted from the promoter and non-promoter data sets. Let us refer to $a_0(i_1, j)$, $b_0(i_2, j)$ and $c_0(i_3, j)$ as the 3-mer, 4-mer and 5-mer frequency matrix respectively, where $i_1 = 1, 2, \cdots, 64$, $i_2 = 1, 2, \cdots, 256$, $i_3 = 1, 2, \cdots, 1024$ and j represents the number of sequences.

3. Normalize the matrices as follows:

$$a(i_1, j) = \frac{a_0(i_1, j)}{a_{\max}} \tag{3.27}$$

$$b(i_2, j) = \frac{b_0(i_2, j)}{b_{\max}} \tag{3.28}$$

$$c(i_3, j) = \frac{c_0(i_3, j)}{c_{\max}} \tag{3.29}$$

where $a(i_1, j)$, $b(i_2, j)$ and $c(i_3, j)$ are the 3-mer, 4-mer and 5-mer feature matrix respectively. $a_{\max}$, $b_{\max}$ and $c_{\max}$ are maximum values of the 3-mer, 4-mer and 5-mer matrix respectively.

Combine these three normalized matrices in four ways:

$$C_1(i_{ab, j}) = \begin{bmatrix} a(i_1, j) \\ b(i_2, j) \end{bmatrix} \qquad i_{ab} = 1, 2, \cdots, 320 \tag{3.30}$$

$$C_2(i_{ac, j}) = \begin{bmatrix} a(i_1, j) \\ c(i_3, j) \end{bmatrix} \qquad i_{cd} = 1, 2, \cdots, 108 \tag{3.31}$$

$$C_3(i_{bc, j}) = \begin{bmatrix} b(i_2, j) \\ c(i_3, j) \end{bmatrix} \qquad i_{bc} = 1, 2, \cdots, 128 \tag{3.32}$$

$$C_4(i_{abc}, j) = \begin{bmatrix} a(i_1, j) \\ b(i_2, j) \\ c(i_3, j) \end{bmatrix} \qquad i_{abc} = 1, 2; \cdots, 13 \tag{3.33}$$

where $C_1(i_{ab,j})$ is the combination of the 3-mer and 4-mer matrix, and $C_2(i_{ac,j})$ is the combination of the 3-mer and 5-mer matrix. $C_3(i_{bc,j})$ is the combination of the 4-mer and 5-mer matrix, and $C_4(i_{abc}, j)$ is the combination of the 3-mer, 4-mer and 5-mer matrix.

Now, the 3-mer, 4-mer and 5-mer matrices $a(i_1, j)$, $b(i_2, j)$ and $c(i_3, j)$ together with their combined matrices $C_1(i_{ab,j})$, $C_2(i_{ac,j})$, $C_3(i_{bc,j})$ and $C_4(i_{abc}, j)$ can be extracted from each sequence group. In order to select features to build three classifiers— Promoter versus Exon , Promoter versus Intron and Promoter versus 3'UTR classifier, one promoter matrix and one non-promoter matrix are combined in pairs and the n-mer frequency matrices of non-promoter sequence are normalized by dividing the maximum value of corresponding promoter matrices..

3.3.2 Feature Selection Based on PCA

The next step is to find the most effective n-mer feature combination for classification from different matrices. Promoter and exon feature matrices are used in a comparative experiment. Seven pairs of matrices are separately used as input matrices of the previously designed PCA algorithm: firstly, generate the covariance matrices of

each matrix; secondly, calculate the eigenvectors and eigenvalues of these covariance matrices; thirdly, rank the eigenvalues in descending order. The first three eigenvectors of each covariance matrix are selected to form seven new hybrid feature vectors, on which the original promoter feature matrices and exon feature matrices are projected. Next, the new promoter and exon vectors are sent to the neural network for training. The target outputs are set to ″1″ corresponding to promoter vectors and ″0″ corresponding to exon vectors. Figure 3.5 shows the flow chart of the training process.

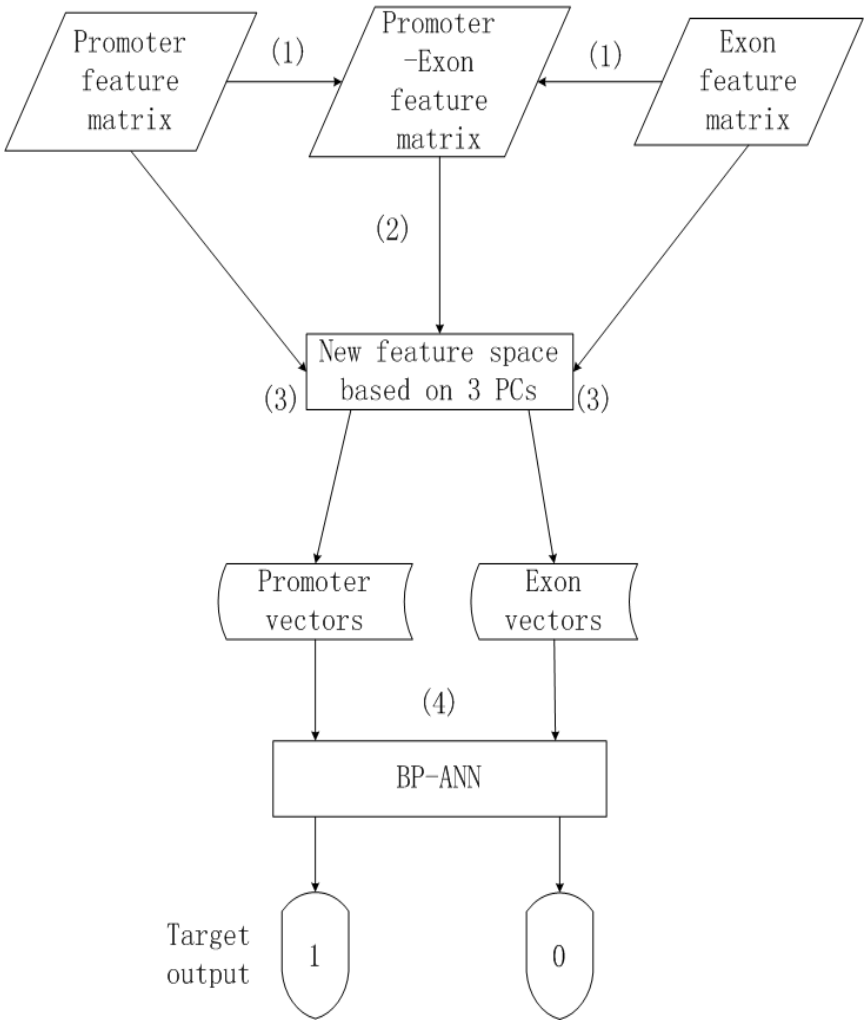

Figure 3.5 The flow chart of training process

(1) Combination (2) PCA (3) Projection (4) Neural Network Training

In total, there are seven networks based on seven different n-mer combined feature matrices. Five thousand promoter sequences from DBTSS and 5000 exon sequences from the exon-intron database constitute the training set of the comparative experiment. The 3-mer, 4-mer and 5-mer frequency matrices are extracted from the promoter and exon training sequences separately, and then seven promoter-exon combined feature matrices (3-mers, 4-mers, 5-mers, 3 and 4-mers, 3 and 5-mers, 4 and 5-mers and 3, 4 and 5-mers) are generated to train those seven networks. For test purposes, five thousand promoter and 5000 exon sequence segments are selected from the same database as the training sequences and the length of these sequences are all 300bp. Sequences of training sets and test sets are not repeated or overlapped. A simple three layers BP network is built to train and test these feature vectors. Three transfer functions of the three layers are “tan-sigmoid”, “log-sigmoid” and “tan-sigmoid” respectively and the holdout validation training method is used in the training and test processes. The training epochs are set to 10000 and the classification threshold is set to 0.5. The network parameters are not optimized here as the purpose of this comparative experiment is only to testify which n-mer combination shows the best discrimination. The test process is shown in Figure 3.6.

