2004

Protein family classification using multiple-class neural networks.

Xi Zhang
University of Windsor

Protein Family Classification Using Multiple-class Neural Networks

By

Xi Zhang

A Thesis
Submitted to the Faculty of Graduate Studies and Research
through the School of Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science at the
University of Windsor

Windsor, Ontario, Canada
2004

Your file Votre référence
ISBN: 0-612-92459-9
Our file Notre référence
ISBN: 0-612-92459-9

Abstract

The objective of genomic sequence analysis is to retrieve important information from the vast amount of genomic sequence data, such as DNA, RNA and protein sequences. The main task includes the interpretation of the function of DNA sequence on a genomic scale, the comparisons among genomes to gain insight into the universality of biological mechanisms and into the details of gene structure and function, the determination of the structure of all proteins and protein family classification. With its many features and capabilities for recognition, generalization and classification, artificial neural network technology is well suited for sequence analysis.

At the state of the art, many methods have been devised to determine if a given protein sequence is member of a given protein superfamily. This is a binary classification problem, and efficient neural network techniques are mentioned in literature for solving such problem. In this Master's thesis, we consider the problem of classifying given protein sequences into one among at least three protein families using neural networks, and, propose two methods: "Pair-wise Multiple Classification Approach" and "Single Network Approach" for this problem. In "Pair-wise Multiple Classification Approach", several sub-networks are employed to perform the task whereas a compact network system is used in "Single Network Approach". We performed experiments, using SNNS and UOWNNS neural network simulator on our NNs with different input/output representation, and reported accuracies as high as 95%.

Acknowledgements

First of all, I wish to express my deep sense of gratitude to Dr. Dan Wu, Dr. Adnan Ali and Dr. Luis Rueda, for their guidance, elaborate instruction and valuable technical assistance through the course of this thesis. They have also provided critical reviews of this material enhancing the quality of this thesis.

I would also like to thank Dr. Alioune Ngom, my advisor, for his encouragement, support and inspiring idea on this thesis. His kind instruction and comments and other assistance in the preparation of this thesis is invaluable. His influence pervades this thesis. Without his consideration and assistance, this thesis would still remain a challenge.

Also, great appreciation is expressed to my parents, sisters, and brothers for their enduring love, encouragement, and support, along the long way, without which I would be too weak to do anything.

Table of Contents

List of Tables

List of Figures

1. Introduction

Nucleic acid and protein sequences contain a wealth of information of interest to molecular biologists since the genome forms the blueprint of an organism. Genome is the complete set of all genetic material present in the chromosomes of an organism. As the molecular data continues to grow exponentially due to the Human Genome Project as well as for other model systems, the biological science has entered a new era of genomics. It greatly affects the way biology and medicine will be explored in the next century and beyond.

The dawn of the genome era began in the early 1990s when the Human Genome Organization (HUGO) organized international efforts to start the Human Genome Project (HGP). The goal of the HGP was to sequence entire human genome and to determine the biological functions of its genes. This major research effort focused on large-scale genome sequence analysis. The first complete genome of a living organism, *H. influenza* was published in 1995 (Fleischmann et al., 1996). Three more genomes from *S. cerevisiae* (Goffeau *et al.*, 1996), *M. jannaschii* (Bult *et al.*, 1996) and *M. genitalium* (Fraser *et al.*, 1995) were completed the following year. In mid 2000, the first human draft genome sequence was released and published in early 2001 (International Human Genome Sequence Consortium, 2001). The genomic analysis of several other model organisms, including fruit fly (*Drosophila melanogaster*), mouse, and a flowering plant (*Arabidopsis thaliana*), is also underway.

In order to fully understand and analyze the meaning of genomic data, computational methods are needed to identify the relevant features and information within the sequences and to provide insight into their structures and functions. In this regard, conventional computer science algorithms have been useful. But they are becoming unable to address many sequence analysis problems due to the complexity of biological system. The newly emerged field of computational molecular biology is known as *genome informatics*, which can be defined as the systematic development and application of computing systems and computational solution techniques for analyzing genomic sequences.

On the other hands, artificial neural networks have attracted lots of interest from researchers across different disciplines because it provides a unique computing architecture. Useful applications have already been designed, built, and commercialized, and much research continues in hopes of extending this success. The field of neural networks developed from the study of biological neural networks in human brain. Neural networks (NNs) is an important aspect in the field of Artificial Intelligence and Machine Learning. One can simply view a neural network as a massively parallel distributed processor that has a natural propensity for storing knowledge and making it available for use. Knowledge is acquired through a learning process. Interneuron connection strengths known as synaptic weights are used to store the knowledge. A NNs learns the relationship between inputs and outputs and stores important features of the inputs. As a technique for computational analysis, neural network technology is very well suited for the analysis of genomic data.

