# Promoter Recognition in Human Genome

A Project Report Submitted in the partial fulfillment of the requirements for the award of degree of

Master of Technology
in
Artificial Intelligence

By

**J. Chandra Shekar**


Department of Computer and Information Sciences
University of Hyderabad
Hyderabad, India

April, 2010

# CERTIFICATE

This is to certify that the project work entitled "**Promoter Recognition in Human Genome**" being submitted to University of Hyderabad by **J. Chandra Shekar** (Reg. No. 08MCMI17), in partial fulfillment for the award of the degree of Master of Technology in Artificial Intelligence, is a bonafide work carried out by him under my supervision.

Dr. T. Sobha Rani
Project Supervisor,
Department of CIS,
University of Hyderabad

Head of Department,
Department of CIS,
University of Hyderabad

Dean,
School of MCIS,
University of Hyderabad

*To,*

*My Parents*

# Acknowledgments

I would like to express my sincere gratitude to my supervisor **Dr. T. Sobha Rani**. Her valuable suggestions and keen personal interest throughout the progress of my course of research.Her vast experience, profound knowledge and willingness have been a constant source of inspiration for me throughout this project work. she encouraged, supported, corrected and guided me during the project. The project has been a learning and growing experience for me. Working under her guidance I learned many things.

I am also grateful to the head of department **Prof. Arun Agarwal**, for providing excellent facilities and such a nice atmosphere for doing this project.I would like to extend my sincere thanks to Dean of School of Mathematics, Computers and Information Sciences for his valuable cooperation.

I am extremely thankful to **Prof. Bapi Raju** for his support and valuable advises throughout the entire course during the period of two years of M.Tech in Department of Computers and Information Sciences.

I convey my heartfelt thanks to AI Lab staff for their help in completing the project work successfully.

I convey my heartfelt thanks to Srinivas Research scholar, Shankara Rao, Avinash and my labmates.

Finally I would like to take this opportunity to thank my beloved parents, who constantly supported and encouraged me throughout my life with their love and affection.

With sincere regards
Chandra Shekar . J

# Abstract

A promoter is a specific region of DNA that facilitates the transcription of a particular gene. Promoters are typically located near the genes they regulate, on the same strand and upstream (towards the $5'$ region of the sense strand). Promoters contain specific DNA sequences and response elements which provide binding sites for RNA polymerase and for proteins called transcription factors that recruit RNA polymerase. Promoter prediction programs (ppps) are computational models which aim at identifying the promoter regions in a genome. The main approaches of the promoter prediction are either assigning scores to all single nucleotides to identify TSS or identifying a promoter region without providing scores to all nucleotides. In this project n-gram features are extracted and used in promoter prediction. Here a systematic study is made to discover the efficacy of n-grams (n=2,3,4,5) as features in promoter prediction problem. Neural network classifiers with these n-grams as features are used to identify promoters in a human genome. In this case for n=4 we are getting optimal values.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

The human genome is made up of all of the DNA in the chromosomes as well as that in mitochondria. The human genome is the genome of Homo sapiens, which is stored on 23 chromosome pairs. The haploid human genome occupies a total of just over 3 billion DNA base pairs. Twenty-two of these are autosomal chromosome pairs, while the remaining pair is sex-determining. The haploid human genome contains 23,000 protein-coding genes, far fewer than had been expected before its sequencing. In fact, only about 1.5% of the genome codes for proteins, while the rest consists of non-coding RNA genes, regulatory sequences, introns, and (controversially named) ”junk” DNA (inter gene regions). The general scheme of the shown in figure 1.1.

## 1.1 Promoter

A promoter is a region of DNA that facilitates the transcription of a particular gene. Promoters are typically located near the genes they regulate, on the same strand and upstream (towards the $5'$ region of the sense strand). Promoters contain specific DNA sequences and response elements which provide a binding site for RNA polymerase and for proteins called transcription factors that recruit RNA polymerase.

- In bacteria, the promoter is recognized by RNA polymerase and an associated sigma factor, which in turn often brought to the promoter DNA by an activator protein binding to its own DNA binding site nearby.
- In eukaryotes, the process is more complicated, and at least seven different factors are necessary for the binding of an RNA polymerase II to the promoter.

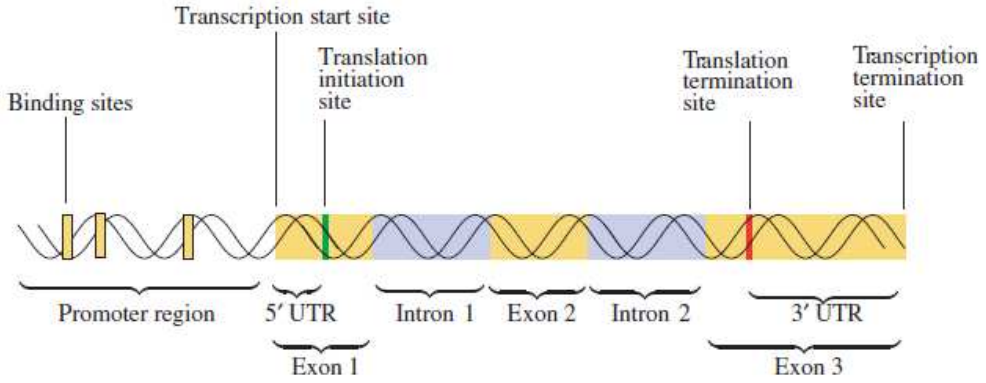


Figure 1.1: A schematic representation of the locations of the promoter region, TFBSs, exons, introns and $3'$utr regions.

## 1.2 Structure of a genome

Promoter of the genes that transcribe relatively large amount of mRNA have similar structure. They have a TATA sequences about 30bp upstream from the site where transcription begins, one or more promoter elements fixture upstream. The function of the region can be analyzed by determining its bases which are necessary for efficient transcription. Once the transcription of gene is confirmed one uses restriction enzymes to make specific deletion in the gene or in the regions surrounding it. The main elements of the promoter are shown in the figure 1.1.

### 1.2.1 Promoter region

The non-coding nucleotide sequence prior to the transcription start region that is characterized by the presence of a number of conserved or consequence sequences is general termed as a promoter.

### 1.2.2 Introns

An intron is a DNA region within a gene that is not translated into protein. These non-coding sections are transcribed to precursor mRNA (pre-mRNA) and some other RNAs (such as long non coding RNAs), and subsequently removed by a process called splicing during the processing to mature RNA. Introns are common in eukaryotic pre-mRNA, but in prokaryotes they are only found in tRNA and rRNA. Introns have variable length and alternate with exons in intron-containing genes. Introns contain several short sequences that are

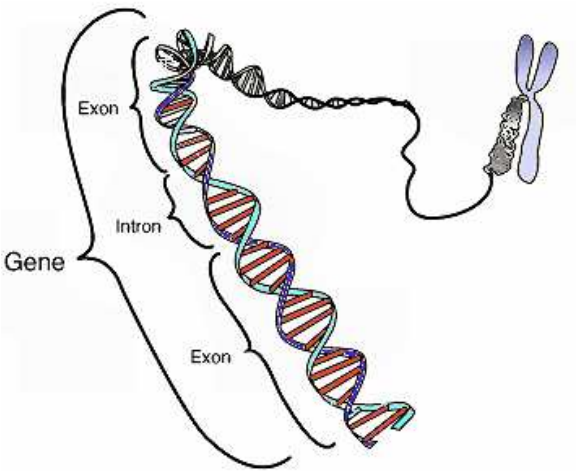


Figure 1.2: location of introns and exons within a gene.

important for efficient splicing, such as acceptor and donor sites at either end of the intron as well as a branch point site, which are required for proper splicing by the Spliceosome. Some introns are known to enhance the expression of the gene that they are contained in by a process known as intron-mediated enhancement (IME). Some introns, such as the Group I and Group II introns, after transcription possess ribozyme activity, enabling them to catalyze their own splicing out of a primary RNA transcript. The main element of the introns are shown in fig.1.2.

### 1.2.3 Exons

An exon is a nucleic acid sequence that is represented in the mature form of an RNA molecule after either portions of a precursor RNA (introns) have been removed by cis-splicing or by two or more precursor RNA molecules have been ligated by trans-splicing. The mature RNA molecule can be a messenger RNA or a functional form of a non-coding RNA such as rRNA or tRNA. Depending on the context, exon can refer to the sequence in the DNA or its RNA transcript. Each exon contains part of the open reading frame (ORF) that codes for a specific portion of the complete protein.

### 1.2.4 $3'$Utr

The three prime untranslated region ($3'$ UTR) is a particular section of messenger RNA (mRNA). It follows the coding region. An mRNA molecule codes for a protein through translation. The mRNA also contains regions that are not translated. In eukaryotes these regions are the cap, $5'$ untranslated region, $3'$ untranslated region, and polyA tail.

## 1.3 Promoter elements

### 1.3.1 Core promoter

The minimal portion of the promoter required to properly initiate transcription.

- Transcription Start Site (TSS).
- A set of binding sites for RNA polymerase to bind.
- General transcription factor binding sites.

### 1.3.2 Proximal promoter

The proximal sequence upstream of the gene that tends to contain primary regulatory elements.

- Specific transcription factor binding sites

### 1.3.3 Identification of promoter location

As promoters are typically immediately adjacent to the gene in question, positions in the promoter are designated relative to the transcription start site (TSS). Transcription of RNA for a particular gene (i.e, positions upstream are negative numbers counting back from, for example -100 is a position 100 bp upstream) starts from TSS.