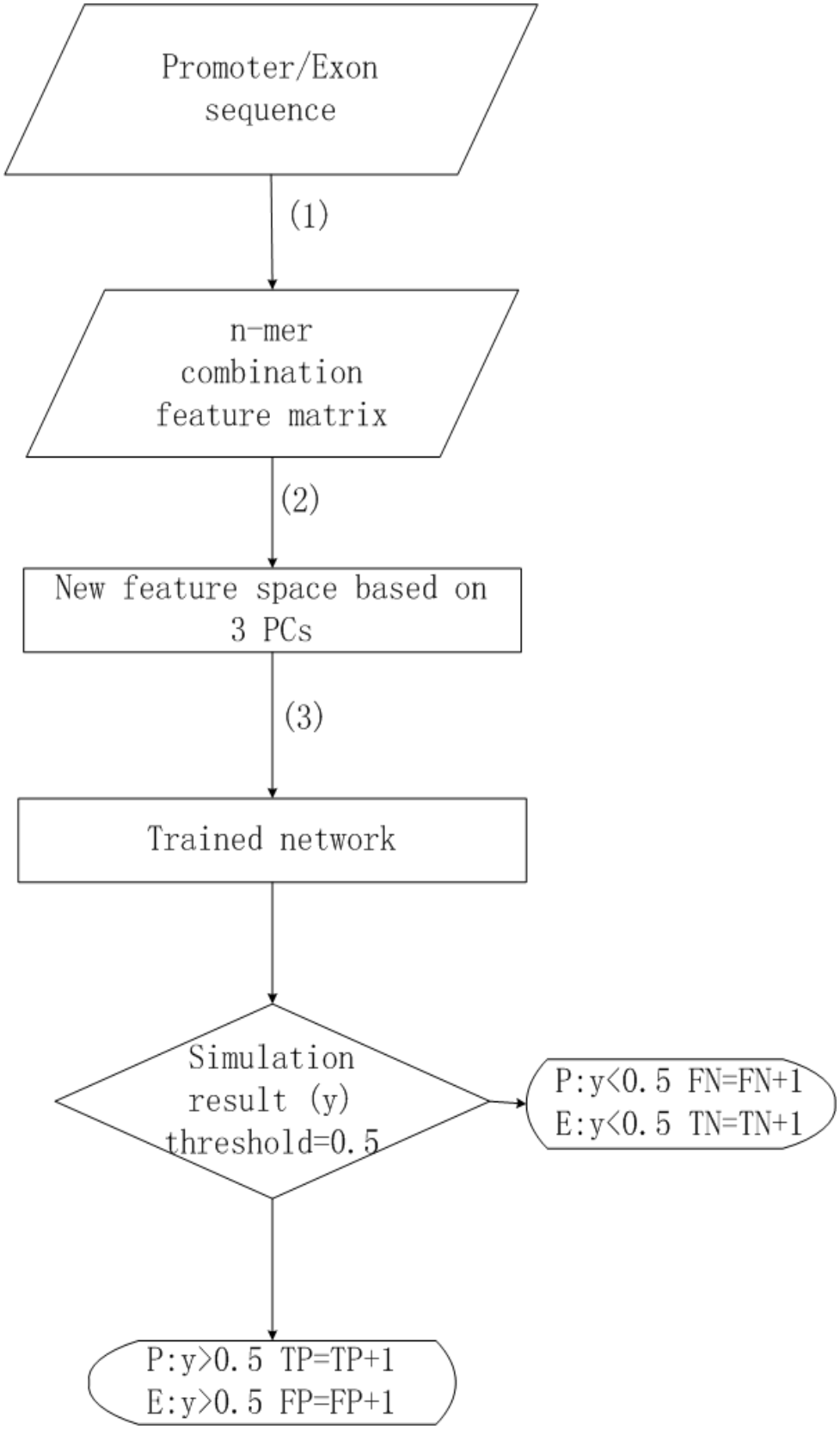

Figure 3.6 The flow chart of the testing process

(1)Feature matrix generation (2) Projection (3) Classification with ANN

In the comparison, two evaluation criteria—sensitivity (S_e) and specificity (S_p) are evaluated. True positive (TP), false negative (FN) and false positive (FP) are also counted for statistical purposes.

$$Se = \frac{TP}{TP + FN} \tag{3.34}$$

$$S_p = \frac{TP}{TP + FP} \tag{3.35}$$

In order to reduce the error produced in the ANN training process, all the networks are trained and tested three times with the same data groups and the results taken are the average values. Table 3.1 shows the comparison results of the seven trained networks.

Table 3.1 Comparison results of seven networks

Network	*Se*	Sp	Se + Sp	Ranking
3-mer	0.6856	0.5945	1.2801	4
4-mer	0.6512	0.5715	1.2227	5
5-mer	0.7100	0.6150	1.3250	3
3&4-mer	0.6260	0.5823	1.2083	7
4&5-mer	0.6830	0.5342	1.2172	6
3&5-mer	0.7340	0.6500	1.3840	1
3&4&5-mer	0.6980	0.6282	1.3262	2

3.3.3 Conclusion

In the comparative experiment, the ANN trained with the 3 and 5-mer feature matrices

achieves the best results among the seven networks, so the 3 and 5-mer combined feature matrices will be adopted in the human promoter recognition network.

Considering other feature selection, TFBSs are relative short and if without reasonable combinations with other features, this kind of features will lead to a high false positive rate. According to the analysis of 1871 human promoter sequences, in EPD [Schmid, Perier et al. 2006], TATA-only and Inr-only promoters account for just 6% and 9% respectively. As we mentioned in Chapter 2, more than half of human promoters are CpG-islands related, therefore, CpG-island features are combined with the selected context feature in our human promoter recognition network. The implementation of the whole network will be introduced in Chapter 4.

Chapter 4 Human Promoter Recognition Network

In this chapter, two schemes are proposed first for a human promoter recognition network, and the implementations of the sub-modules are also specifically presented. Network parameters are optimized by a comparative experiment. Marked promoter sequences, non-promoter sequences and three human DNA sequences are selected to test the performance of the two proposed schemes. Finally, Scheme I is adopted as the HPR-PCA model as it achieves better overall results.

4.1 Overview of Human Promoter Prediction Network

Here the overall structures of two proposed human promoter prediction networks are presented. The difference between Scheme I and Scheme II is the utilization of CpG-islands signal. Scheme II divides sequences into CpG-islands related and CpG-islands non-related groups before sending these sequences into to classifiers, while Scheme I combines the CpG-island signal directly with the classification results from the classifiers. The overall structures of Scheme I and Scheme II are show in

Figure 4.1 and Figure 4.2.

In order to predict TSS along large genome sequences, a sliding window is set up first. The window size is 300bp and it moves 20bp in each step in our model. In Scheme I, each sequence segment from the sliding window will receive a score from the CpG-islands module and at the same time the feature generation module extracts a 3 and 5-mer feature vector of each sequence segment. The PCA module projects the feature vector into the new space which is constructed during the training processes. Next the new vector is sent to three classifiers: the Promoter vs. Exon classifier, Promoter vs. Intron classifier and Promoter vs. 3' Utr classifier; which perform separate classifications. The three scores from the classifiers together with the one form the CpG-islands module are processed in the data processing model, where the final prediction of TSS is produced.

In Scheme II, each 300bp sequence segment is sent to CpG islands module first and is classified to a CpG islands related sequence or non-CpG islands related sequence. As mentioned in Chapter 2, when the GC percentage is over 0.5 and the observed/expected CpG ratio is over 0.6, the sequence is CpG related, otherwise, it