This thesis focuses on protein family classification using multiple-class neural networks. The protein family classification problem can be divided into two categories: binary classification problem and multiple classification problem.

The binary classification problem can be stated as: Given an unlabelled protein sequence S and a known superfamily F: Determine whether or not sequence S belongs to superfamily F. The multiple classification problem is defined formally as: Given an unlabelled sequence S and known superfamilies F_1, ..., F_N: Determine the superfamily of S.

Several networks architecture and input/output representation are examined and compared with each other in this thesis. In particular, we have designed one efficient multiclass network architecture for protein family classification, called Pair-wise NN. The Pair-wise NN is composed of smaller binary NNs that classify pair of protein classes. The individual results of the binary NN, are combined in some ways as to minimize the global classification errors. Such Pair-wise architecture has achieved the best classification results, compared with a single compact multiclass NN. We have also investigated back-propagation NN with Error-Correcting Output-Code (ECOC) for protein classification and compared it with Pair-wise NN and with single NN (without ECOC).

This thesis report is divided into nine chapters. Chapter two describes the background of genomic sequence analysis in bioinformatics. Some basic concepts and definitions as

well as challenges of genomic sequence analysis are presented in this chapter. The third chapter introduces some basic knowledge about proteins. The fourth chapter describes basic neural network background, terminology, neural network models and some definitions. The fifth chapter surveys existing techniques of protein family classification. The sixth chapter discusses the main design issues of NNs for classify protein families. The seventh chapter proposes two methods for multiple protein family classification using Neural Networks. The experiments, results, and analysis are described in chapter eight and conclusions and future work are summarized in chapter nine. Finally the references and appendices are attached.

2. Genomic Sequence Analysis in bioinformatics

2.1 Background

As a new integrated multidisciplinary field, *bioinformatics* is not only a new area of computer science which draws from the field of computational theory, artificial intelligence, machine learning, dynamic programming, but also a new approach of life science drawing from biochemistry, genetics and structural biology. Bioinfomatics is applied in the genome project. As depicted in Figure 2.1, major study in genomic sequence analysis includes: *gene recognition*, the identification of genes; *functional analysis*, the interpretation of the functions of DNA sequence on a genomic scale; *structural determination*, the determination of the tertiary structure of all proteins; and *family classification*, the classification of proteins.

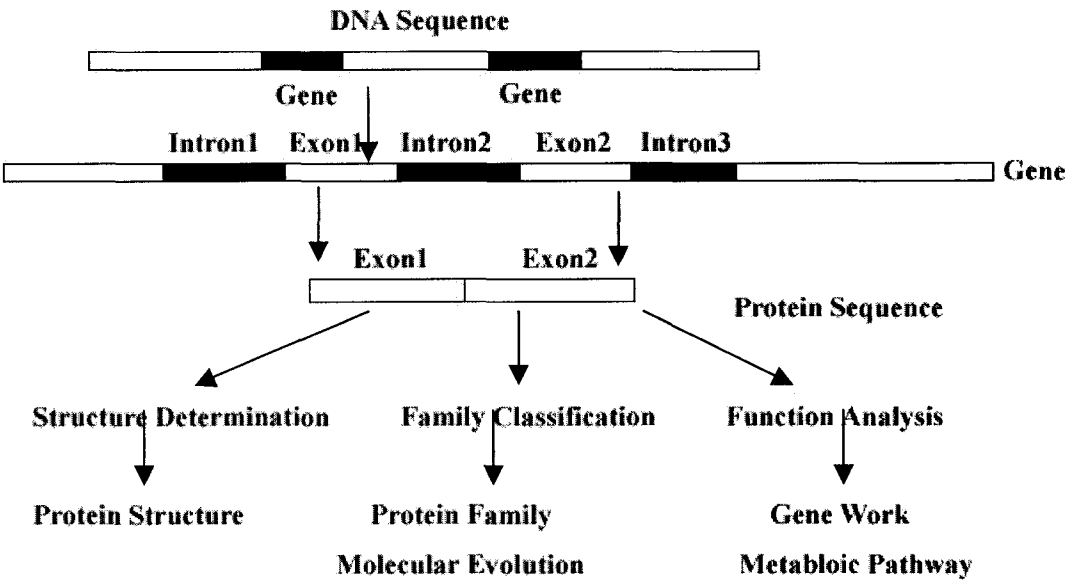


Figure 2.1 *The major studies in genomic sequence analysis*

2.2 Gene Recognition

The identification of DNA functional elements as well as signals recognized by transcriptional, splicing and translational machinery is called gene recognition, an active area of research in computational molecular biology for more than 15 years. The functional elements include promoters (eukaryotic and prokaryotic promoters) exons (initial, internal, and terminal exons), introns, 5' and 3' untranslated regions, and intergenic regions. It has been observed that coding DNA sequences exhibit characteristic context features that distinguish them from non-coding sequences, such as the codon usage, base composition, and periodicity (Konopka, 1994). Signal information refers to the DNA functional signals, sites or motifs. They may relate to basal gene biochemistry and are common to all or most genes, such as the TATA-box and cap site in eukaryotic RNA polymerase II promoters, donor and acceptor splice sites, translation initiation and termination signals. In addition to these general features, there are specialized signals related to transcription or splicing. They may be complex signals for transcriptional initiation of specific classes of genes, and may relate to regulation of gene expression, such as transcription factor binding sites or sites recognized by other DNA- and RNA-binding proteins.