### 1.3.4 Promoter Recognition

#### TATA Binding protein

TATA binding protein plays a role in promoter recognition. TATA binding promoter (TBP)

was first isolated and purified from humans in 1996. It is composed of single polypeptide chain and capable of binding specifically to that TATA box of many promoters. TBP is saddle shaped protein with DNA binding fold. The concave DNA binding surface mediates specific and non-specific contacts with the DNA through an anti parallel beta sheet. TBP is functional in mutants lacking the N-terminus. The c-terminus contains two homologous repeats of 88 amino acids that is TBP originally evolved from true dimer of identical chains. The identical domains have asymmetric surface patterns of AA side chains. TBP can thus bind DNA other proteins.

### 1.3.5 Eukaryotic promoters

Eukaryotic promoters are extremely diverse and are difficult to characterize. They typically lie upstream of the gene and can have regulatory elements several kilobases away from the transcription start site. In eukaryotes, the transcription complex can cause the DNA to bend back on itself, which allows for placement of regulatory sequences far from the actual site of transcription. Many eukaryotic promoters, between 10 and 20% of all genes, contain a TATA box (sequence TATAAA), which in turn binds a TATA binding protein which assists in the formation of the RNA polymerase transcription complex. The TATA box typically lies very close to the transcription start site (often within 50 bases).

## 1.4 Problem Statement

Identification of promoters in human genome.

## 1.5 Motivation

- Promoter prediction is an important and complex problem. This complexity arises from the uncertainty in the position of the occurrence of the promoter in the genome.
- Pattern recognition algorithms typically require features that could captures complexity.
- DNA also provides code for proteins and targets for activators, enhancers, repressors, transcription binding factors etc.

- In earlier work most of the times only small portion of the genome is used to evaluate a promoter prediction programs (PPP), which is not a realistic setting for whole genome promoter recognition/prediction.

- Promoter prediction programs (ppp) are computational models which aim at identifying the promoter regions in a genome.

## 1.6 Our Approach - Neural Network Method

We would like to find out the efficacy of these features in identifying the promoters in Homo Sapiens. N-gram features of promoters are given as input to the neural network classifier. Here prediction of promoter is modeled as a binary classification problem. Supervised learning is used for the classification of promoters. Here we have used the DBTSS data as positive data set and the EID, $3'$UTR data negative data set.

## 1.7 Organization of thesis

The current chapter describes the promoter identifies the problem and briefly explains approach being used and introduction to problem. Chapter 2 introduces the related work, existing methods and discuss the material that is a prerequisite to understand our approach clearly and completely. Chapter 3 refers to the data collection, feature extraction, (n-gram) and pattern file generation. Chapter 4 refers to classification of the promoter and non-promoter sequences and Also explains the SNNS simulation. Chapter 5 gives the classification of results promoter recognition, discussion of results and conclusions of this project. The future work needed to be carried out for further research.

# Chapter 2

# Literature Survey

Promoter recognition is a real problem that computationally identifies the transcription start site (TSS) or the $5'$ end of the gene without time-consuming and expensive experimental methods that align ESTs, cDNAs or mRNA against to the entire genome. Promoter prediction is an important and complex problem. This complexity arises from the uncertainty in the position of the occurrence of the promoter in the genome. Pattern recognition algorithm typically requires features that could capture this complexity. In earlier work in most of cases only a small portion of the genome is used to evaluate the promoter prediction program (PPP), which is not realistic setting for whole genome promoter prediction. Promoter prediction programs(PPP) aim to identify promoter regions in a genome using computational models.

## 2.1 Promoter prediction programs

Most of the promoter prediction programs try to predict the exact location of the promoter region of the known protein-coding genes, while some focus on finding the transcription start site (TSS). Some research has show that there is often no single TSS, but rather a whole transcription start region(TSR) containing multiple TSSs that are used with different frequencies. Generally two main approaches are used in promoter prediction [4].
1. First approach assigns scores to all single nucleotides to Identify TSS.
2. Second approach identifies a promoter region without providing scores for all nucleotides.
In this article analyzes the performance of 17 programs on two tasks:

- Genome wide identification of the start of genes.
- Genome wide identification of TSRs.

## 2.2 Existing methods and results in literature

- precision = $\frac{TP}{(TP+FP)}$
- recall = $\frac{TP}{(TP+TN)}$

| Name | Precision | recall |
|---|---|---|
| DragonGSF | $37-48$ | $51-70$ |
| DragonPF | $61-65$ | $62-64$ |
| FristEF | $79-81$ | $35-40$ |
| Eponine | $\approx 40$ | $\approx 67$ |
| NNPP2.2 | $69-93$ | $2.0-4.5$ |
| McPromoter2.0 | $26-57$ | $70-87$ |
| Promoter2.0 | $44-57$ | $\approx 4.5$ |
| proSom | $0.38$ | $0.66$ |

Table 2.1: Results of all protocols on all PPPs.

The above protocols gives the results of precision and recall values. DragonGSF predicts the value of precision is between 37-48% and recall is 51-70%, DragonPF predicts the value of precision is between 61-65% and recall 62-64%, FristEF predicts the value of precision is 79-81% and recall 35-40%, Eponine predicts the value of precision is $\approx 40$ and recall $\approx 67\%$, NNPP2.2 predicts the value of precision is 69-93% and recall 2.0-4.5%, McPromoter2.0 predicts the value of precision is 26-57% and recall $\approx 4.5$, proSom predicts the value of precision is 0.38% and recall 0.66% [5].

- **Promoter prediction programs,features and classifications**

We select the 13 representative PPPs that can analyze large genome sequences and report strand-specific TSS predictions(refer table 2.2). ARTS, Eponine used SVM as the part of their deign, EP3, Promoterscan, Wu-method used Position weight matrix (PWM). CpGcluster used distance based algorithm. CpGProD used Linear discriminating analysis (LDA). DragonGSF, DragonPF, McPromoter used neural networks.NNPP2.2 has used Time Delay neural network.Promoter Explorer has used AbaBoost algorithm. proSOM has used SOM. These programs have used as features various aspects of promoter and other regions of the DNA [6].

| Name | Classifier | Features |
|---|---|---|
| ARTS | SVM,Kernels | CpGIsland,Specific transcription factor binding site(TFBS) |
| CpGcluster | Distance based algo. | CpGIsland,TSS,CpGDenucleotide |
| CpGProD | LDA | CpGIsland,AT/GC content |
| DragonGSF | neural network | CpGIsland,TSS,DPF promoter |
| EP3 | PWM | TATA Box,TSS,Promoter,CpGIsland,INR |
| Eponine | SVM | TATA Box,GC Box,TSS |
| DragonPF | NN | Promoter,Exon,Intron,TSS |
| McPromoter | NN,Interpolated Markov Models | TATA Box,CITA Box,GC Box, nucleosome position |
| NNPP2.2 | Time Delay NN | TATA Box ,INR |
| PromoterExplorer | AbaBoost algorithm | CpGIsland,DNA sequence |
| Promoterscan | PWM(position weight matrix) | TATA Box,Transcription factor binding site |
| proSOM | SOM | TATA boX, |
| Wu-method | PWM | TATA Box,CAAT BoxEnd TranScription site(INR),CpGlands |

Table 2.2: PPV, feature and classifier

## 2.3 Promoter recognition system

There are three classification methods for human promoter recognition system [12].

- Discriminative model that finds the optimal thresholds or classification boundaries in the signal, context and structure features space. Typical methods include artificial neural networks (ANNs), discriminant functions and support vector machines (SVMs).
- Generative model that describes the generative process of signal, context and structure observations. Position weight matrix (PWM), nearest neighborhood and hidden Markov models (HMMs) belong to generative models.
- Ensemble that combines multiple classifier for multiple features in order to achieve a consensus and robust recognition results.

## 2.4 Promoter recognition in a genome sequence

The main aim of the promoter recognition is to locate the promoter region in the genome sequence. In this case we propose a scheme for locating promoters in a given DNA sequence segment of human genome of length N in a particular direction. Locating TSS in the promoter region is being taken here. Consider moving window of length 300bp extracting from the DNA sequence. These are represented as the $(v_1^n, v_2^n ... v_i^2 ..)$ feature vectors which are used by the neural network classifier. Each of the segment gets classified as

Table 2.3: Results of three promoterInspector classifiers

| **Non-promoter set** | **Crossvalidation** | | **Evalution set** | |
|---|---|---|---|---|
| | Promoter | Non-promoter | Promoter | Non-promoter |
| Exon | 82.6 | 70.6 | 80.3 | 75.1 |
| Intron | 57.6 | 80.8 | 60.3 | 81.2 |
| $3'$Utr | 65.5 | 78.6 | 63.5 | 77.6 |

Promoter (P) or non-promoter (NP). If a segment m-(m+299) is classified as a promoter, then the nucleotide m s annotated as P and if it is classified as non-promoter then m is annotated as NP. This process of annotation is continued for the entire sequence to get a sequence of Ps and NPs. We proposed that if a contiguous segment of length more than a certain threshold has all Ps then we annotate that region as promoter region otherwise non-promoter regions[11].