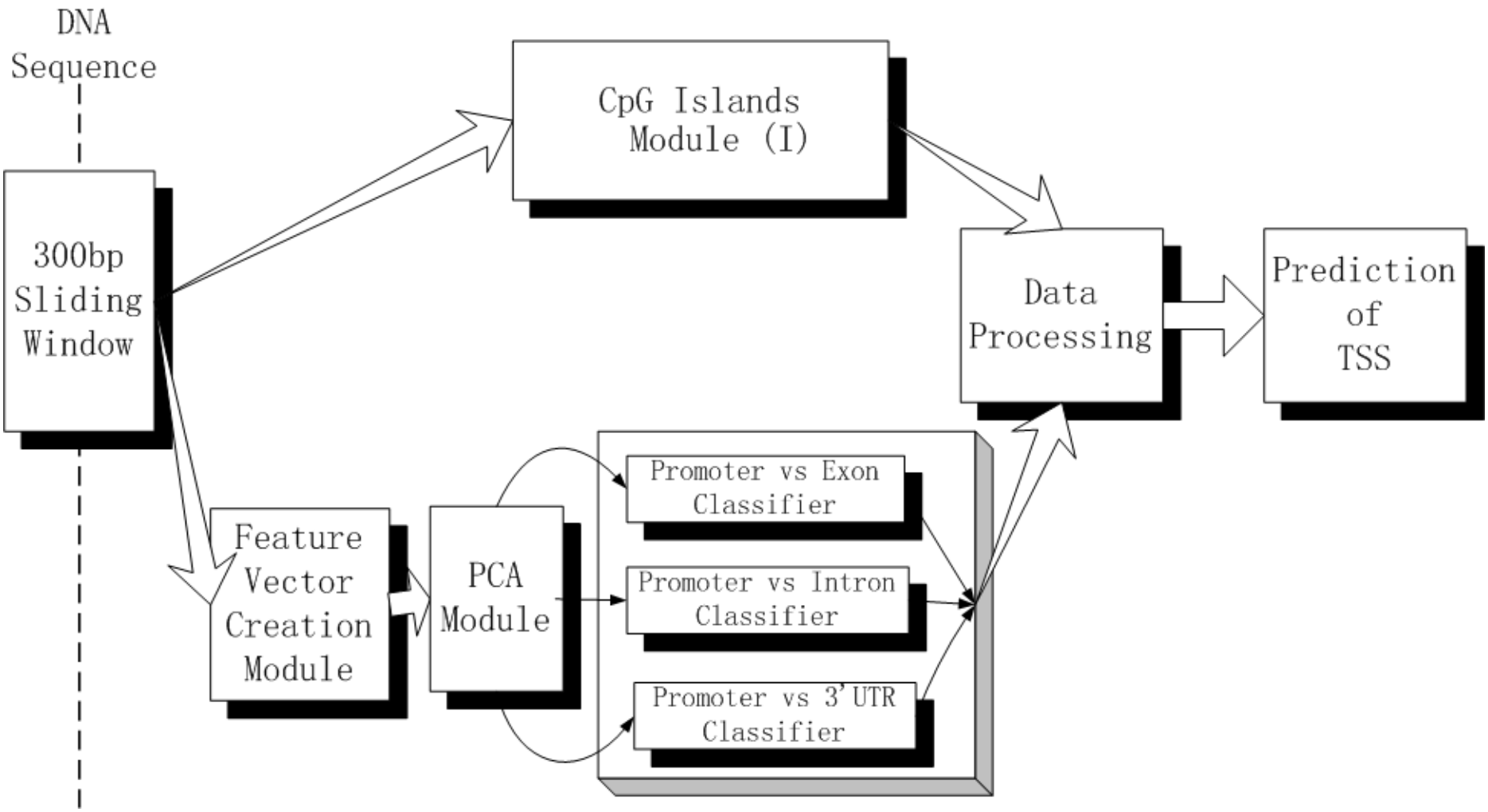

Figure 4.1 The overview structure of Scheme I

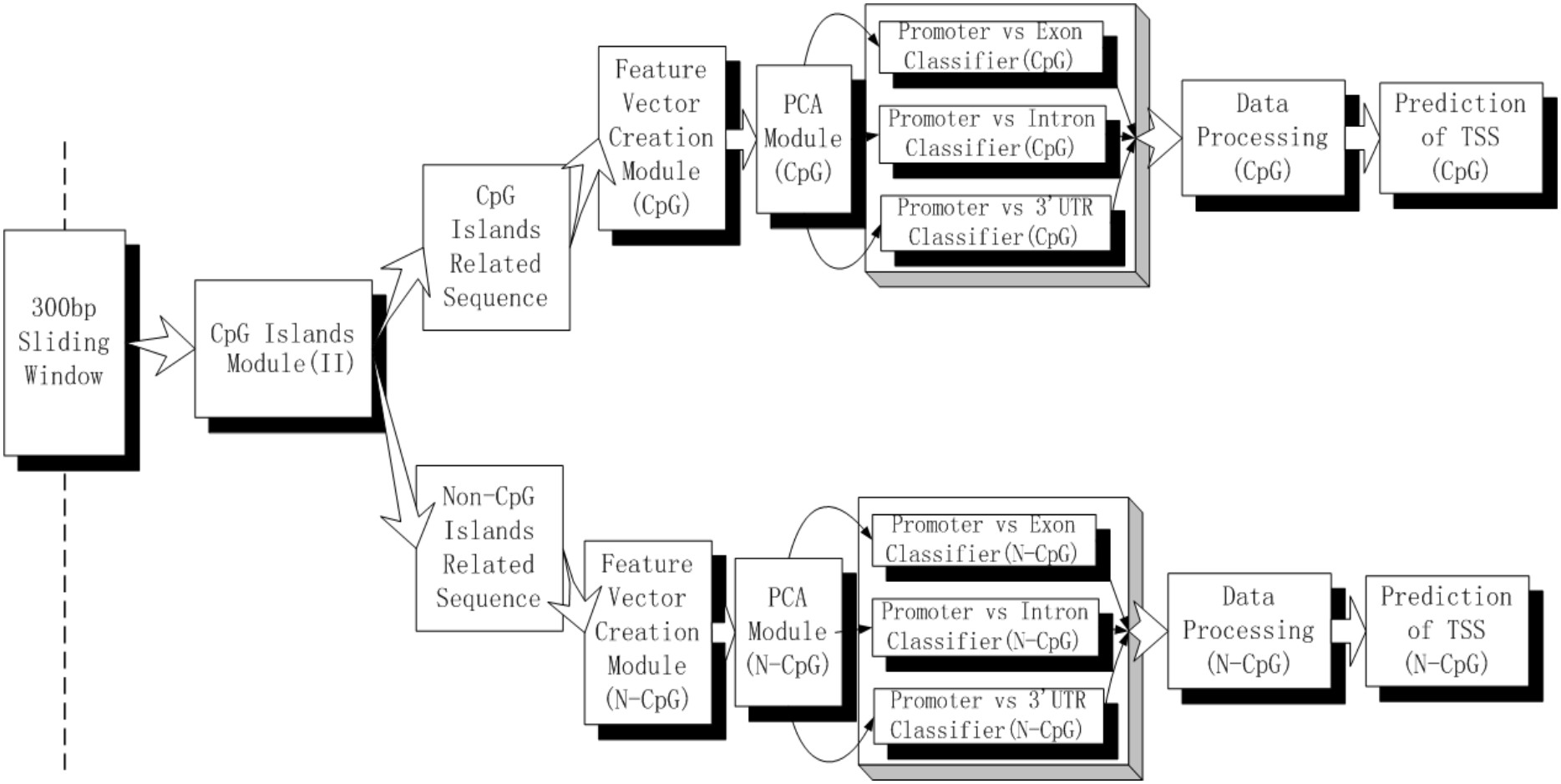

Figure 4.2 The overview structure of Scheme II

is non-CpG islands related. The further processes of feature extraction, principal components selection, sequence classification and TSS prediction are the same as Scheme I, but the classified CpG islands related sequences and non-CpG islands related sequences are processed separately. Therefore, the CpG islands related and non-CpG islands related classifiers are trained with corresponding CpG islands related and non-CpG islands related sequences. The training process will be introduced specifically in the classifiers construction section.

4.2 Implementations of Sub-Modules

In the human promoter recognition network, all of the Sub-Modules are divided into four main groups: feature vector creation and PCA modules, classifiers for promoter and non-promoter sequences, CpG islands module, and data processing and TSS prediction modules. In this section, the implementations of these groups and their differences in Scheme I and Scheme II are discussed.

4.2.1 Feature Vector Creation and PCA Modules

Feature vector creation and PCA modules are two of the most important modules in the whole network. Here we can use the conclusion in Chapter 3 to calculate the principal components (PCs) of sequence feature matrices and initialize the PCA module. There are no essential differences in the module's implementation between Scheme I and Scheme II except in the training sample selection and classification

approach.

In Scheme I, the positive training samples are 1000 promoter sequences from EPD [Schmid, Perier et al. 2006], together with 7000 promoter sequences form DBTSS [Suzuki, Yamashita et al. 2002]. All of these promoter sequences are from 250bp upstream to 50bp downstream of the transcription binding sites and those sequences with letter "N" are not included in the positive training sets. The negative training samples are divided into three groups: exon, intron and 3'UTR sequence groups. These non-promoter sequences whose lengths are over 1200bp (compared to the sequence length in EPD) are selected and arranged into 300bp. Each negative training group contains 10000 sequences. Four 3 and 5-mer combined feature matrices are extracted from promoter, exon, intron and 3'UTR sequence groups. One promoter feature matrix and one non-promoter feature matrix from the negative training groups are combined into pairs. The size of each matrix is 1088×18000. PCA is applied on the three matrices which include both promoter and non-promoter information, and 1088 eigenvalues of each covariance matrix are ranked in descending order. According to Equation (3.14), at least six PCs are needed in our network by choosing a cutoff value $t_M = 0.7$. Therefore, six eigenvectors corresponding to the first six eigenvalues of each covariance matrix are selected as PCs.

In Scheme II, all of promoter sequences and non-promoter sequences are classified into a CpG islands related sequence group and a non-CpG islands related sequence

group for feature matrix extraction and principal components generation. In the CpG islands related sequence group, the positive training samples are formed with 5000 promoter sequences from EPD and DBTSS, and the negative training samples include 3000 exon sequences, 3000 intron sequences and 3000 3'UTR sequences. In the non-CpG islands related sequence group, the positive training samples are 8000 promoter sequences from EPD and DBTSS, and in the negative training group, there are 10000 exon sequences, intron sequences, and 3'UTR sequences respectively. The lengths of sequences in Scheme II are also 300bp and the selection approach is the same as in Scheme I. One promoter feature matrix and one non-promoter feature matrix from negative training groups are combined within the CpG islands related training group and non-CpG islands related training group. Therefore, the size of each of the four 3 and 5-mer feature matrices in the CpG islands related training group is 1088×8000, and 1088×18000 in the non-CpG islands related training group. The process of PCA application on feature matrices is the same as in Scheme I and both the CpG islands related group and the non-CpG islands related group have their own principal components. The specific steps of calculating PCs of sequence feature matrices can be referred to in Chapter 3.