Many early approaches to the gene recognition problem focused on predicting individual functional elements in isolation, using either the gene search-by-content or gene search-by-signal method (Staden, 1984). In the search-by-content approach, the characteristic context features (coding potential) are measured by functions that calculate, for any window of a sequence, a number of vectors that measures attributes correlated with a

protein coding function. In the search by signal approach, functional signals are often represented by consensus sequences or position-weight matrices (Stormo, 1990a; 1990b; Waterman & Jones, 1990; Day & McMorris, 1993). For example, weight matrices have been derived for the TATA-box, GC-box, and cup signal in eukaryotic polymerase II promoters (Bucher, 1990). More recently, a number of gene prediction programs have been developed based on the analysis and integration of multiple types of content and signal information as well as gene structure. These programs include GeneID (Guigo et al., 1992), GRAIL (Xu et al., 1994), GeneParser (Snyder & Stormo, 1995), GENSCAN (Burge & Karlin, 1997) and MORGAN (Salzberg et al., 1998). Programs containing gene structure information achieve better result than programs that do not use gene structure information (Wu, 2000).

2.3 Protein Structure Prediction

At present, the three-dimensional structures of only a small fraction of known proteins are available, although experimental structure determination has improved. With rapid producing speed of genomic sequences, there is imbalance between the number of known protein sequences and the number of experimentally determined protein structures. One of the main challenges in genome informatics is to reduce the imbalance by predictions. There are two problems in this area, *protein folding problem* and *inverse folding problem* (threading). The protein folding problem, a difficult problem to resolve due to the size and complexity of proteins, is to predict the three-dimensional structure of a given amino acid sequence. For example, with an average of m equally likely conformations per

amino acid residue and n residue in the protein, the total number of possible conformations will be m^n. For a random coil polypeptide consisting of 100 amino acid residues, there are about 8^{100} different possible conformations. The inverse folding problem is to predict if a new sequence match a given structure, which assumes that many different sequences fold in similar ways and there is a relatively high probability that a new sequence possesses a previously observed fold.

Many research in the prediction of protein structure have concentrated on predicting the elements of secondary structure because 90% of the residues in most proteins are involved in three classes of secondary structures, the α-helices, β-strands or reverse turns. Helices and sheets are termed regular structures because their residues have repeating main-chain torsion angles, and their backbone groups are arranged in a periodic pattern of hydrogen bonding. An α-helix is a periodic polypeptide structure with 3.6 residues per turn and a pitch of 1.5 A per residue. Sheets in proteins are almost invariably buried structures. In contrast to helix and sheet, turns are non-regular structures with non-repeating backbone torsion angles. Remaining residues are often loosely classified as random coil.

2.4 Protein Family Classification

Similarity between the query sequence and another sequence or database of sequences provides clues for exploring the biological knowledge hidden in them. A range of algorithms is available to solve somewhat different sequence comparison problems. In

(Needleman-Wunch, 1970; Sellers, 1974; Smith-Waterman, 1981; Altschul et al., 1990), Needleman-Wunch, Sellers, Smith-Waterman and Altschul are the most sensitive algorithm for alignment but computationally intensive. Recently, the faster methods of BLAST (Altschul et al., 1997) and FASTA (Pearson & Lipman, 1988, Pearson, 1991) have emerged. There are two search methods in database searching, direct searching and family searching. The former is based on pair-wise sequence comparisons, whereas the latter is based on comparisons of protein motifs, domains or families. So, protein family classification provides an effective mean for understanding gene structure and function, and, for the systematic studies of functional genomics. The classification approach is very useful in annotating the newly sequenced genes. For example, Nakata annotated the protein sequences with zinc finger motif using classification method (Nakata, 1995). Furthermore, the information about given families can improve the identification of genes greatly, which are otherwise not easy to detect based on pair-wise alignment method. Finally, gene families are essential for phylogenetic analysis and comparative genomics.