PromoterInspector is a program that predicts eukaryotic pol II promoter regions with high specificity in mammalian genomic sequences. The program PromoterInspector focuses on the genomic context of promoters rather than their exact location. Promoter Inspector is based (refer table 2.3) on three classifiers, which specialize in to differentiating between promoter region and a subset of non-promoter sequences(intron, exon and $3'$utr). In contrast to this, PromnFD and PromFind use only one classifier, i.e the features are extracted from one promoter set and one set of various non-promoter sequences. To compare the two approaches, two versions of PromoterInspector are built. version v1 was based on one st of mixed non-promoter sequences, while version v2 was built on the basis of exon, intron and $3'$utr. Both versions of promoterInspector were applied to exon, intron, $3'$utr and promoter evaluation sets[10].

The identification of promoters and first exons has been one of the most difficult problems in gene-finding. The FirstEF [7] program identifies a promoter region and first exons in the human genome, which may be also be useful for the annotation of other mammalian genomes. FirstEF consists of different discriminant functions structured as a decision tree. The probabilistic models are designed to find potential first splice donor sites and CpG-related and non-CpG-related promoter regions based on discriminant analysis. For every potential first splice-donor site and upstream promoter region, FirstEF decides whether the intermediate region could be a potential first exon based on a set of quadratic discriminant

functions. Training and testing using different discriminant functions, the first exons and promoter regions from the first-exon database are used. In this paper accuracy is tested in two ways. First, a systematic cross-validation analysis is performed using the data in the first-exon database; second, the program on the complete sequences of human. By using different models to predict CpG-related and non-CpG-related first exons, cross-validation results predicted that the program could predict 86% of the first exons with 17%.

Table 2.4: Accuracy of FirstEF based on cross-validation

| Exon type | Sensitivity | Specificity | correlation coefficient |
|---|---|---|---|
| CpG-related | 0.92 | 0.97 | 0.94 |
| not CpG-related | 0.74 | 0.60 | 0.65 |
| all exons | 0.86 | 0.83 | 0.83 |

A promoter recognition method named PCA-HPR is used to locate eukaryotic promoter regions and predict transcription start sites (TSSs). Here the autors have computed codon (3-mer) and pentamer (5-mer) frequencies and created codon and pentamer frequency feature matrices to extract informative and discriminative features for effective classification. Principal component analysis (PCA) is applied to the feature matrices and a subset of principal components (PCs) are selected for classification. They used three neural network classifiers to distinguish promoters versus exons, promoters versus introns, and promoters versus 3' un-translated region (3'UTR). These are compared with three well-known existing promoter prediction systems such as DragonGSF, Eponine and FirstEF. Validation shows that PCA-HPR achieves the best performance with three test sets for all the four predictive systems.

Promoter prediction systems use two type of features for classification namely, context features like n-mers, and signal features such as TATA box, CCAAT-box and CpG islands. Among the favorable promoter prediction programs, Eponine builds a PWM to detect TATA-box and G+C enrichment regions as promoter-like regions;FirstEF uses CpG-related and non-CpG related first exons as signal features; PromoterInspector uses IUPAC words with wild cards as context features. Good experiment results are achieved by integrating these two types features. DPF applies a separate module on G+C rich and G+C poor regions, and selects 256 pentamers to generate a PWM for prediction. Furthermore, DragonGSF adds the CpG-island feature to DPF. In this paper selected three promoter systems,

DragonGSF, Eponine and FirstEF to compare the performance on 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 test results of Eponine and FirstEF. On 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 in the region are counted as FP [8].

| program | True positive | False positive | Sensitivity(%) | PPV(%) |
|---|---|---|---|---|
| DragonGSF | 9 | 14 | 64.2 | 39.1 |
| FirstEF | 9 | 12 | 64.2 | 42.9 |
| Eponine | 9 | 16 | 64.2 | 36.0 |
| PCA-HPR | 9 | 11 | 64.2 | 45.0 |

Table 2.5: Performance comparison of four prediction systems for four human gene

| program | True positive | False positive | Sensitivity(%) | PPV(%) |
|---|---|---|---|---|
| DragonGSF | 269 | 69 | 68.4 | 79.6 |
| FirstEF | 331 | 501 | 84.2 | 39.8 |
| Eponine | 199 | 79 | 50.6 | 71.6 |
| PCA-HPR | 301 | 65 | 76.6 | 82.2 |

Table 2.6: Performance comparison of four prediction systems for 22 chromosomes

## 2.5 Related work

Earlier n-gram based promoter recognition methods were tried in promoter prediction and its application to whole genome promoter prediction in E.coli and Drosophila[11]. Here we extending the earlier work by extracting n-grams and using them to identify promoters in human genome. Patterns or features that characterize a promoter/non-promoter are needed to be extracted from the given set of promoter and non-promoter sequences. Here promoter recognition is addressed by looking at the global signal characterized by their frequency of occurrence of n-grams in the promoter region. In few papers have made an investigation using nucleotide frequencies as features in promoter recognition [9].

# Chapter 3

# Experimental Methods

## 3.1 Data Collection

### 3.1.1 DBTSS database

In this project we are using three benchmark data sets. These are generated in a collaboration between the Informatics group of the Berkeley Drosophila Genome project at the Lawrence Berkeley National Laboratory (LBNL), the Computational Biology Group at the UC Santa Cruz, the Mathematics Department at Stanford and the Chair for Pattern Recognition at the University of Erlangen, Germany. These database contain three data sets.

1. DBTSS
2. EID
3. 3′UTR

These data sets are collected from website:www.fruitifly.org/data/seq-tool/datasets.com.

## 3.2 Dataset Creation

In this project we are using four data sets.

- DBTSS(23,027).

- EID(16,000).
- $3'$UTR(16,000).
- EPD(1863).

In these data sets DBTSS, EPD are the promoter data sets and Exons, Introns(from EID) and $3'$UTR are non-promoter data sets. From DBTSS we have extracted promoter sequences [-250, +50] bp around the experimental TSS. DBTSS contains 24 chromosomes, each chromosome has 1000 nucleotide sequences. From EID and $3'$UTR we have extracted non-promoter sequences of length 300 bp.

## 3.3 Feature Extraction

- **What is n-Gram?**
  An **n-gram** is a subsequence of length **n** from a given sequence. The subsequence in question can be phonemes, syllables, letters, words or base pairs according to the application. It is DNA sequence segments in this case.
- To extract the global signal for a promoter, frequency of occurrence of **n-grams** is calculated on the DNA alphabet A,T,G,C.
  1. A-Adenine
  2. T-Thymine
  3. G-Guanine
  4. C-Cytosine
- The set of *n-grams* for n=2 is 16 possible pairs such as AA,AG,AC,TA, ... and so on. Similarly for n=3,4,5 are also calculated.
- $V_n^i = \frac{f_i^n}{|L|-(n-1)} 1 \leq i \leq 4^n$, for n = 2,3,4,5.

  $f_i^n$= the frequency of occurrence of the ith n-gram feature for a particular n value.

  |L|= denotes the length of the sequence.

  The features values $V_n^i$ are normalized frequency counts.

  N-grams for n=2,3,4 and 5 are computed for both promoter and non-promoter data sets.

## 3.4 Neural Network Input File Generation (Stuttgart Neural Network Simulator)

**1. n-gram File Generation**

**2. Pattern File Generation**

Promoter prediction in Homo Sapiens is done using n-gram feature extraction from the sequence where n=2, 3, 4, 5. n-gram features are extracted for each training sequence and these are given to neural network. we use Stuttgart Neural Network Simulator (SNNS) for simulations. Similarly n-grams are computed for test data also.

### 3.4.1 n-gram File Generation(Description with an Example)

The 2-gram method extracts various patterns of two consecutive nucleotide in sequence and counts the number of occurrences of the extracted pairs. The pair frequency count of sequence extracted from positive data set is called positive bi-gram features and pair frequency count of sequence extracted for negative data set is called negative bi-gram features.

For example, for following sequence, the 2-gram features is done in the following way.

CCATCACAATGGAAAGAAGCTTCCCTGTCAAGAGGACTCAGCTACAGAAGGTACCA
AATGTGGTAGGAGGGGCCTGTTAATTAGACCAAGGCAGTCACACATCAGCAGGTAA
AACAGAGACAAGAGGAGGTGTGGCTGGGCTGGGCTGGATCTTGGATGAATCAAGCC
TTCCCATAGGGCAGGATATCCTGTCTAAAACAAGAGCCTTGGTTAAAACCCCTATA
AAAGGTTCTCATCACACTGACCTGGTACTCCTCACACCACTTAACAGCCACTTGTT
TCATCCCACCTGGGCATTAG

21 for AC (indicating AC occurs 21)
27 for AG (indicating AG occurs 27)
21 for CC (indicating CC occurs 21)
18 for TG (indicating TG occurs 18)
15 for AT (indicating AT occurs 15)
18 for TC (indicating TC occurs 18)
17 for GA (indicating GA occurs 17)
27 for AA (indicating AA occurs 27)
13 for TT (indicating TT occurs 13)
22 for CT (indicating CT occurs 22)
27 for GG (indicating GG occurs 27)

15 for TA (indicating TA occurs 15)
13 for GC (indicating GC occurs 13)
31 for CA (indicating CA occurs 31)
14 for GT (indicating GT occurs 14)
For example x4 x5 x1 x4 x2 x5 x1 bi-gram method gives the following results:
1 for x4 x5,
2 for x5 x1,
1 for x1 x4,
1 for x4 x2,
1 for x2 x5

### 3.4.2 2-gram program Generation

PERL program is used to generate the 2-gram file. This program uses FASTA format sequence as input file and generates bi-gram output file. Example is given below for sequences are convert the FASTA format sequence files into the bi-gram output file. The bi-gram output file is used for pattern file generation, which is a input file for the SNNS (Stuttgart Neural Network Simulator).