A feature vector creation module can extract a 3 and 5-mer feature vector of each input sequence. Each PCA module in Scheme I and Scheme II contains three groups of PCs: PCs of a promoter-exon combined feature matrix, PCs of promoter-intron combined feature matrix, and PCs of promoter-3'UTR combined matrix. These three

groups of PCs construct three new feature spaces. Then, the 3 and 5-mer feature vector from the feature vector creation module are projected into the three spaces and form three new feature vectors as the output of the PCA module. The illustration of a feature vector creation module and PCA module is shown in Figure 4.3.

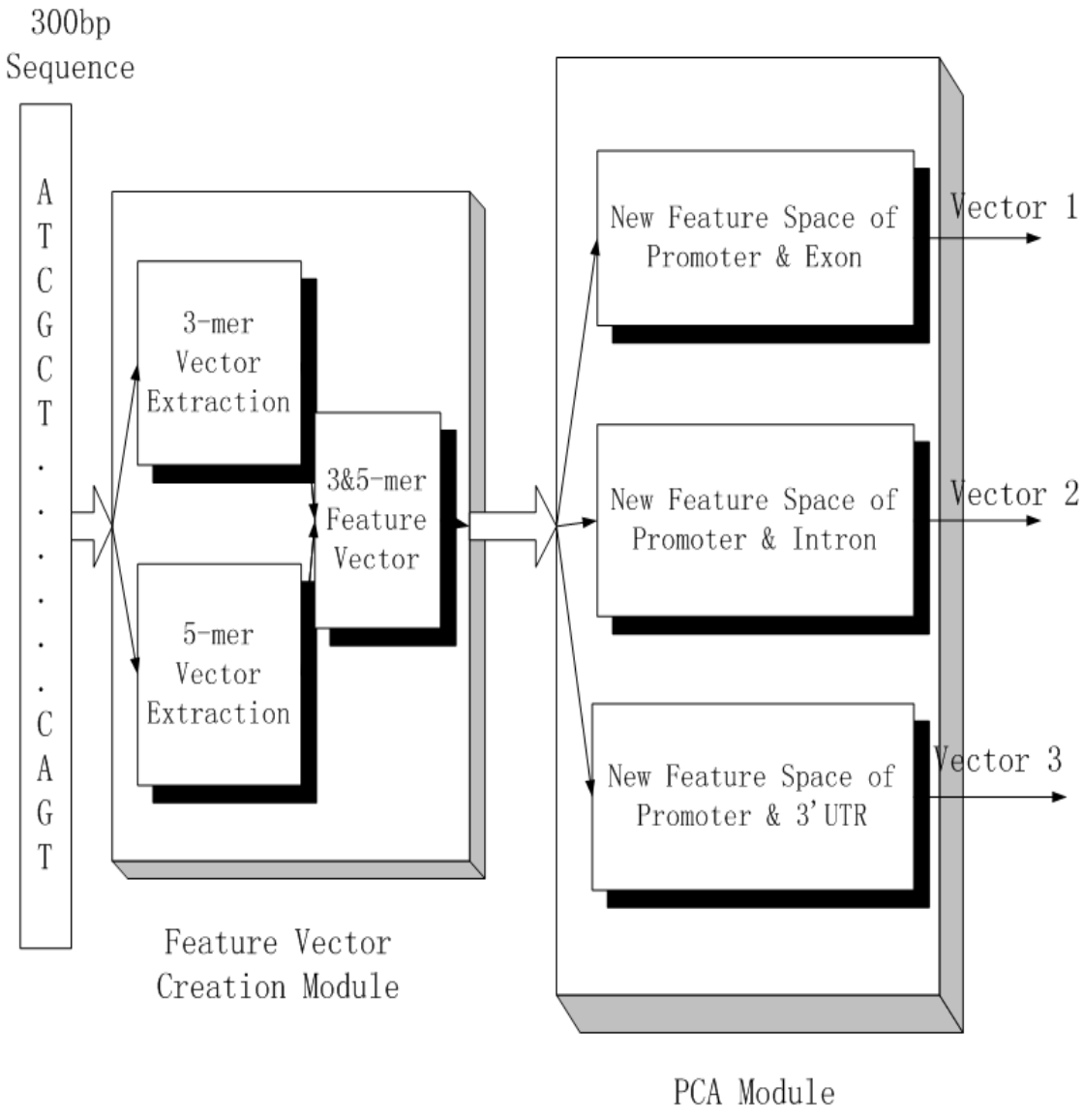

Figure 4.3 The illustration of a feature vector creation module and PCA module

4.2.2 Classifiers for Promoter and Non-Promoter Sequences

There are three classifiers in Scheme I and six classifiers (three for CpG islands related sequences and three for non-CpG related sequences) in Scheme II. These classifiers are built with a back propagation (BP) artificial neural network (ANN) and the theoretical background of BP-ANN is introduced in Chapter 3. In this section, the method of building promoter vs. exon classifier, promoter vs. intron classifier and promoter vs. 3'UTR classifiers is discussed and the parameters of the network are optimized with comparative experiments.

The outputs of the PCA module are three vectors based on promoter-exon feature space, promoter-intron feature space and promoter-3'UTR feature space respectively. In this module, these three vectors are taken as the input of the classifiers and each classifier can classify the input vector into a promoter cluster or a non-promoter cluster.

ANN can learn from training samples and has the ability to recognize new data, so it is ideal for building classifiers by training existing marked sequence samples and recognizing new input sequences. There are three problems that need to be considered in the classifier building process: first, the dimension of the input vector; second, the number of layers in the ANN and the number of neurons in hidden layers; third, transfer function selection of each layer.

As is known, the number of PCs is equal to the dimension of input vectors. As calculated in a previous section, at least six PCs are needed to separate the promoter sequence and the non-promoter sequence in the newly built promoter and non-promoter feature space. We need to further test if the network performance will be better by using higher dimensional vectors. In the model building process, we need to find the optimal size of the neural network. A small network might have good generalization ability, but learn very slowly or not learn at all. So increasing the number of layers or the neurons in hidden layers within a range should improve the performance of the network. However, further increment of the number of layers or the number of neurons will increase the computational complexity and lead to bad performance on generalization. In that case, training samples might be rare compared to the size of the network and thus, over fitting occurs, which means the validation error still increases even when the training error steadily decreases. It is proposed to use the log-sigmoid and tan-sigmoid nonlinear transfer functions in the network but there are no specific regulations to assign these two transfer functions, i.e. the order and times. Due to the above three problems, comparative experiments are designed for achieving better performance and optimizing the parameters of the network.

As the CpG islands feature is not considered in the comparative experiments, the training samples are not divided into CpG islands related and non-CpG islands related group as required for Scheme II. Here we classify the training samples into three groups for three classifiers: promoter and exon vectors based on promoter and exon

feature space, promoter and intron vectors based on promoter and intron feature space, and promoter and 3'UTR vectors based on promoter and 3'UTR feature space. These three groups of vector samples are generated by the sequence feature matrices used for calculating PCs in Scheme I. Three 1088×18000 3 and 5-mer promoter and non-promoter feature matrices are projected into corresponding feature spaces as follows:

$$\begin{bmatrix} p_{1,1} & \cdots & p_{n_0,1} \\ p_{1,2} & \cdots & p_{n_0,2} \\ \vdots & & \vdots \\ p_{1,1088} & \cdots & p_{n_0,1088} \end{bmatrix}' \times \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,18000} \\ \vdots & \vdots & & \vdots \\ c_{1088,1} & c_{1088,2} & \cdots & c_{1088,18000} \end{bmatrix} \tag{4.1}$$

$$= \begin{bmatrix} v_{1,1} & v_{1,2} & \cdots & v_{1,18000} \\ \vdots & \vdots & & \vdots \\ v_{n_0,1} & v_{n_0,1} & \cdots & v_{n_0,18000} \end{bmatrix}$$

where P represents the $1088\times n_0$ principal components vectors of the promoter and non-promoter feature matrix, n_0 is the number of principal components, C represents the 1088×18000 promoter and non-promoter feature matrix and V represents the $n_0\times18000$ vector samples. Additionally, the target outputs corresponding to the training samples are set to ″1″ corresponding to promoter vectors and ″0″ corresponding to non-promoter vectors. Figure 4.4 is an illustration of the classifier training process.