Protein family relationships are studied at three different levels: *superfamily*, *domain* and *motif*. *Superfamily* or family is used to indicate full-length end-to-end sequence similarity at the whole protein level; *domain* to indicate local similarity at the functional or folding unit level; and *motif* to indicate short local similarity at the functional or active site level. Recently, many family databases have been compiled, such as the PIR-International Protein Sequence Database with superfamily organization (Barker et al., 1999); domain databases, such as Pfam (Bateman et al., 1999), ProDom (Corpet et al., 1999), and DOMO (Gracy & Argos, 1999); motif databases, such as PROSITE

(Hofmann et al., 1999), Blocks (Henikoff et al., 1999), and PRINTS (Attwood et al., 1999); as well as databases combined with family classification, such as ProClass (Wu et al., 1999). Several protein family search methods have been developed, such as BLOCKS (Henikoff & Henikoff, 1994), hidden Markov modeling (Eddy et al., 1995), neural network (Wu, 1996). Correspondingly, the family information may be stored as blocks, hidden Markov models, or neural network weight matrices.

3. Protein Basics

3.1 Definition, Sizes and Shapes

Protein is a molecule composed of amino acid residues in a specific sequence by peptide bonds. Proteins take primary essential part in structure and function of cells, tissues and organs. The linear arrangement of the amino acids comprises the primary structure of protein. There are 20 amino acids found in protein which have a common structure in which a carbon atom (α-carbon) is linked to a carboxyl group, a primary amino group, a proton and a side chain (R) which is different in each amino acid (Figure 3.1). An amino acid residue is an amino acid in a peptide or protein linkage.

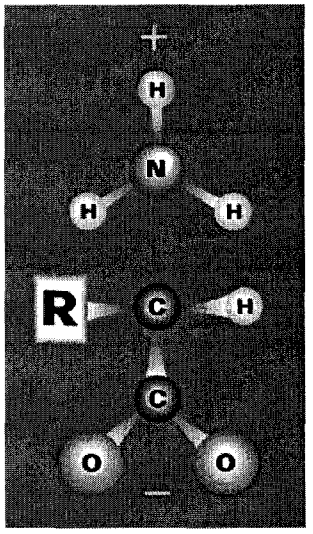


Figure 3.1 *General structure of an amino acid*

To a large extent cells are made of protein, which constitutes more than half of their dry weight. There are two broad classes of protein: globular protein and fibrous protein.

Globular protein are folded, compacted, and behave in solution more or less as spherical particles; most enzymes are globular in nature. Fibrous proteins have very high axial ratios (length/width) and are often important structural protein, for example silk fibroin and keratin in hair and wool. Molecular masses can range from a few thousand Daltons (Da) (e.g. the hormone insulin with 51 amino acids has a molecular mass of 5734 Da) to at least 5 million Daltons in the case of the enzyme complex pyruvate dehydrogenase.

3.2 Primary Structure

The amino acid sequence is called the *primary structure* of the peptide. The α-carboxyl group of one amino acid is covalently linked to the α-amino group of the next amino acid by a peptide bond. As a result, a *dipeptide* is produced when two amino acid residues are linked in this way. Many amino acids linked by peptide bonds form a polypeptide (Figure 3.2). The repeating sequence of α-carbon atoms and peptide bonds provides the backbone of the polypeptide while the different amino acid side chains play the functional roles of the protein. Typical sizes for single polypeptide chains are within the range 100-1500 amino acids, though longer and shorter ones exist.

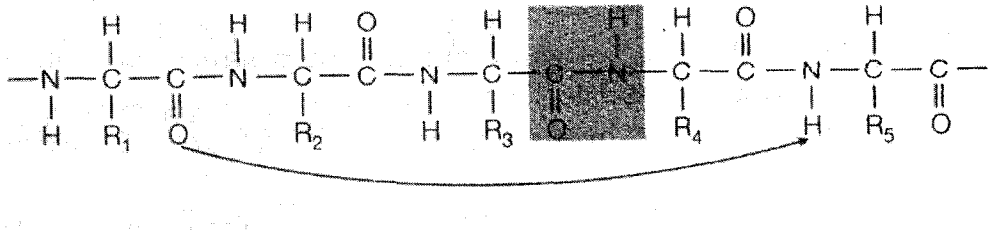


***Figure 3.2** Section of a polypeptide chain*
(Adopted from Alberts et al., 1994)

3.3 Secondary Structure

Because of the highly polar nature of the C=O and N-H groups of the peptide bonds, regular hydrogen-bond interactions exist within contiguous stretches of polypeptide chain, which makes polypeptide chain to form a number of regular structures such as α-helices and β-sheets. They comprise the protein's *secondary structure.* In the α-helix structure, the polypeptide backbone forms a right-handed helix with 3.6 amino acid residues per turn such that each N-H peptide group is hydrogen bonded to the C=O peptide group that is three residues away (Figure 3.3). Sections of α-helical secondary structure are often found in globular proteins and in some fibrous proteins. The β-sheet structure is formed through hydrogen bonding of the N-H and C=O peptide groups to the complementary peptide groups of another section of the polypeptide chain (Figure 3.4). In a polypeptide sequence, if the amino acid residue which is connected to the end of the sequence by its amino group, leaving it with a free carboxy group, it is called C-terminal. If the amino acid is connected to the end of sequence by its carboxy group, leaving it with a free amino group, we call it as N-terminal. Several sections of polypeptide chain may be involved side-by-side, giving a sheet structure with the side chains (R) projecting alternately above and below the sheet. If these sections run in the same directions (e.g. N terminus → C terminus), the sheet is parallel; if they alternate N → C and C → N, then the sheet is antiparallel. β-sheets are strong and rigid and are important in structural proteins such as silk fibroin.