1)0.07023 0.07023 0.06020 0.05017 0.09030 0.07358 0.00000 0.05017 0.04348 0.10368 0.04682 0.09030 0.06020 0.05686 0.04348 0.09030
2)0.06355 0.11706 0.05351 0.05351 0.06355 0.05686 0.08027 0.03010 0.09030 0.08361 0.04348 0.04348 0.07023 0.04348 0.03679 0.07023
.....
.....
unto 3382 sequences it is Positive sequence patterns represented with output pattern unit ’1’.
.....
From 3383 sequences it is negative sequence patterns represented with output pattern unit ’0’.
3383)0.06689 0.11371 0.06355 0.05686 0.09699 0.05686 0.01003 0.04013 0.05686 0.09699 0.03010 0.07023 0.04013 0.05351 0.07692 0.07023
.....
.....
6754)0.04348 0.10702 0.07023 0.05351 0.06020 0.06689 0.01672 0.05686 0.02007 0.05017

0.07358 0.06355 0.07023 0.05686 0.13712 0.05351

Above patterns are for 2-grams for one test set and training set similarly generate the n-gram files for all n=3,4,5 values.

### 3.4.3 Pattern File Generation

- Generate the training set and test pattern files.
- The training set will be a mixture of both promoter and non-promoter data sets.
- Similarly the test set is also consisting of both promoter and non-promoter data and is used to evaluate the performance of the classifier.
- we have generated pattern files for all DBTSS database, the EID database and EPD database.

SNNS pattern definition file V3.2
generated at mon Apr 04 3:33:30 2010
No. of patterns : 6754
No. of input units : 16
No. of output units : 1
# Input pattern : 1
0.07023 0.07023 0.06020 0.05017 0.09030 0.07358 0.00000 0.05017 0.04348 0.10368 0.04682 0.09030 0.06020 0.05686 0.04348 0.09030
# Output pattern : 1
1
# Input pattern : 2
0.06355 0.11706 0.05351 0.05351 0.06355 0.05686 0.08027 0.03010 0.09030 0.08361 0.04348 0.04348 0.07023 0.04348 0.03679 0.07023
# Output pattern : 2
1
.....
.....
Upto 3381 sequences it is positive sequences patterns represented with output unit ’1’
# Input pattern : 3382
0.05351 0.07358 0.08696 0.08027 0.08696 0.09365 0.00334 0.04682 0.05686 0.08027 0.02676 0.05351 0.06689 0.06020 0.11371 0.01672

# Output pattern : 3382

0

From 3382 sequences onwards it is negative sequence patterns represented with output pattern unit ’0’.

.....

.....

# Input pattern : 6754

0.03679 0.04348 0.06689 0.01003 0.07358 0.01338 0.01672 0.01003 0.05017 0.06689 0.06355 0.18395 0.01003 0.15385 0.02007 0.18060

# Output pattern : 6754

0

Above patterns are for 2-grams for one test set and training set and as like generate the training set and test pattern files for all n=3,4,5.

# Chapter 4

# Classification

## 4.1 Supervised Learning

Classification methods can be either supervised or unsupervised. In supervised learning a set of example pairs(x,d), where x belongs to the example set, d is the category of x are given. A set of training examples are used to train the classifier. Once training is complete, same configuration values can be used to test the test data set and estimate the classification accuracy.

## 4.2 Feed Forward Neural Network

An artificial neural network (ANN), often just called a ”neural network” (NN), is a mathematical model or computational model based on biological neural networks. It consists of an interconnected group of artificial neurons and processes information using a connection list approach to computation. Most common neural network is a single layer Perceptron In this network, the information moves in only one direction, forward, from the input nodes, through the hidden nodes (if any) and to the output nodes. There are no cycles or loops in the network. For a given artificial neuron, let there be $m + 1$ inputs with signals $a_1$ through $a_n$ and weights $w_1$ through $w_n$. Usually, the $a_1$ input is assigned the value + 1, which makes it a bias input with $w_{j1} = b_j$. This leaves only n actual inputs to the neuron: from $a_1$ to $a_n$. Output of j-th neuron is:$S_j = (\sum_{i=0}^{m} W_{ji} a_i)$.The output is analogous to the axon of a biological neuron, and its value propagates to input of the next layer, through a synapse. It may also exit the system possibly as part of an output vector.

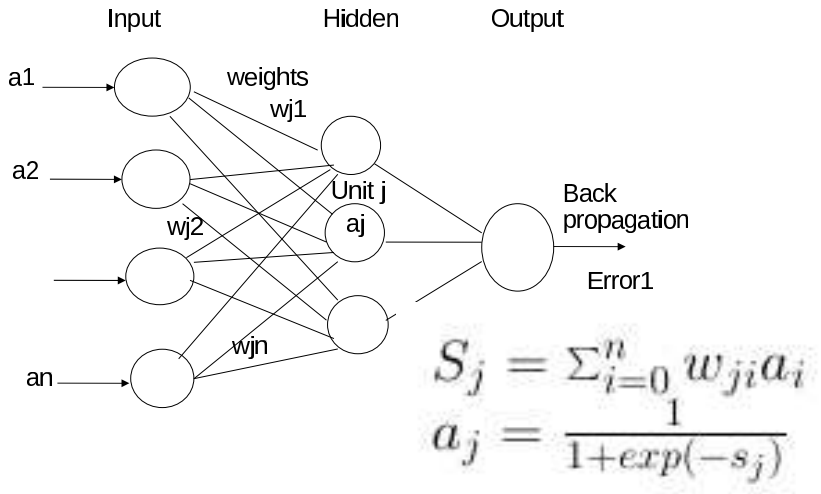


Figure 4.1: Feed Forward Neural Network

## 4.3 SNNS Simulation Flow Chart

A neural network consists of units and directed, weighted links (connections) between them. In analogy to activation passing in biological neurons, each unit receives a net input that is computed from the weighted outputs of prior units with connections leading to this unit. The actual information processing within the units is modeled in the SNNS simulator with the activation function and the output function. The activation function first computes the net input of the unit from the weighted output values of prior units. It then computes the new activation from this net input (and possibly its previous activation). The output function takes this result to to generate the output of the unit.

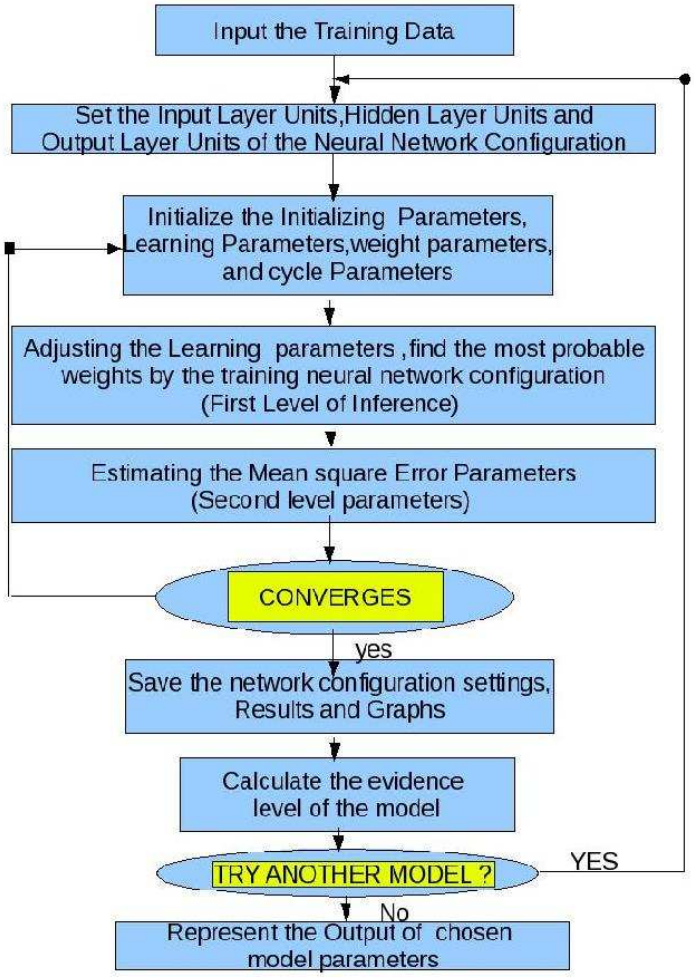


Figure 4.2: neural network learning flow chart

## 4.4 SNNS Simulation

**Loading the pattern files**

1. Pattern file for Training
2. Pattern file for Testing
3. Pattern file for validation
4. Gnerate neural network using BIGNET(Details in annexture-2)
5. Trainning of the neural network is started.

## 4.5 Classification Performance

In a binary classification problem the training set will be a mixture of both positive and negative data sets. Similarly the test set also consisting of both positive and negative data is used to evaluate the performance of classifier. A neural network classifier is trained using n-grams of training set as input feature vectors and then the test set is evaluated using by same network. The below figures are depict the average separation between the positive and negative for n=2, 3, 4,5 respectively. It can be observed that the plots depict the separability of promoter and non-promoter data sets in different feature spaces.