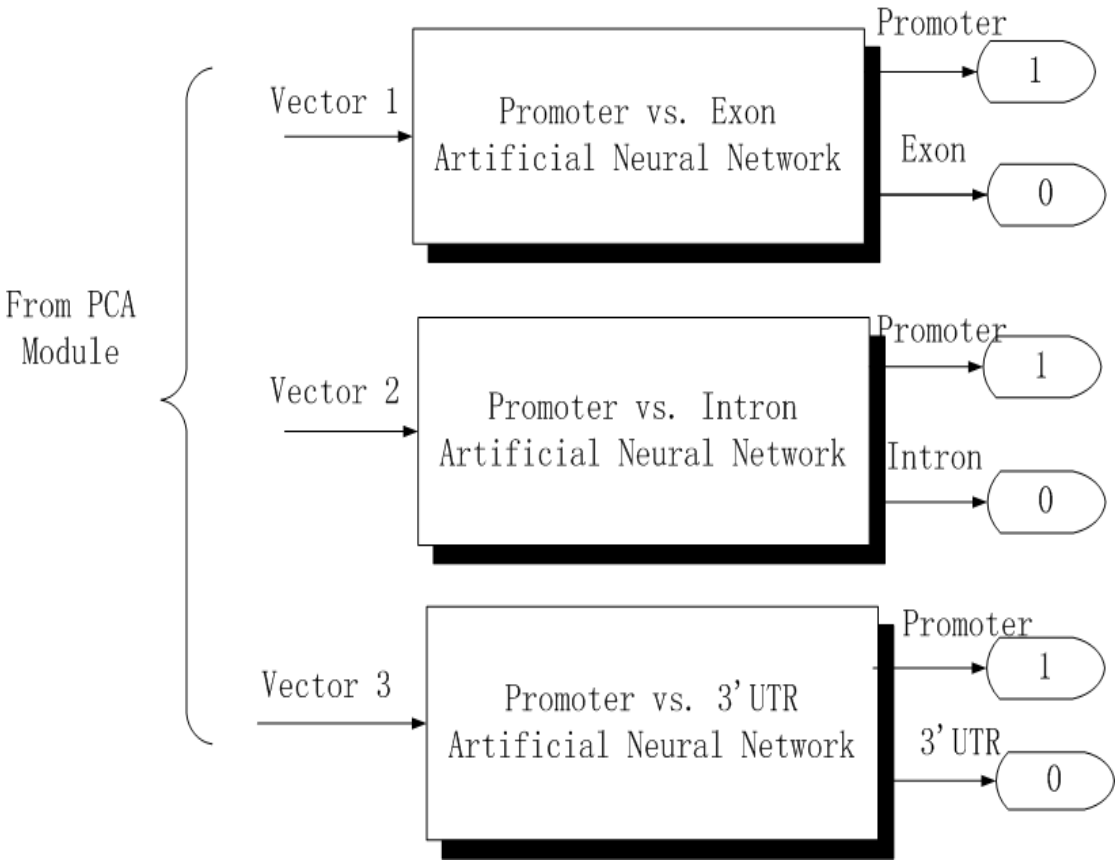

Figure 4.4 The illustration of the classifier training process

The comparative experiments are developed in three steps. In the tests contained in the first step, the dimension of the input vectors and the layers are all fixed at three. The number of neurons in the hidden layer of the three tests are 10, 20 and 20 respectively. In Test 1.1 and Test 1.2, the transfer functions are all set to "tan-sigmoid", "log-sigmoid" and "log-sigmoid". In Test 1.3, the transfer functions are "tan-sigmoid", "log-sigmoid" and "tan-sigmoid". The BP learning algorithm is used in these ANNs, which is introduced in Chapter 3. The training epochs of networks are all set to 30000. In order to test the performances of these three groups of classifiers, five thousand promoter sequences and 6000 non-promoter sequences (2000 of each of three non-promoter datasets) are used. The feature vector generation process of the test set is as follows: first, extract the 3 and 5-mer feature vector of each sequence;

second, project the above vector to three feature spaces built with PCs of promoter & non-promoter feature matrices, and generate three new feature vectors; third, send these new vectors into the corresponding classifiers and obtain the classification results. The training set and test set do not overlap. In each group, three classifiers work together and when the output of two classifiers are over 0.5 (threshold=0.5), the input sequence is reported as a promoter sequence; otherwise it is regarded as a non-promoter sequence. The results of the first step are shown in Table 4.1.

Table 4.1 The experiment results in Step 1 of comparative experiments

Test	Test 1.1	Test 1.2	Test 1.3
Dimension of Input Vectors	3	3	3
No. of Layers	3	3	3
No. of Neurons in Each Layer	(3, 10, 1)	(3, 20, 1)	(3, 20, 1)
Transfer Functions of Each Layer	(tan, log, log)	(tan, log, log)	(tan, log, tan)
TP	4072	4132	4198
FP	1771	1482	1325
Se	0.8156	0.8264	0.8396
Sp	0.6969	0.7360	0.7601

tan: tan-sigmoid log: log-sigmoid

From the above results, we can make following conclusions: first, within a certain range, the more neurons there are in hidden layers, the better performance the network can achieve; second, in the transfer function arrangement of each layer, the "tan-sigmoid", "log-sigmoid" and "tan-sigmoid" shows the greatest advantages.

The tests of Step 2 are developed based on the results of Step 1. The input vector's dimension being fixed on 3 in Step 1 as the training speed of the neural network is influenced by the input vectors: the higher dimension of the input vectors, the lower the calculation speed of the network. The lower dimension of the input vector may influence the classification results but it enables us to achieve the above two important conclusions quicker. Using Function 3.14 we are able to calculate that at least six PCs are needed in the promoter and non-promoter space, so at least 6-dimension input vectors are needed in the classifiers. In comparison, ANNs with a 5-dimension input vector and an 8 dimension input vector are built and tested in the Step 2 experiments. There are three layers of these ANNs and 20 neurons in the hidden layers. The transfer function of each layer is kept as "tan-sigmoid", "log-sigmoid" and "tan-sigmoid". Table 4.2 contains the results of Step 2.

Table 4.2 The experiment results in Step 2 of comparative experiments

Test	Test 2.1	Test 2.2	Test 2.3
Dimension of Input Vectors	5	6	8

No. of Layers	3	3	3
No. of Neurons in Each Layer	(5,20,1)	(6,20,1)	(8,20,1)
Transfer Functions of Each Layer	(tan, log, tan)	(tan, log, tan)	(tan, log, tan)
TP	4327	4429	4360
FP	1077	983	1135
Se	0.8654	0.8858	0.8720
Sp	0.8007	0.8136	0.7934

The results of the experiments that undertaken in Step 2 show that more PCs and a higher dimension of the input vector can not absolutely lead to better classification results. The results of ANN with a 6-dimension input vector obtain the highest sensitivity and specificity in the group. The specificity of ANN with an 8-dimensional input vector shows even lower specificity than the one with a 5-dimensional input vector.

Based on the results obtained from the experiments in Step 2, the experiments in step 3 are designed as 6-dimensional input vector's ANNs. The number of layers in Test 3.1, Test 3.2 and Test 3.3 are three, four and five respectively. The number of neurons of hidden layers are all set to 20. The transfer functions of input and output layers are "tan-sigmoid" and are "log-sigmoid" in hidden layers. The result is shown in Table

4.3.

Table 4.3 The experiment results in Step 3 of the comparative experiments

Test	Test 3.1	Test 3.2	Test 3.3
Dimension of Input Vectors	6	6	6
No. of Layers	3	4	5
No. of Neurons in Each Layer	(6,20,1)	(6,20,20,1)	(6,20,20,20,1)
Transfer Functions of Each Layer	(tan, log, tan)	(tan, log, log, tan)	(tan, log, log, log, tan)
TP	4429	4733	4614
FP	983	602	693
Se	0.8858	0.9466	0.9228
Sp	0.8136	0.8871	0.8694

Test 3.2 obtains a relative high sensitivity and specificity in the comparative experiments, which is 0.9466 and 0.8871 separately. Therefore, the optimized network parameters are used for training the classifiers in the human promoter recognition network. The training sample sequences of the classifiers of the two schemes are those sequences for generating 3 and 5-mer feature matrices and calculating PCs of each scheme. Each classifier which has a 6-dimension input vector

in Scheme I and II is built with four layers of ANN. The numbers of neurons of these four layers are 6, 20, 20 and 1, and the transfer functions are set to “tan-sigmoid”, “log-sigmoid” “log-sigmoid” and “tan-sigmoid” of each layer respectively. The training epochs are 30000.

4.2.3 CpG islands module

A CpG islands module can determine whether the input sequence segment is CpG islands related or not by two criteria: GC percentage (GCp) and observed/expected CpG ratio (o/e) which are calculated according to Equations (2.1) and (2.2). Given an input sequence, if it’s GCp and o/e are over 0.5 and 0.6, the module will report it as a CpG islands related sequence.

A CpG islands module plays different roles in the two schemes. In Scheme I, the CpG module I gives a score for each input sequence segment: “1” for a CpG islands related segment and “0” for a non-CpG islands related segment. The score from a CpG islands module together with the outputs from the classifier module of each input sequence segment will be processed in a data processing module.

In Scheme II, the criteria used in the CpG islands module are the same as the criteria in Scheme I. Each sequence from the CpG islands module is classified as a CpG islands related or non-CpG islands related sequence instead of obtaining a score. And

then, according to the classification, the sequence will be further processed by following either the CpG islands related branch or non-CpG island related branch.

4.2.4 Data Processing and Prediction of TSS

The data processing module and the transcription start site (TSS) prediction module obtain the final results of the human promoter recognition network: the data processing module reports the windows of the potential promoter regions and the TSS prediction module predicts the exact location of TSS.