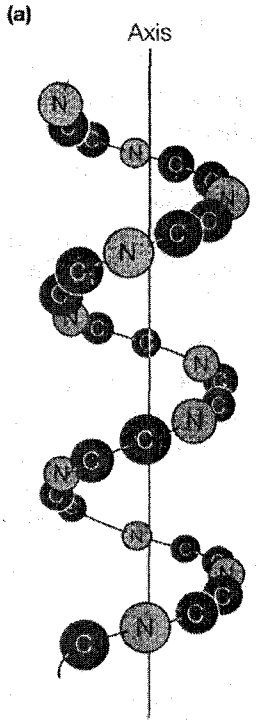


Figure 3.3 *α-Helix secondary structure*
(Adopted from Turner, 2000)

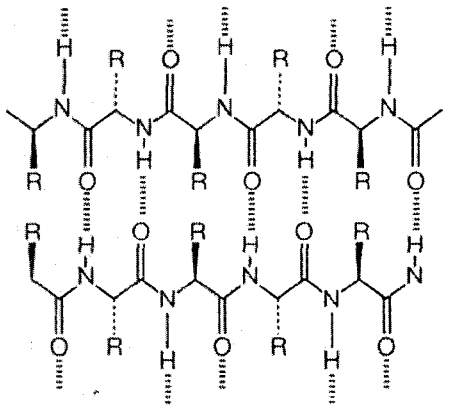


Figure 3.4 *Section of a β-sheet secondary structure*
(Adopted from Turner, 2000)

3.4 Tertiary Structure and Quaternary Structure

Based on the way in which the different sections of α-helix, β-sheet, other minor secondary structures and connecting loops, a protein molecule folds further into an irregular and particular conformation which is called the *tertiary structure* of the polypeptide (Figure 3.5).

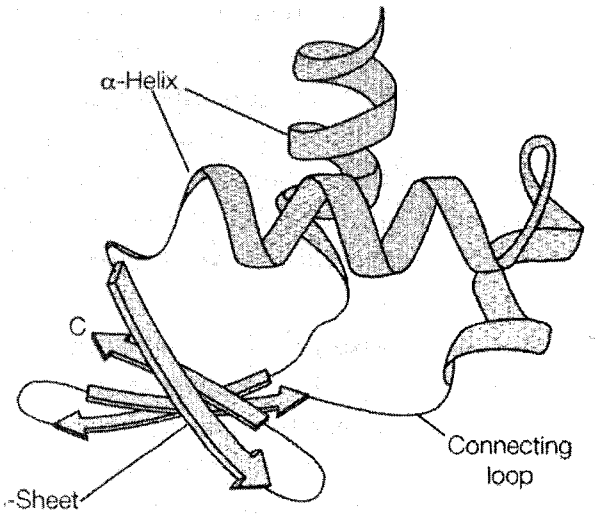


Figure 3.5 *Schematic diagram of a section of protein tertiary structure*
(Adopted from Alberts et al., 1994)

Many proteins are composed of two or more polypeptide chains (subunits). For example, Hemoglobin has two α-globin and two β-globin chains ($\alpha_2\beta_2$). The same forces which stabilize tertiary structure hold these subunits together, including disulfide bonds between cysteines on separate polypeptides. This level of organization is known as the *quaternary structure.*

3.5 Domains, Motifs and Families

Domains are structurally independent units, which are particular regions along a single polypeptide chain that are associated with certain biological activity. Domains indicate local similarities at the functional or folding level, such as the catalytic domain or the binding domain. Sometimes domains can be connected together into multi-domain complexes.

Structural *motifs* (also known as supersecondary structures) are groupings of secondary structural elements. They often have functional significance and represent the essential parts of binding or catalytic sites that have been conserved during the evolution of protein families from a common ancestor. A common example is the βαβ motif in which the connection between two consecutive parallel strands of α β sheet is an α-helix (Figure 3.6). Other motifs consist of only a few conserved, functionally important amino acids rather than supersecondary structures.