A feed forward neural network with three layers is used for promoter classification. The nodes in the input layer are 16, 64, 256, 1024 features for n=2, 3, 4, 5 respectively. The experimentation is done with different number of hidden nodes that give an optimal classification performance. The output layer has one node to give a binary decision as to whether the given input sequence is a promoter or non-promoter. 5-fold cross validation is used to investigate the effect of various n-gram on promoter classification by neural network. Average performance over these folds is being reported. These simulation are using Stuttgart Neural Network Simulator(SNNS). The classification results are evaluated using performance measures such as Precision, Specification, Sensitivity 5.3, 5.4 given these results. Using these, we would like to find out the efficacy of these features in identifying promoter in human genome [2].

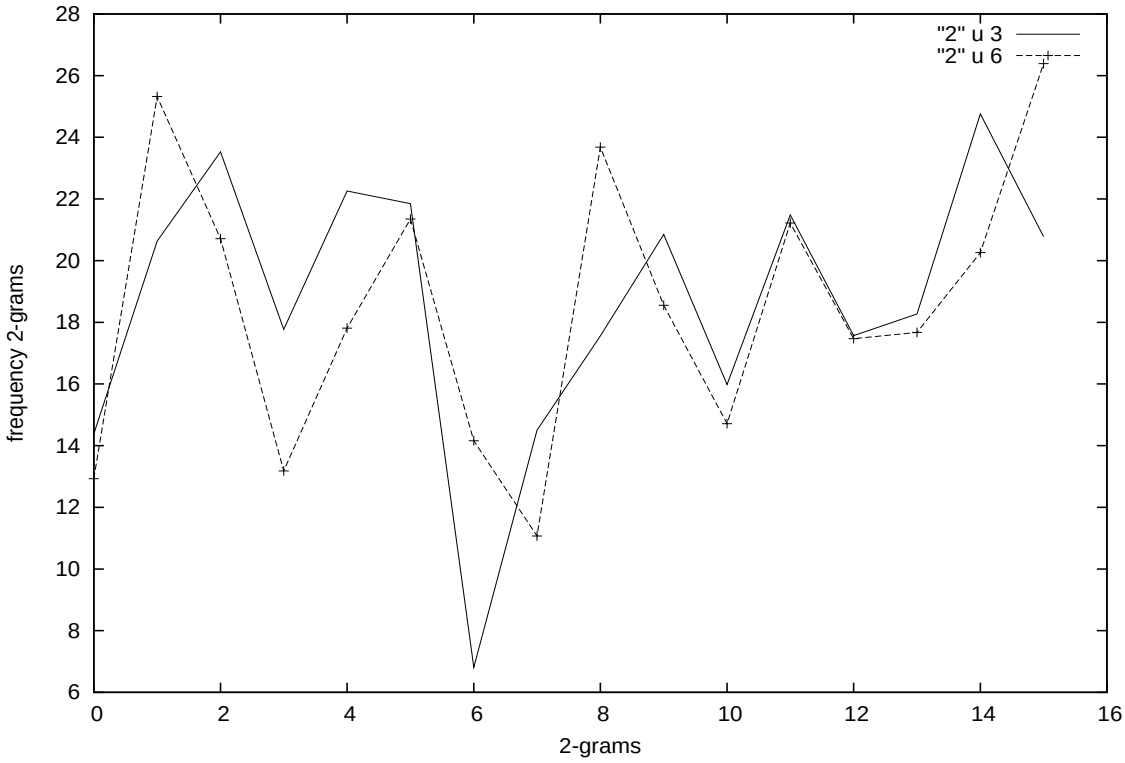


Figure 4.3: Average separation between promoter and non promoter for n=2. Here 0..16 represent the AA, AT, AG, AC..etc

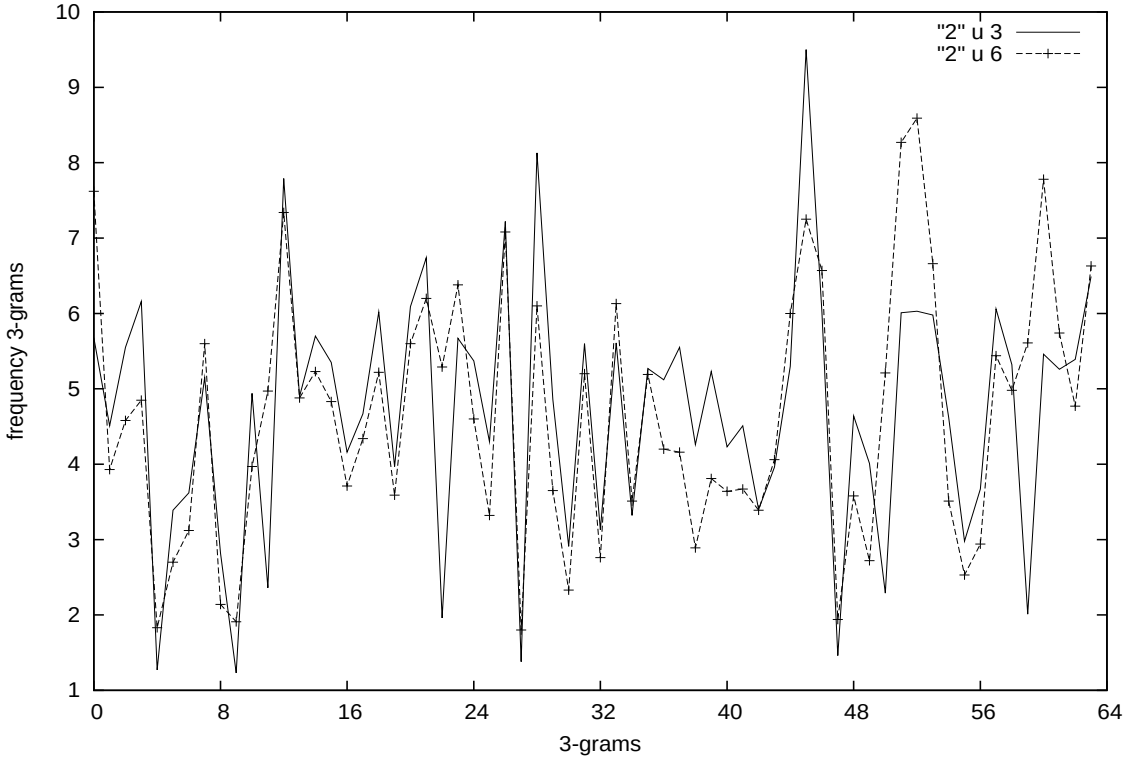


Figure 4.4: Average separation between promoter and non promoter for n=3. Here 0..64 represent the AAA, AAT, AAG, AAC..etc

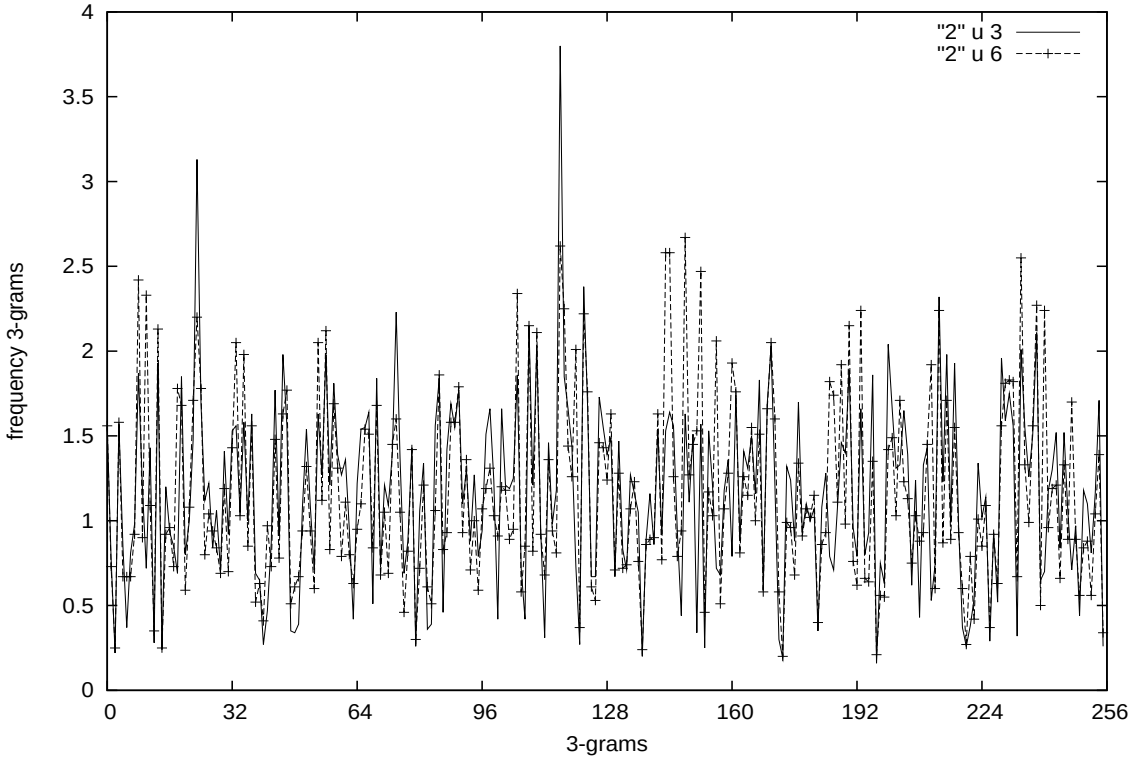


Figure 4.5: Average separation between promoter and non promoter for n=4. Here 0..256 represent the AAAA, AAAT, AAAG, AAAC..etc

# Chapter 5

# Results, Discussion and Conclusions

## 5.1 Confusion Matrix

In Predictive Analysis a table of confusion (also known as a confusion matrix) is useful in estimating the prediction accuracies. Confusion matrix has two rows and two columns that report the number of true Negatives (TN), false Positives (FP), false Negatives (FN), and true Positives (TP). In the field of artificial intelligence, a confusion matrix is a visualization tool typically used in supervised learning (in unsupervised learning it is typically called a matching matrix). Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class. One benefit of a confusion matrix is that it is easy to see if the system is confusing two classes (i.e. commonly mislabeling one as another). When a data set is unbalanced (when the number of samples in different classes vary greatly) the error rate of a classifier is not representative of the true performance of the classifier.