In Scheme I, the outputs of the three promoter vs. non-promoter classifiers and the scores from the CpG islands module are sent to the data processing module. The threshold of each classifier is set to 0.4, and if the outputs of two of three classifiers are over the predefined threshold, the data processing module will sum up the outputs of the three classifiers and the score from the CpG island module. If the sum is over 2.2, the data processing module will report the window as the potential promoter region. In the TSS prediction module, a promoter region is identified if the number of consecutive windows is more than 30 and the consecutive windows are defined here if the offset of two windows is less than 300bp. The predicted TSS is the location that contains the maximum likelihood.

In Scheme II, the predefined thresholds of classifiers are set to 0.4, and as the promoter vs. non-promoter classifiers are trained with CpG islands related sequences

and non-CpG islands related sequences separately, the data processing module report a potential promoter region window when two of three classifies' outputs are more than 0.4 and the sum of scores from three classifiers is over 1.2. The rules of predicting the location of TSS are the same as the ones in Scheme I.

4.3 Performance Evaluation of Scheme I and Scheme II

In order to evaluate the performance of Scheme I and Scheme II, 5000 promoter sequences and 6000 non-promoter sequences as used in Section 4.2.2 for testing classifier performance.are used to form Test Set 1. Additionally, three Homo Sapiens chromosome 22 genomic sequences—NT_028395.3, NT_011519.10 and NT_011521.4 are extracted from GeneBank to form Test Set 2. In Test Set 1, TP is counted when a true promoter sequence is recognized, otherwise FP is counted. In Test Set 2, we adopt the same evaluation method as DragonGSF: when one or more predictions fall in the region of [-2000, +2000] relative to a TSS, a TP is counted. All predictions which fall on the annotated part of the gene on the region [+2001, EndofTheGene] are counted as FP. Other predictions are not considered in counting TP and FP. The comparison results of Test Set 1 and Test Set 2 are listed in Table 4.4 and Table 4.5.

Table 4.4 The comparison results of two schemes on Test Set 1

Scheme	Scheme I	Scheme II
TP	4763	4604
FP	388	646
Se	0.9526	0.9208
Sp	0.9246	0.8769

Table 4.5 The comparison results of two schemes on Test Set 2

Scheme		Scheme I	Scheme II
NT_028395.3	TP	1	1
	FP	1	1
NT_011519.10	TP	18	17
	FP	6	15
NT_011521.4	TP	2	3
	FP	1	10
Se		0.4038	0.4038
Sp		0.7241	0.4468

According to the above results, the networks of Scheme I and Scheme II achieve comparable sensitivities, while Scheme I obtains the higher specificity in both Test Set 1 and Test Set 2. Although Scheme II has more a complex structure than Scheme I, the false positives produced by it are three times as many as for Scheme I. As there are

not enough CpG islands related non-promoter training samples for the classifiers of Scheme II, the network is more likely to identify a CpG islands related sequence as a promoter sequence, which leads to more false positives. In conclusion, the network of Scheme I obtains better overall test results, therefore, it is adopted as the HPR-PCA model.

Chapter 5 Results and Discussion

To evaluate the ability of HPR-PCA to predict TSSs in human DNA sequences, we compare HPR-PCA with three well-known existing promoter prediction systems: DragonGSF, Epoin and FirstEF. The methodologies of these systems are reviewed in Chapter 2. In HPR-PCA, the network parameters use the default setting of Scheme I. The comparison results are obtained based on three different test datasets.

5.1 Test Results and Discussion

Test set 1 consist of four human genomic sequences from GenBank with a total length of 0.95Mb and 14 known TSS. These sequences are tested using promoter prediction systems mentioned in Chapter 2 and the results are available for comparison. Table 5.1 shows an overview of the four selected genomic sequences.

Table 5.1 Description of the large genomic sequences in Test set 1.

Accession number	Description	Length (bp)	Number of TSS
L44140	Homo sapiens chromosome X region from filamin (FLM) gene to glucose-6-phoshate dehydrogenase (G6PD) gene. There are 13 known and six candidate genes in the sequence.	219447	11
D87675	Homo sapiens DNA for amyloid precursor protein	301692	1
AF017257	Homo sapiens chromosome 21-derived BAC containing erythroblastosis virus oncogene homolog 2 protein (ets-2) gene	101569	1
AC002368	Homo sapiens Xq 28 BAC PAC and cosmid clones containing FMR2 gene	324816	1
Total		947524	14

Three promoter prediction systems — DragonGSF, Eponine and FirstEF are selected to compare the performance in Test set 1. A promoter region is counted as a true positive (TP) if TSS is located within the region, or if a region boundary is within 200bp 5' of

such a TSS. Otherwise the predicted region is counted as a false positive (FP). The results and comparisons are listed in Table 5.2 and Table 5.3.

Table 5.2 Performance comparison of four prediction systems for Test set 1 (I)

Accession number	System	*TP*	*FP*	Coverage (%)
L44140	DragonGSF	6	11	54.5
	FirstEF	6	11	54.5
	Eponine	6	12	54.5
	HPR-PCA	6	11	54.5
D87675	DragonGSF	1	1	100
	FirstEF	1	0	100
	Eponine	1	1	100
	HPR-PCA	1	0	100
AF017257	DragonGSF	1	0	100
	FirstEF	1	0	100
	Eponine	1	3	100
	HPR-PCA	1	0	100
AC002368	DragonGSF	1	2	100
	FirstEF	1	1	100
	Eponine	1	0	100
	HPRPCA	1	0	100

Table 5.3 Performance comparison of four prediction systems for Test set 1 (II)

System	*TP*	*FP*	S_e	S_p
DragonGSF	9	14	0.6429	0.3913
FirstEF	9	12	0.6429	0.4286
Eponine	9	16	0.6429	0.3600
HPR-PCA	9	11	0.6429	0.4500

For Test set 1, all four systems predict 9 of the 14 TSSs of the four human genomic sequences and achieve the equivalent sensitivity. However, HPR-PCA produces the least false positives and the highest specificity. The result of FirstEF is comparable as it only produces one more false prediction than HPR-PCA.

In Test 2, the Chromosome 22 sequence and its annotation data (http://www.sanger.ac.uk/HGP/Chr22) are adopted. The sequence with a total length of 34.75Mbp is tested and the results are evaluated by 393 annotated TSSs.

The comparative systems are the three systems used in Test set 1, but this time we use the same evaluation method as DragonGSF: only a TP is counted if one or more predictions falls in the region of [-2000, +2000] relative to a TSS. All predictions which fall on the annotated part of the gene on the region [+2001, End of The Gene] are

counted as FP. Other predictions are not considered in counting TP and FP. Experiment results of DragonGSF, FirstEF and Eponine were from [Bajic and Seah 2003].. Table 5.4 shows the result on Test set 2.

Table 5.4 Performance comparisons of four prediction systems for Test set 2.

System	*TP*	*FP*	S_e	S_p
DragonGSF	269	69	0.6844	0.7959
FirstEF	331	501	0.8422	0.3978
Eponine	199	79	0.5064	0.7158
HPR-PCA	301	65	0.7659	0.8224

In Test set 2, the sensitivity and specificity of HPR-PCA are 0.7659 and 0.8224 respectively, FirstEF obtains the highest sensitivity, but it produces the most false positives, which leads to the specificity of FirstEF being only half of HPR-PCA. DragonGSF retains a good balance between sensitivity and specificity. If we tune the parameters of HPR-PCA and the sensitivity is adjusted to 0.7277, which is closer to the sensitivity of DragonGSF (0.6844), the specificity of HPR-PCA will achieve 0.8910, which is much higher than that for DragonGSF (0.7959). Although Eponine also produce low false positives, it only predicts 199 of 393 TSSs, which is far less than 301 TSSs obtained by HPR-PCA. In conclusion, HPR-PCA obtains better overall results on Test set 2, and on the prediction of long human genomic sequences, HPR-PCA is more

competitive among these four prediction systems.

In Test 3, seven Homo sapiens chromosome 22 genomic contigs were extracted from GenBank with a total length of 11.56Mbp and 94 TSSs in the forward strands. We test these sequences because the annotations of chromosome 22 provided by GenBank has different number of TSSs with the one in test 2, so the results are more convincing. Table 5.5 shows an overview of these genomic sequences.

Table 5.5 Description of the large genomic sequences in the test set 3.

Contig	Description	Length (bp)	Number of TSS
NT_028395.3	Homo sapiens chromosome 22 genomic sequence	647850	6
NT_011519.19		3661581	38
NT_011521.4		830225	8
NT_011523.11		4248192	21
NT_011525.7		1384186	7
NT_019197.5		320440	5
NT_011526.6		464629	9
Total		11557103	94

On Test set 3, we compare HPR-PCA with DragonGSF because DragonGSF is the only online system which can accept relatively longer sequences for those systems compared in our analysis. In order to get fair results, for those sequences which are longer than 1,000,000bp (the limitation of a file in the DragonGSF web tool), we arrange them to be equal to, or less than, 1,000,000bp each before sending them to HPR-PCA and DragonGSF. The evaluation criteria of Test set 3 is the same as the one in Test set 2 and the test results are shown in Table 5.6 and Table 5.7.