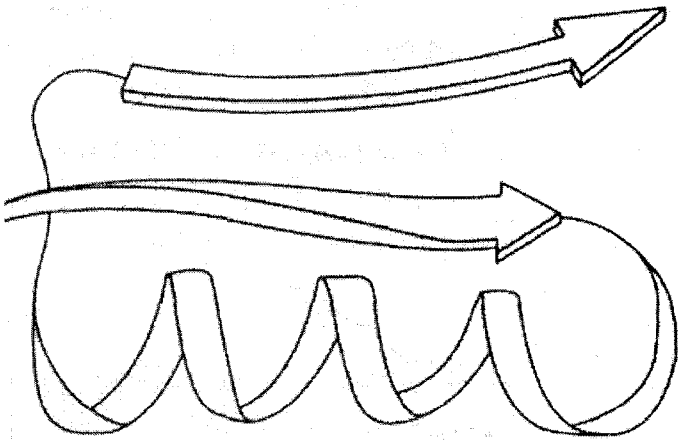


Figure 3.6 *Representation of a βαβ motif*
(Adopted from Turner, 2000)

Protein families arise through successive duplications and subsequent divergent evolution of an ancestral gene. For example, globin family consists of Myoglobin, the oxygen-carrying protein in muscle, the α- and β-globin chains of adult hemoglobin and the γ-globins and ε-globins of embryonic and fetal hemoglobins. The degree of similarity between the amino acid sequences of equivalent members (homologs) of a protein family in different organisms (e.g. rat and mouse myoglobin) depends on how long ago the two

organisms diverged from their common ancestor and on how important conservation of the sequence is for the function of the protein.

4. Neural Network Concept

4.1. Overview of Neural Network

An artificial neural network is an information-processing system that has certain performance characteristics in common with biological neural networks (Rumelhart & McClelland, 1986). The simple processing elements in the system are called neurons or units. A neural network is characterized by its pattern of connections between the neurons (called network architecture) and its method of determining the weights on the connections (called training or learning algorithm) and its activation function. The weights are adjusted on the basis of data. In other words, neural networks learn from examples and exhibit some capability for generalization beyond the training data. This feature makes such computational models very appealing in application domains where one has little or incomplete understanding of the problems to be solved, but where training data is readily available. Artificial neural networks are viable computational models for a wide variety of problems. Based on these characteristics, neural networks have been successful applied to many domains (Murray, 1995). The main applications are: signal processing, control, pattern recognition, medicine, speech production, speech recognition, business, and others.

4.2 The Basic Elements of Neural Networks

An artificial neuron, a model representation of a biological neuron, is the basic unit within a layer. A neural network consists of groups or layers of processing units with

connections between and sometimes within layers. Like real neurons, these units have input connections and output connections, which correspond to dendrites and axons in biological neurons respectively. Also like biological neurons, neural network units have some form of internal processing that creates an output signal as a function of the input signal. However, the fundamental differences existing between biological neurons and artificial neuron network units are characterized by (1) the output from a biological neuron is a pulse signal, the output from an artificial neuron is just a number within a set range in response to the inputs, usually 0 and 1, and (2) from Figure 4.1, we can see that the output from a biological neuron is dynamically changing with time, whereas the output from an artificial neuron is changing at discrete intervals of times when its inputs are changed.

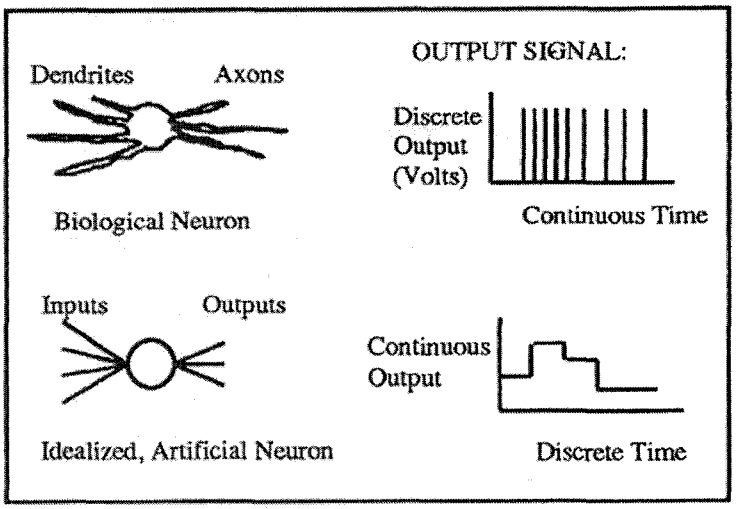


Figure 4.1 *The comparison between biological neurons and artificial neurons*
(Adopted from Fausett, Laurene V., 1994)

A well known model of neuron studied extensively in (Minsky, 1969) is called the perceptron (Figure 4.2). The perceptron computes a weighted sum of its input signals and generates an output of 1 if this sum is above a certain threshold t ∈ R. Otherwise, an output of 0 results. Generally speaking, given a weight vector $\vec{w} = (w_1,...,w_n) \in R^n$ and an input vector $\vec{x} = (x_1,...x_n) \in R^n$, such neuron computes a simple function of the form $f_{\vec{w}} : R^n \mapsto S$, where n ≥ 1, $S \subseteq R$ and

$$f_{\vec{w}}(\vec{x}) = g(\vec{w}\vec{x}) \tag{4. 1}$$

for some transfer function $g : R \mapsto S$.