**Notations:** PPV=Positive Predicted Values, NPV=Negative Predicted Values,
Sensitivity=True Positive Rate, Specificity=False Positive Rate
TP = True positives, FP = False positives,
FN = False negatives, TN = True negatives

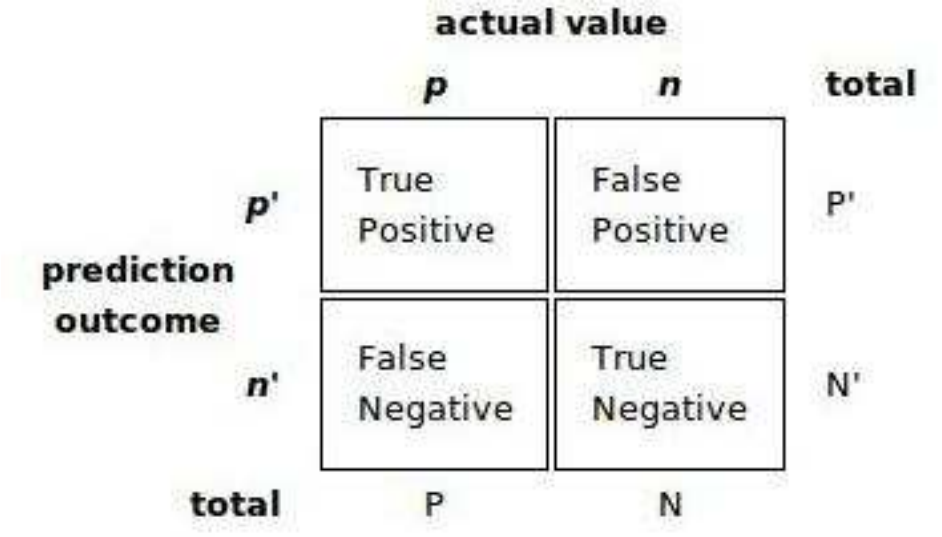


Figure 5.1: Confusion matrix

## 5.2 Classification measures

### 5.2.1 Correct Classification

This specifies the correct classification number of the features.

**Correct Classification** =$(TP + TN)$

### 5.2.2 Wrong Classification

This specifies the wrong classification number of the features.

**mis Classification** =$(FP + FN)$

### 5.2.3 Precision

A precision of 100% means the classifier classifies the sequences 100% accurately. This parameter gives overall accuracy of prediction. Classifier predicts positive and negative sequences, positive sequence prediction is measured with the parameter sensitivity and negative predicted sequences is measured with the parameter specificity.

**Precision** = $\frac{(TP+TN)}{(TotalNo.ofpatterns)}$

### 5.2.4 Sensitivity

Sensitivity, or recall rate, is a statistical measure of how well a binary classification test correctly identifies a condition, whether the human has contain the promoter or not For example, In this one we have to create the pattern file with 27016 patterns for classification of sequences. Out of 27016 patterns with few positive and few negative sequences, the classification output for example is mentioned in the above table can be calculated as follows:

**Sensitivity =true positive rate =TP rate=** $\frac{TP}{(TP+FN)}$

A sensitivity of 100% means that the test recognizes all positive classes as positive. Sensitivity alone does not tell us how well the test predicts other classes (that is, about the negative cases). In the binary classification, this is the corresponding specificity test, or equivalently, the sensitivity for the other classes. Sensitivity is not the same as the positive predictive value (ratio of true positives to combined true and false positives), which is as much a statement about the proportion of actual positives in the population being tested as in the test.

### 5.2.5 Specificity

A specificity of 100% means that the test recognizes all negative cases as negative. The maximum is trivially achieved by a test that claims every class is negative regardless of the true condition. Therefore, the specificity alone does not tell us how well the test recognizes positive sequences. We also need to know the sensitivity of the test to the class, or equivalently, the specificity's to the other classes. A test with a high specificity has a low Type I error rate. Specificity is sometimes confused with the precision or the positive predictive value (PPV), both of which refer to the fraction of returned positives that are true positives. The distinction is critical when the classes of different size. A test with very high specificity can have very low precision if there are far more true negatives than true positives, and vice versa.

**Specificity = false positive rate = FP rate =** $\frac{TN}{(TN+FP)}$

### 5.2.6 Mean squared error

The mean squared error (MSE) of an estimated parameter $\hat{\theta}$ is

$$\text{MSE}(\hat{\theta}) = E[\hat{\theta} - \theta]$$

The MSE of an estimator $\hat{\theta}$ with respect to the estimated parameter $\theta$ is defined as

$$\text{MSE}(\hat{\theta}) = var(\hat{\theta}) + E[\hat{\theta} - \theta]^2$$

The MSE is equal to the sum of the variance and the squared bias of the estimator

$$\text{MSE}(\hat{\theta}) = var(\hat{\theta}) + Bias^2(beta\theta)$$

Since MSE is an expectation, it is a scalar, and not a random variable. It may be a function of the unknown parameter $\theta$, but it does not depend on any random quantities. However, when MSE is computed for a particular estimator of $\theta$ the true value of which is not known, it will be subject to estimation error. In a Bayesian sense, this means that there are cases in which it may be treated as a random variable.

### 5.2.7 SNNS Result File Analysis

**Algorithm** **BackPropagationMomentum**

**Learning Parameters**

| alpha(neta) | Mu | C | Dmax |
|---|---|---|---|
| Learn rate | Momentum term | Flat spot elimination | Max ignored errors |
| 0-1 | 0-0.4 | 0-0.05 | 0-0.25 |

MSE= mean square error

INIT= initialization values

## 5.3 Neural network Parameters

### 5.3.1 DBTSS

| S.no | Features | Input Layer | Hidden Layer | Output Layer |
|---|---|---|---|---|
| 1 | 2-gram | 16 | 8 | 1 |
| 2 | 3-gram | 64 | 24 | 1 |
| 3 | 4-gram | 256 | 64 | 1 |
| 4 | 5-gram | 1024 | 256 | 1 |

Table 5.1: Network Configuration of SNNS. Which shows i) Number of Input Layer Nodes ii) Number of Hidden Layer Nodes and iii) Number of Output Layer Nodes for DBTSS datase.

In this project we generate parameter for input layer,hidden layer and output layer. Then connecting full-connection between the input, hidden and output layer and connect the configuration set for input layer, hidden layer and output layer for DBTSS.

## 5.4 Best Performance values

In this project we are using the best network configuration for 5-fold cross validation. The tables 5.3 gives the results for data sets. Classification of both promoter (DBTSS data sets) and non-promoter is best for n=4. precision of 72.4%, Specificity of 86.5%, Sensitivity of 64.2% and PPV of 89.2% are obtained for this set. Precision of 72.5%, Specificity of 81.8%, Sensitivity of 67.4% and PPV of 87.2%. For promoter set containing of DBTSS and EPD data sets. The result show that for DBTSS n=4 gram features give the better performance than other n-grams and for EPD n=3 gram features give the better performance than other n-gram features.

| S.No | Features | Precision | Specificity | Sensitivity | PPV |
|---|---|---|---|---|---|
| 1 | N=2 gram | 0.68463 | 0.84048 | 0.57364 | 0.83513 |
| 2 | N=3 gram | 0.70851 | 0.81923 | 0.63906 | 0.84882 |
| **3** | **N=4 gram** | **0.72419** | **0.86507** | **0.64323** | **0.89239** |
| 4 | N=5 gram | 0.76556 | 0.69432 | 0.80824 | 0.81527 |

Table 5.2: Best Accuracy Values for DBTSS

### 5.4.1 DBTSS

## 5.5 Discussion

In proposed approach human promoters are classified using a neural network classifier. Since the optimal features are not known we started classification model with with minimum number of features n=2 and incrementally increased to maximum number of features. The features are given as input to a single layer feed forward neural network. Back propagation momentum learning algorithm using is used for training of the neural network. Algorithm parameters are : i) learn rate ii) momentum term iii) Flat spot elimination and iv) maximum ignored errors.

Different number of feature values are used to arrive at the best performance. Test results are measured using measures such as precision, specificity and sensitivity and PPV. Maximum accuracy achieved using SNNS is 89.2% in the case of DBTSS data set. The best accuracy values of the results are tabulated in the table 5.3 in chapter5.

In this experiment, we focus on extracting the statistical features. There is evidence of a statistical preference in terms of codon usage patterns in protein coding regions. The majority of promoter prediction methods available now directly extract a limited number of context features from sequences. Here we are not doing any feature selection and using the entire set of n-grams.