Table 5.6 Performance comparison of two prediction systems for Test set 3 (I)

Accession number	System	*TP*	*FP*	Coverage (%)
NT_028395.3	DragonGSF	1	1	16.7
	HPR-PCA	1	1	16.7
NT_011519.19	DragonGSF	15	8	39.5
	HPR-PCA	18	6	39.5
NT_011521.4	DragonGSF	2	1	25.0
	HPR-PCA	2	1	25.0
NT_011523.11	DragonGSF	15	4	71.4
	HPR-PCA	16	5	76.2
NT_011525.7	DragonGSF	2	0	28.6
	HPR-PCA	2	0	28.6

NT_019197.5	DragonGSF	3	2	42.9
	HPR-PCA	5	1	71.4
NT_011526.6	DragonGSF	6	5	66.7
	HPR-PCA	6	5	66.7

Table 5.7 Performance comparison of two prediction systems for Test set 3 (II)

System	*TP*	*FP*	S_e	S_p
DragonGSF	44	21	0.4681	0.6769
HPR-PCA	50	19	0.5319	0.7246

On Test set 3, HPR-PCA again achieves a better result: the sensitivity and specificity are 0.5319, 0.4681 and 0.7246, 0.6769 of HPRPCA and DragonGSF respectively. Although the annotation of chromosome 22 sequences of Test set 3 is different from the ones on Test set 2, HPR-PCA still shows the advantages of genome wide promoter prediction practice.

Compared to other currently favored promoter prediction systems, HPR-PCA uses rebuilding sequence features selected by PCA instead of those directly taken from DNA sequences. This new feature selection concept is successfully embedded in the optimized promoter prediction network and proved by three test sets. The experiment results of Test set 1 and 3 show that HPR-PCA can reduce the false positive rate which

leads to higher specificity. Predictions on the genome sequence of chromosome 22 made by HPR-PCA are competitive for both specificity and sensitivity. DragonGSF reports good prediction performance on the whole human genome sequence, but it uses TRANSFAC database [V. Matys, O.V. Kel-Margoulis et al. 2006]which includes binding site information only available for known promoters. From this point, HPR-PCA has the advantage to discover unknown promoters without prior information. In conclusion, all the test results indicate that the ANN-built human promoter recognition network—HPR-PCA, which embeds the most informative sequence features selected by PCA algorithm, performs well on genome wide promoter recognition tasks.

Chapter 6 Conclusion and Future Work

6.1 Conclusion and Discussion

The topic of this thesis is human promoter prediction based on principal component analysis (PCA). Promoter prediction is one of the most important problems in DNA sequence analysis. An overview of promoter recognition is shown in the Introduction and Literature Review, which include the significance of promoter prediction, the important features of promoter sequences, and the summary of modeling methodologies used by some existing promoter prediction models.

Chapter 3 presents the application of PCA on the sequence feature selection process, which is a new proposal for promoter feature selection application. In order to find the most discriminative features, n-mer ($n = 3,4,5$) feature matrices are extracted from promoter and exon sequences. PCA applies to seven different n-mer combination promoter-exon feature matrices and the first three PCs of each matrix are selected. In order to test the discriminability of seven feature groups, seven classifiers are built with

three-layer ANN and trained with a BP algorithm. Finally, the network trained with 3 and 5-mer combined feature matrices obtains the highest sensitivity and specificity, so the 3 and 5-mer combined feature is used to build classifiers in HPR-PCA.

Two proposed schemes of HPR-PCA are introduced in Chapter 4. Sub-modules of the human promoter recognition network are divided into four main groups. The main difference between Scheme I and Scheme II is the implementation of the CpG islands module: Scheme I gives a mark to each sequence from the CpG islands module; and Scheme II divides sequences to CpG islands related and non-CpG islands related groups for further processing, which determines the training sequences for generating PCs in this scheme need to be classified into CpG islands related and non-CpG islands related groups. In both schemes, 3 and 5-mer promoter and non-promoter combined feature matrices are extracted for generating PCs and building new promoter and non-promoter feature spaces. Three promoter vs. non-promoter classifiers are built based on the ANN. The structure and parameters of the classifiers are optimized by comparative experiments. In the comparison of the two schemes, Scheme I achieves better results on two test sets so it is adopted as the model for HPR-PCA.

In Chapter 5, three test sets are formed to evaluate the performance of HPR-PCA， and three other promoter prediction techniques: DragonGSF, Epoin and FirstEF are used to compare with HPR-PCA. In the end, HPR-PCA achieves the best overall results on all of the three test sets among these four systems. The prediction result is also a powerful

verification that the PCA algorithm performs efficiently on feature selection, which is one of the most important tasks in the promoter recognition field.

6.2 Future work

As emphasized in the thesis, discriminative features are the crucial elements of promoter prediction systems. HPR-PCA embeds context features and the CpG islands signal feature into the model, but it does not consider the position information of features in promoter sequences. Structure features of sequences, such as flexibility rigidity and bendability, which are extracted from three-dimensional DNA structures [Pedersen, Pierre Baldi et al. 1998], can generate the profile of promoter sequences. As the structure profile originates from the sequential structure of the DNA, rather than the general nucleotide composition, it could provide supplementary information in promoter prediction practice of HPR-PCA.

Structure features of promoters are discussed in recent promoter prediction techniques [Sonnenburg, Zien et al. 2006] [Ohler, Stemmer et al. 2000]. Flexibility, as an important structure feature, has been examined in several organisms [Pedersen, Pierre Baldi et al. 1998]\ [Kanhere and Bansal 2005] [Tirosh, Berman et al. 2007], and it is suggested that it can influence the activities of transcription binding sites (TFBSs) [Fukue, Sumida et al. 2004]. The flexibility profiles of promoters have distinctive mechanical properties [Scherf, Klingenhoff et al. 2000], so it is applicable to use flexibility as a feature to distinguish promoter sequences from non-promoter sequences.

There are two widely used models for calculating flexibility of DNA sequences: one is the trinucleotide model based on DNase I cutting frequencies [Brukner, Sanchez et al. 1995], and the other is the tetranucleotie model from molecular orbital calculations [Packer, Dauncey et al. 2000]. We can use one of these models to create the flexibility profiles and extend the HPR-PCA model by embedding the structure features module in parallel with the CpG island module and the classifiers module, or combining the flexibility profiles with context features for classification. The increase in the true positive rate and the decrease in the false positive rate can be expected as one more dimensional feature is added in the decision network of the promoter prediction model.

Bibliography

Antequera, F. and A. Bird (1993). "Number of CpG islands and genes in human and mouse " Proc Natl Acad Sci U S A. **90**(24): 11995-11000.

Attwood, T. K. and D. J. Parry-Smith (1999). Introduction to Bioinformatics. London, Longman.

Bajic, V. B., A. Chong, et al. (2002). "An Intelligent System for Vertebrate Promoter Recognition." IEEE Intelligent Systems **17**: 64-70.

Bajic, V. B. and S. H. Seah (2003). "Dragon Gene start Finder: An Advanced System for finding Approximate Location of the start of Gene Transcriptional Units." Genome Res. 2003 **13**: 1923-1929.

Bajic, V. B., S. H. Seah, et al. (2003). "Computer model for recognition of functional transcription start sites in RNA polymerase II promoters of vertebrates." Journal of Molecular Graphics and Modelling **21**: 323-332.

Bajic, V. B., S. L. Tan, et al. (2004). "Promoter prediction analysis on the whole human genome." Nature Biotechnology **22**: 1467-1474.

Brukner, I., R. Sanchez, et al. (1995). "Sequence-dependent bending propensity of DNA as revealed by DNase 1: parameters for trinucleotides." The EMBO Journal **14**(8): 1812-1818.

Bucher, P. (1990). "Weight matrix descriptions of four eukaryotic rna Polymerase II promoter elements derived from 502 unrelated promoter sequences." Molecular Biology **212**: 563-589.

Burge, C. and S. Karlin (1997). "Prediction of complete gene structures in human

genomic DNA." Journal of Molecular Biology **268**: 78-94.

Burges, C. J. C. (1998). "A Tutorial on Support Vector Machines for Pattern Recognition." Data Mining and Knowlede Discovery **2**: 121-167.

Chen, C. B. and T. Li (2005). "A hybird neural network system for prediction and recognition of promoter regions in human genome." Journal of Zhejiang University Science **6B**(5): 401-407.

Chen, Q. K., G. Z. Hertz, et al. (1997). "PromFD 1.0: a computer program that predicts eukarotic pol II promotersusing strings and IMD matrices." Comput. Applic. Biosci. **13**(1): 29-35.