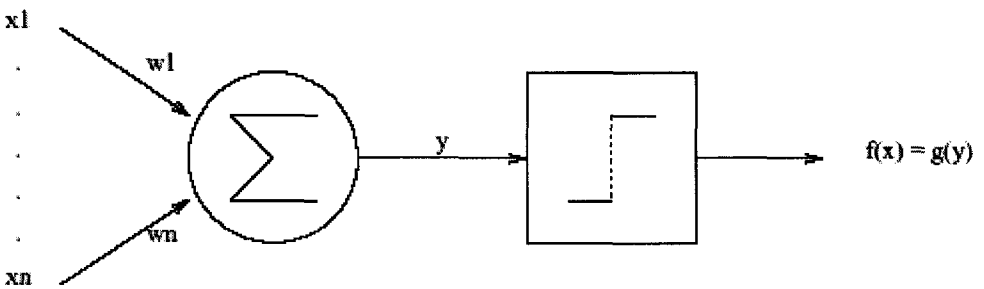


Figure 4.2 *Perceptron*

4.3 Activation Functions

There are two choices for the set S currently popular in literature. The first is the discrete model with $S = \{0,1\}$. In this case, if a neuron's threshold is t, then its transfer function g is a linear threshold function (Figure 4.2) defined by

$$g(y) = \begin{cases} 0 & if \quad y < t \\ 1 & if \quad y \geq t \end{cases} \tag{4.2}$$

and f is called a weighted linear threshold function. The second is the continuous model with S = [0, 1]. In this case, g is typically a monotonically increasing function such as the sigmoid function (Figure 4.3) given by

$$g(y) = \frac{1}{1 + a^{-(by)}} \tag{4.3}$$

for a, b $\in R^+$. The continuous model is popular because it is easier to construct. The discrete model is popular because its behavior is easier to analyze (however it uses more hardware). A neural network is characterized by the network topology, the connection strength (i.e. weights) between pairs of neurons, the neurons properties (i.e. transfer functions) and the learning algorithms.

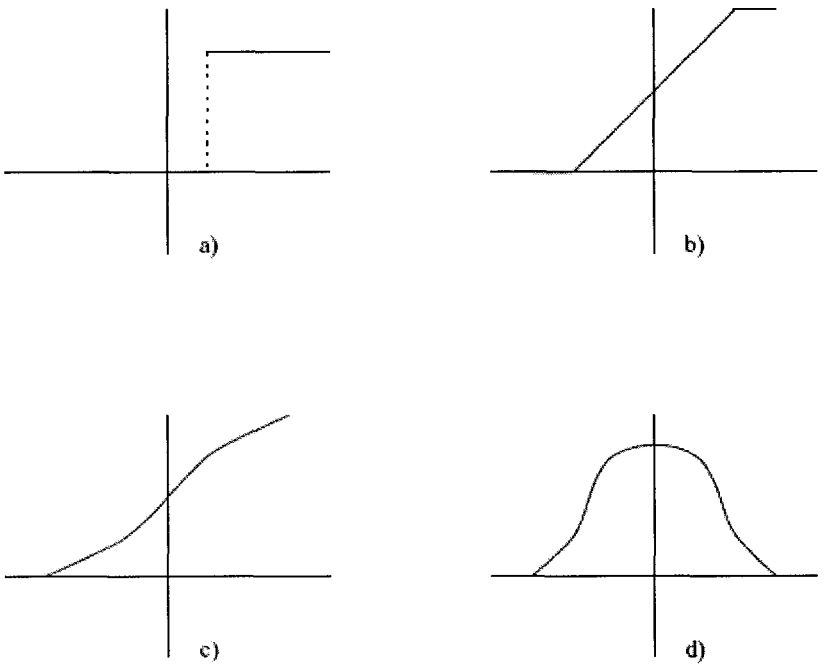


Figure 4.3: *Examples of transfer functions: a) Linear threshold function b) Piecewise linear function c) Sigmoid function d) Gaussian function*

4.4 Typical Architectures

Artificial neural networks can be viewed as weighted directed graphs in which artificial neurons are nodes and directed edges (with weights) are connections between neurons' outputs and neurons' inputs. Based on the interconnection pattern (architecture), artificial neural networks are grouped in two categories: feed-forward networks (in which graphs have no loops and cycles) and feed-back (or recurrent) networks (in which loops or cycles occur because of feed-back connections). Different network topologies yield different network behaviors and also require appropriate learning algorithms. Two types of network topologies are illustrated in the Figure 4.4.