In this project Classification of both promoter (DBTSS data set) and non-promoter is best for n=4. We obtained a precision of 72.4%, specificity of 86.5%, sensitivity of 64.2% and positive predctive value of 89.2% for this set. The result shows that for DBTSS n=4 gram features give the better performance than other n-grams. The results here consolidate the results obtained for Drosophila Melanogaster in the work done by Sobha et al. They

obtained best performance results for n=4. Does this then make 4-grams as a special parameter for the eukaryotes is needed to be further investigated?

## 5.6 Conclusions

A study of the n-gram (n=2, 3, 4, 5) as features for a binary neural network classifier is done. In human genome 4-gram features give the an optimal performance with neural network classifier. The results show that the classification result 4-gram are better in identification of the promoter than the other n-grams. Human promoter classification gives the better accuracy results of 89.2%.

## 5.7 Future work

As the future work two probelms can be addressed:

1. Application of n=4 grams for promoter prediction in whole genome.
2. Promoter identification and prediction of other higher eukaryote genomes can be done. We can further focus on the association between predictions and TSS or gene starts.

# Bibliography


[1] Benchmark datasets. www.fruitifly.org/data/seq-tool/datasets.com.

[2] Feed forward neural network. http://reference.wolfram.com/applications/neuralnetworkstheory.

[3] Stuttgart neural network simulator. http://www-ra.informatik.uni-tuebingen.de/SNNS/.

[4] Thomas Abeel, Yves van de peer, and Yvan Saeys. Towards a gold standard for promoter prediction evalution. *Bioinformatics*, 25(53):i313–i320, July 2009.

[5] Bajic. V. B, Brent. M. R, Brown. R. H, Frankish. A, Harrow. J, Ohler. U, Solovyev. V. V, and Tan. S. L. Performance assessment of promoter predictions on encode regions in the egasp experiment. *Genome Biology*, 1(3):1–13, Aug 2006.

[6] Vladimir B Bajic, Sin Lam Tan andYutaka Suzuki, and Sumio Sugano. Promoter prediction analysis on the whole human genome. *nature biotechnology*, 22(11):1467 – 1473, November 2004.

[7] Ramana V. Davuluri, Ivo Grosse, and Michael Q. Zhang. Computational identification of promoters and first exons in the human genome. *Nature Publishing Group*, November 2001.

[8] Xiaomeng Li, Jia Zeng, and Hong Yan. A principle component analysis model for human promoter recognition. *Bio information*, 2(9):373–378, june 2008.

[9] Shobha Rani.T, Durga bahvani, and Raju.S.Bapi. Analysis of e.coli promoter recognition problem in dinucleotide feature space. *Bioinformatics*, 23(5):582–588, January 2009.

[10] Matthias Scherf, Andreas Klingenhoff, and Thomas Werner. Highly specific localization of promoter regions in large genomic sequences by promoterinspector: a novel context analysis approach. *Molecular Biology*, 297(3):599–606, March 2000.

[11] Sobha Rani T and Raju S.Bapi. Analysis of n-gram based promoter recognition methods and application to whole genome promoter prediction. *In Silico Biology*, 9(1-2):S1–S16, March 2009.

[12] Jia Zeng, shanfeng Zhu, and Hong Yan. Towards accurate human promoter recognition:a review of currently used sequence features and classification metthods. *Briefings in Bioinformatics*, 10(05):498–508, 2009.

# Appendix A

# Annexture-1

## A.1 Data sets

In this project we are using DBTSS, EID, $3'$Utr and EPD data sets. DBTSS and the EPD are the promoter datasets and EID, $3'$Utr are the non-promoter data sets.

1. DBTSS.
   chr1
   CCATCACAATGGAAAGAAGCTTCCCTGTCAAGAGGACTCAGCTACAGAAGGTAC
   CAAATGTGGTAGGAGGGGCCTGTTAATTAGACCAAGGCAGTCACACATCAGCAG
   GTAAAACAGAGACAAGAGGAGGTGTGGCTGGGCTGGGCTGGATCTTGGATGAAT
   CAAGCCTTCCCATAGGGCAGGATATCCTGTCTAAAACAAGAGCCTTGGTTAAAA
   CCCCTATAAAAGGTTCTCATCACACTGACCTGGTACTCCTCACACCACTTAACA
   GCCACTTGTTTCATCCCACCTGGGCATTAG
   .....
   chr2
   AGATGCAGGAGGAACCCCTCTCAAAGAGCGCCACGGAGAAAACTTCAGAGTACGG
   GCGCGATGATCCAGCACAACGCCAGGACCTGCAGCCCAGCCCACCTCTCCCCGGG
   CTGCTGTGGCGCAGGCGCAGTGGCGCCGCGCTCCGGCCCCAGCGCGCATGCTCTG
   CCCGGCTGCGGCGCTTCGGGCAggcggcggcggcggcggcggcggcAGAGGGAGTTTCC
   GCTTTGTACTCCACCCCGGTAGCAGCTCCGCGGCAGGGACAGCTTCCTCCG
   GACGCTTGGCGGGCTTCGCTCT
   .....

.....

.....

chrY

CATCCCAGAGACTACCCTTTTCTCCATAGACGTGACCATCAACCAACCAGCG
GTCAGAATCAGTCAGCCTCTGTCATGTTCCTAGGTCCTTGGCGAACTGGCTG
GGCGGGGTCCCAGCAGCCTAGGAGTACAGTGGAGCAATGCCTGACGTAAGTC
AACAAAGATCACGTGAGACGAATCAGTCGCCTAGATTGGCTACAACTAAGTG
GTTGGGAGCGGGGAGGTCGCGGCGGCTGCGTGGGGTTCGCCCGTGACACAAT
TACAACTTTGTGCTGGTGCTGGCAAAGTTTGTGATTTTAA

.....

In DBTSS there are 24 chromosomes. Each chromosome contain 1000 nucleotide sequences of the length 300 bp.

2. EID

EID consist of exon and intron data sets. These are non-promoter data sets.

**1.Exon**

chr1

ATGCGTAGACACACACATCCTTACTCTGCGCGCATCCCTGGCCTGGTGGACG
GAAGATCGAGCGCTCTGGGTGGACTTACGGCCACAGGACGGGGGCAGAGTCG
GCAGGGAGGCCCCTCCGAGGCCCAGTGGGCCCTGCGCTGGCCCCGGCCGCAG
ACGCCGCCGCGGTGGGGTGGGCGGATTCCCCCGGAGCAGGCCCAGGCCCCTG
CTCCTGAGCTCTCCTGCAGCGCCGCCTGCTGGCCACAGAGAGCCCACGTGCG
CCGGCCGCCAGGCCTGGGCATCTCCCCTCCTGCAGCGCCG

.....

.....

**2.Intron**

gcatgaaaggctggccaggttggctaaatgggaccacagcagaagcatgagccccagaatgtgcacgaaggaagagagagc
cgggggaggtggcgggctgggtgtgcagagtgggcctgagctccggcctcctccctggacgccctcccgtggccgcagcag
ggtcatgaggggagctgacttctgattagggcatttcatccttctctgaaatgcagctgagaactggtcagcctcactcccttgctg
agaccaatagcaacccctgatgatctcgccacaggtccagcaggtgcccca

.....

.....

3. 3′Utr

AGCATTTTCTTGATGACCCTGCACAATACTGTGAGGAAAATTGACTGCAGAA

GCCTACTTCACACCGCCTTCTCTTATTTTCTGCCCATTGATAAACCTCTCCC
CATATTTTGCAAAGAGGAAATTCACAGCAAAAGTCCACATTATGTCAGCTTT
CTCATATTGAGAGCTCTGCTATGCCACTGTTGAATTTTTCCCAAGATTCCTG
TCCCTAGCCCTCACTTCAAACTCTGCTTCCTTGGACAGATTTGGCAATAGCT
TTGTAAGTGATGTGGACATAATTGCCTACAATAATGAAAA
.....

4. EPD
CTGTCGGTGACATCACGGATAGGGCGACTTCTATGTAGATGAGGCAGCGCAGG
GGCTGCTGCTTCGCCACTGGCTGTTTCACCACGAAGGAGCTCCCGTGCCGTGG
GAGCGGGTTCAGGACCGCTGGTCGGACCTGAGGGTCCCAGCTGTGTGTCAGGG
CTAGGAAGGCTCGGGGGTGCGCGGGGCAAGTGACCATGTGTGTAAAGGGTGAG
GTATATGGAGCTGTGACAGGGCAGAAGTGTGTGAAGTCATACTTACCTGGCAG
GGGAGATACCATGATCACGAAGGTGGTTTTCCCAG
.....

# Appendix B

# Annexture-2

## B.1 Generate neural network using BIGNET

**step 1:** Loading the pattern files of

1. Pattern file for Training
2. Pattern file for Testing
3. Pattern file for validation
4. The Graph file will be between these two files and the finally after the simulation stops, the promoter pattern file of validation is used and results are stored through step 8.
   **Note:** After loading the protein pattern file of validation simulations should not be run, this file is only used to save the results on the training done for the neural network with that configuration, by clicking on the File button in SNNS manager panel and clicking on the Res button for storing the results with the file name of your choice.