Chen(Ed.), Y.-P. P., Ed. (2005). Bioinfomatics Technologies, Springer.

Davuluri, R. V., I. Grosse, et al. (2001). "Computational identification of promoters and first exons in the human genome." Nature Genetics **29**: 412-417.

Deonier, R. C., S. Tavare, et al. (2005). Computational Genome Analysis: An Introduction, Springer.

Down, T. A. and T. J. P. Hubbard (2002). "Computational Detection and Location of Transcription Start Sites in Mammalian Genomic DNA." Genome Research **12**: 458-461.

Fickett, J. W. and A. G. Hatzigeorgiou (1997). "Eukaryotic Promoter Recognition." Genome Research **7**: 861-878.

FitzGeralad, P. C., A. Shlyakhtenko, et al. (2004). "Clustering of DNA Sequences in Human Promoters." Genome Res. **14**: 1562-1574.

Fukue, Y., N. Sumida, et al. (2004). "Core promoter elements of eukaryotic genes hace

a hihly distictive mechanical property." Nucleic Acids Research **32**: 5834-5840.

Gardiner-Garden, M. and M. Frommer (1987). "CpG islands in vertebrate genomes." Journal of Molecular Biology **196(2)**: 261-282.

Jolliffe, I. T. (1986). Principal Component Analysis. New York, Springer.

Kanaya, S., Y. Yamada, et al. (1999). "Studies of codon usage and tRNA genes of 18 unicellular organisms and quantification of Bacillus subtilis tRNAs: Gene expression level and species-specific diversity of codon usage based on multivariate analysis." Genes & Dev **238**: 143-155.

Kanhere, A. and M. Bansal (2005). "Structural properties of promoters: similarities and differences between prokaryotes and eukaryotes." Nucleic Acids Research **33**(10): 3165-3175.

Kaufmann, J., C. P. Verrijzer, et al. (1996). "CIF, an essential cofactor for TFIID-dependent initiator fuction." Genes & Dev **10**: 873-886.

Krogh, A. and M. Brown (1994). "Hidden Markov models in Computational biology applications to protein modeling." Journal of Molecular Biology **235**(5): 1501-1531.

Kulp, D., D. Haussler, et al. (1996). A generalized hidden Markov model for the recognition of human genes in DNA. Proc Int Cong Intell Syst Mol Biol. **4:** 134-142.

Lander, E.S., L. M. Linton, et al.(2001). "Initial sequencing and analysisi of the human genome." Nature Biotechnology **209**: 860-921.

Medigue, C., T. Rouxel, et al. (1991). "Evidence for horizontal gene transfer in

Escherichia coli speciation." Journal of Molecular Biology **222**: 851-856.

Murakami, K., Y. Ohta, et al. (2000). "A Transcription Regulatory Region Analysis System." Genome Infromatics **11**: 297-298.

Ohler, U., G. Stemmer, et al. (2000). Stochastic Segment Models of Eukaryotic Promoter Regions. Pacific Symposium on Biocomputing. **5:** 377-388.

Ostendorf, M., V. Digalakis, et al. (1995). "From HMMs to segment Models: a unified view of stochastic modeling for speech recognition." IEEE Transactions on Speech and Audio Processing **4**(360-378).

Packer, M. J., M. P. Dauncey, et al. (2000). "Sequence-dependent DNA Structure: Tetranucleotide Conformational Maps." journal of Molecular Biology **295**: 85-103.

Pedersen, A. G., P. Baldi, et al. (1999). "The Biology of Eukaryotic Promoter Prediction —a Review." Computers and Chemistry **23**(3): 191-207.

Pedersen, A. G., Pierre Baldi, et al. (1998). "DNA Structure in Human RNA polymerase II Promoters." Journal of Molecular Biology **281**: 663-673.

Pesole, G., S. Liuni, et al. (2001). "UTRdb and UTRsite: specialized databases of sequences and functional elements of 5' and 3' untranslated regions of eukaryotic mRNAs. Update 2002." Nucleic Acids Research **30**: 6.

Ponger, L. and D. Mouchiroud (2002). "CpGProD: identifying CpG islands associated with transcription start sites in large genomic mammalian sequences." Bioinformatics **18**: 631-634.

Prestridge, D. S. (1995). "Predicting Pol II promoter Sequences using Transcription

Factor Binding Sites." Journal of Molecular Biology **249**: 923-932.

Raychaudhuri, S., Ed. (2006). Computational Text Analysis for Functional Genomics and Bioinformatics, Oxford University Press Inc., New York.

Raychaudhuri, S., J. M. Stuart, et al. (2000). "Principal analysis to cummarize microarray experinents: application to sporulation time series." Pac Symp Biocomput.: 455-466.

Raychaudhuri, S., J. M. Stuart, et al. (2000). "Pattern recognition of genomic features with microarrays: site typing of Mycobacterium tuberculosis strains." Proc Int Conf Intell Syst Mol Biol. **8**: 286-295.

Reese, M. G. (2001). "Application of a time-delayed neural network to promoter annotation in the drosophila melanogaster genome." Computers & Chemistry **26**: 294-302.

Rojas, R. (1996). Neural Networks: A Systematic Introduction, Springer-Verlag, Berlin.

Saxonov, S., I. Daizadeh, et al. (2000). "EID: the Exon–In tron Database — an exhaustive database of protein-coding intron-containing genes." Nucleic Acids Research **28**(1): 185-190.

Scherf, M., A. Klingenhoff, et al. (2000). "Highly Specific Localization of Promoter Regions in Large Genomic Sequences by PromoterInspector: A Novel Context Analysis Approach." Journal of Molecular Biology **297**: 599-606.

Schmid, C. D., R. Perier, et al. (2006). "EPD in its twentieth year: towards complete promoter coverage of selected model organisms." Nucleic Acids Research **34**:

82-85.

Scholkopf, B., A. Smola, et al. (1998). "Nonlinear component analysis as a kernel eigenvalue problem." Neural Computation **10**(5): 1299-1319.

Shlens, J. (2005). A Tutorial on Principal Component Analysis. http://www.snl.salk.edu/~shlens/pub/notes/pca.pdf

Smale, S. T. and J. T. Kadonaga (2003). "The RNA Polymerase II Core Promoter." Annu. Rev. Biochem. **72**: 449-479.

Smith, L. I. (2002). A tutorial on Principal Components Analysis. http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf

Solovyev, V. V. and K. S. Makarova (1993). "A novel method of protein sequence classification based on oligopeptide frequency analysis and its application to search for functional sites and to domain localization." Comput. Applic. Biosci. **9**(1): 17-24.

Sonnenburg, S., A. Zien, et al. (2006). "ARTS: accurate recognition of transcription starts in human." Bioinformatics **22**(14): e472-e480.

Stormo, G. D. and D. Haussler (1994). Optimally parsing a sequence into different classes based on multiple types of evidence. Int Conf Intell Syst Mol Biol. , Stanford, California, USA.

Suzuki, Y., R. Yamashita, et al. (2002). "DBTSS: DataBase of human Transcriptional Start Sites and full-length cDNAs." Nucleic Acids Research **30**: 328-331.

Tipping, M. E. (2001). "Sparse Bayesian Learning and the Relecance Vector Machine." Journal of Machine Lerning Research **1**: 211-244.

Tirosh, I., J. Berman, et al. (2007). "The pattern and evolution of yeast promoter bendability." Trends in Genetics **23**(7): 318-321.

V. Matys, O.V. Kel-Margoulis, et al. (2006). "TRANSFAC and its module TRANSCompel: transcriptional gene regulation in eukaryotes." Nucleic Acids Research **34**: 108-110.

Wu, S., X. Xie, et al. (2007). "Eukaryotic promoter prediction based on relaive entropy and positional information." Physicl Review E **75**: 041908 1-7.

Xie, X., S. Wu, et al. (2006). "PromoterExplorer: an efective promoter identificaiton method based on the Adaboost algorithm." Bioinformatics **22**: 2722-2728.

Yeung, K. Y. and W. L. Ruzzo (2001). "Principal Component Analysis for clustering gene expression data." Bioinformatics **17**(9): 763-774.

Zhang, M. Q. (1998a). "A discrimination study of human core-promoters in silico." Proc. Pacific Symp. Biocomputing 1998. R. Altman, A. K. Donker, L. Hunter and T. E. Klein. Worl Scientific, Singapore**:** 240-251.

List of Publication

Li, X. M., Y. M. Liu and H. Yan. (2008). “Eukaryotic Promoter Predication Based on Principal Component.” International Multiconference of Engineers and Computer Scientists 2008. Hong Kong.

Li, X. M., J. Zeng and H.Yan. (2008). "PCA-HPR: A New Method of Human Promoter Recognition Based on Principle Component Analysis." Bioinofrmation 2(9): 373-378

Liu, Y. M., X. M. Li, H. Yan. (2008). “Codon Relation Analysis for Promoter Recognition Using Indepent Component Analysis", Journal of Information & Computational Science 5(1): 33-39