**step 2:** Generation of Bignet in general type

1. Input Layer
   X-direction : 1
   Y-direction: 16

2. Hidden Layer
   X-direction : 1
   Y-direction : 8

3. Output Layer
   X-direction : 1
   Y-direction : 1
   In this way We calculate the n = 3, 4, 5.

4. Connecting full-connections between the input, hidden and output layer.

5. Creating the Bignet for the configuration set for input, hidden and ouput layer.

**step 4:** Setting the Training functions:

1. learning function: Backpropagation Momentum (Feed Forward Learning)
   Setting the four values for this function
   1. Learn rate ( range : 0 to 1)
   2. Momentum term ( range: 0 to 0.4)
   3. Flat spot elimination ( range: 0 to 0.05)
   4. Max ignored errors ( range: 0 to 0.25)

2. update function: Topological order

3. initialization function: Randomized weights ( range: -0.05 to 0.05)

**step 5 :** control panel settings

1. valid option : 1 (for enable valid graph visibility)
   0 (for disabling valid graph visibility)

2. pattern: 1 ( by default it is 0)

3. cycles: 1 to 50,000

**step 6:** Simulation Graph

1. From SNNS manager Panel by clicking the graph button, Graph of simulation can be seen

2. Graph for Mean Square Error setting (MSE)

3. Grid Layout for pointing out the graph location

## B.2 Backpropagation Learning Algorithm

Initialize the weights in the network (often randomly)
Do
For each example e in the training set
O = neural-net-output(network, e) ; forward pass
T = teacher output for e
Calculate error (T - O) at the output units
Compute $\delta w_i$for all weights from hidden layer to output layer ; backward pass
Compute $\delta w_i$ for all weights from input layer to hidden layer ; backward pass continued
Update the weights in the network
Until all examples classified correctly or stopping criterion satisfied
Return the network

### B.2.1 Backpropagation algorithm summary

1. Present a training sample to the neural network.
2. Compare the network's output to the desired output from that sample. Calculate the error in each output neuron.
3. For each neuron, calculate what the output should have been, and a scaling factor, how much lower or higher the output must be adjusted to match the desired output. This is the local error.
4. Adjust the weights of each neuron to lower the local error.
5. Assign "blame" for the local error to neurons at the previous level, giving greater responsibility to neurons connected by stronger weights.
6. Repeat the steps above on the neurons at the previous level, using each one's "blame" as its error..

**step 7:** Simulation start and end from control panel

1. Clicking on Reset button, the Bignet learning weights are reset to 0
2. Clicking on Init button, the initialized to values set in the control panel
3. Clicking on Shuffle button, shuffles the pattern features to start simulation

4. Clicking on All button, simulation starts and could be seen in the graph and display window

5. Clicking on Stop button, simulation stops

**step 8:** Saving Results

1. Clicking File button, from SNNS manager panel, the file browser is displayed

2. Click Res button, type the result file name and click Save button for saving the file

3. After clicking on Save button Result File Format window opens where we can choose start pattern, end pattern,result file mode, include input patterns and include output patterns.

4. Default Settings for result file storage start pattern will be 1
   end pattern will be number of patterns and pattern file has, say 27016
   include input patterns will be no
   include output patterns will be yes for easier result file analysis.

# Appendix C

# Annexture-3

PERL program is used to generate the pattern files, which is input file for Stuttgart Neural Network Simulator.
for example:
**1. Test file:**
SNNS pattern definition file V3.2
generated at Mon Apr 04 3:33:30 2010
No. of patterns : 6754
No. of input units : 16
No. of output units : 1
# Input pattern : 1
0.07023 0.07023 0.06020 0.05017 0.09030 0.07358 0.00000 0.05017 0.04348 0.10368 0.04682 0.09030 0.06020 0.05686 0.04348 0.09030
# Output pattern : 1
1
# Input pattern : 2
0.06355 0.11706 0.05351 0.05351 0.06355 0.05686 0.08027 0.03010 0.09030 0.08361 0.04348 0.04348 0.07023 0.04348 0.03679 0.07023
# Output pattern : 2
1
.....
.....
Upto 3381 sequences it is positive sequences patterns represented with output unit '1'
# Input pattern : 3382

0.05351 0.07358 0.08696 0.08027 0.08696 0.09365 0.00334 0.04682 0.05686 0.08027 0.02676 0.05351 0.06689 0.06020 0.11371 0.01672

# Output pattern : 3382

0

From 3382 sequences onwards it is negative sequence patterns represented with output pattern unit ’0’.

.....

.....

# Input pattern : 6754

0.03679 0.04348 0.06689 0.01003 0.07358 0.01338 0.01672 0.01003 0.05017 0.06689 0.06355 0.18395 0.01003 0.15385 0.02007 0.18060

# Output pattern : 6754

0

**2.Training file:**

SNNS pattern definition file V3.2

generated at Mon Apr 04 3:33:30 2010

No. of patterns : 27016

No. of input units : 16

No. of output units : 1

# Input pattern : 1

0.04682 0.01003 0.09030 0.10368 0.10368 0.04013 0.00000 0.10368 0.03010 0.04348 0.06020 0.08361 0.01003 0.08696 0.13378 0.05351

# Output pattern : 1

1

# Input pattern : 2

0.05017 0.07358 0.10702 0.03010 0.05017 0.09365 0.01672 0.02676 0.06689 0.06020 0.06020 0.08361 0.05017 0.07692 0.03679 0.11706

# Output pattern : 2

1

.....

.....

Upto 13524 sequences it is positive sequences patterns represented with output unit ’1’

# Input pattern : 13524 0.03344 0.02676 0.08361 0.08361 0.08361 0.07023 0.01003 0.07358 0.03344 0.04682 0.06355 0.05017 0.06020 0.04682 0.18395 0.05017

# Output pattern : 13524

1

.....

.....

# Input pattern : 13525

0.01672 0.09699 0.06355 0.00669 0.02676 0.06020 0.12040 0.00669 0.14047 0.02676 0.05686 0.06020 0.05017 0.05017 0.02341 0.19398

# Output pattern : 13525

0

.....

.....

From 13525 sequences onwards it is negative sequence patterns represented with output pattern unit ’0’.

.....

# Input pattern : 27016

0.04013 0.01672 0.05686 0.07692 0.16388 0.08696 0.00000 0.06689 0.04348 0.06355 0.02676 0.07692 0.06689 0.06689 0.11371 0.03344

# Output pattern : 27016

0

Above patterns are for bi-gram for one test set and training set and as like Generate the training set and test pattern files all n=3,4,5.

when given the above pattern file gives to input file for Stuttgart Neural Network Simulator. Here PERL program is use to calculate the values for Precision, Specificity, Sensitivity, promoter prediction value(PPV).After simulate the above file we have to calculate the Precision, specificity, Sensitivity, PPV values for each n-grams.

SNNS result file V1.4-3D

generated at Tue Mar 2 23:47:21 2010

No. of patterns : 27016

No. of input units : 16

No. of output units : 1

startpattern : 1

endpattern : 27016

teaching output included

#1.1

1 ————>Expected value

0.45925 ——>Actual value

#2.1

1

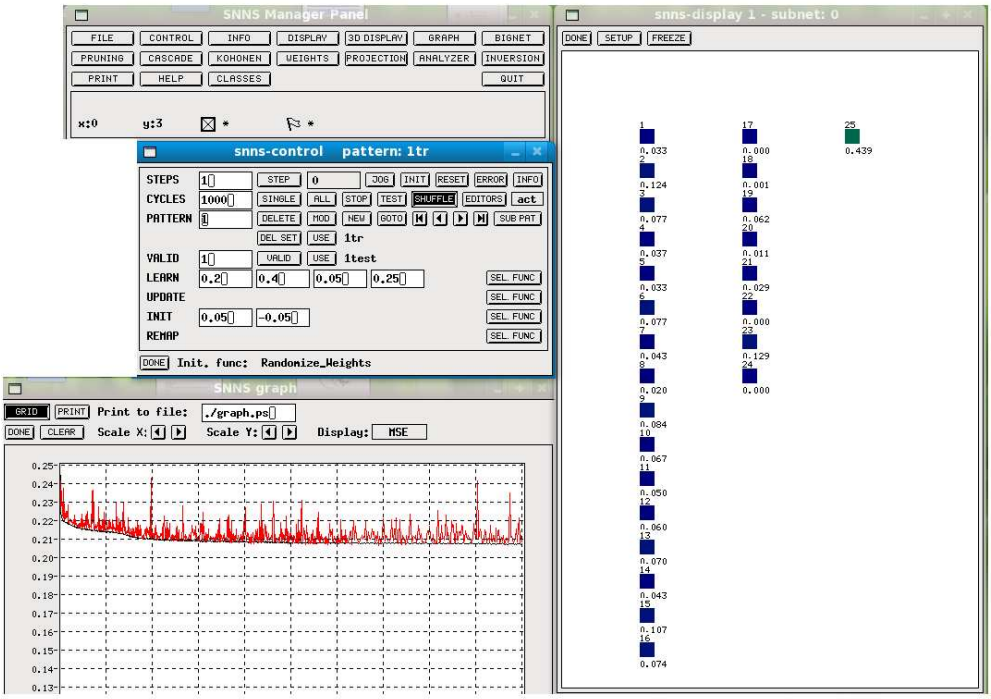


Figure C.1: SNNS simulation diagram.

.....

.....

#13525.1

0

0.76407

#13526.1

0

.....

.....

#27016.1

0

0.47119

**In this way we are calculate results for pattern files of n=3,4,5.**