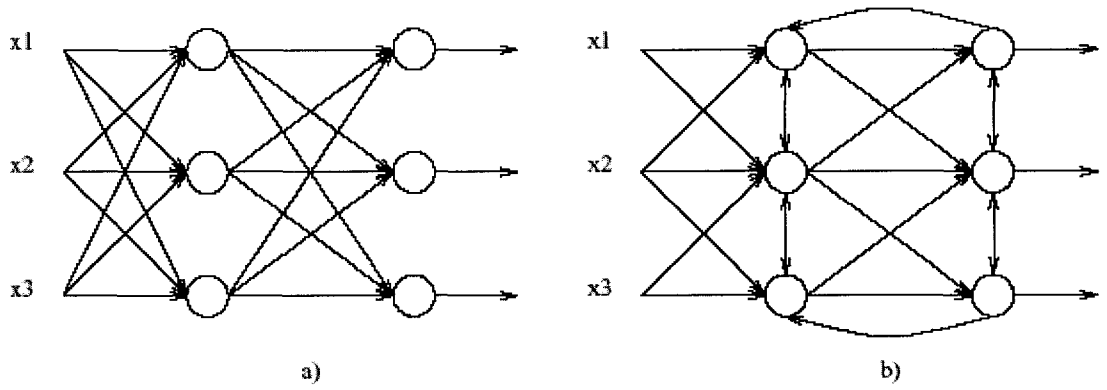


Figure 4.4 *Examples of neural networks architectures: a) Feed-forward network b) Feed-back network.*

Neural nets are often classified as single layer or multiplayer. A single-layer net has one layer of connection weights. Often, the units can be distinguished as input units, which

receive signals from the outside world, and output units, from which the response of the net can be read. In the typical single layer net shown in Figure 4.5.

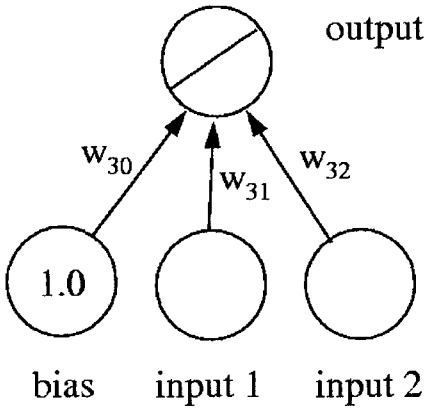


Figure 4.5 *Single-layer Net*

A multilayer net is a net with one or more layers of nodes between the input units and output units. A simple multilayer network is illustrated in Figure 4.6. These layers are called hidden layers because they are not connected directly either to network inputs or outputs. Typically, there is a layer of weights between two adjacent levels of units (input, hidden, or output). Multilayer nets can solve more complicated problems than can single-layer nets, but training maybe more difficult because the property of differentiability is essential to training multilayer network.

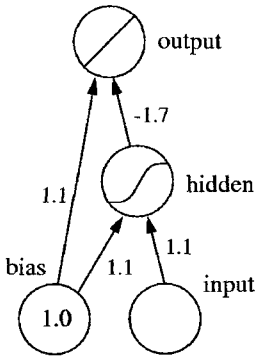


***Figure 4.6** Multilayer Net*

4.5 Training of Neural Networks

Before neural networks are able to accomplish a certain task, they must be trained to do so. Therefore, the process of building neural networks includes the training algorithms (also called learning), just as biological neurons are trained to perform certain tasks. Learning algorithms are generally divided into two types, supervised and unsupervised.

4.5.1 Supervised Training

For supervised training, inputs and expected outputs are applied to the network, and the error between the network outputs and the expected outputs is calculated and then used to adjust the weights in such a way to minimize the error. Once the network has been trained using training examples, it is then validated using test examples so as to further minimize the network error and to ensure that it will correctly classify future unseen examples.

A typical error function to be minimized is

$$E = \sum_{i=1}^{n} (o_i - t_i)^2 \tag{4.4}$$

where n is the number of input data vectors, o_i and t_i are the network output (for a given set of parameters and input vectors) and target values, respectively.

The method of adjusting the weights, so as to minimize the network error, is called the learning rule.

Different network architectures yield different learning rules, and therefore, there are different supervised learning algorithms. The most well-known supervised learning algorithms are the back-propagation algorithm for multilayer feed-forward networks, and the perceptron algorithm for single discrete neurons.

4.5.3 Unsupervised Learning

In unsupervised learning no expected output values are provided during the training process of a neural network. The goal of unsupervised learning is to cluster the data according to their similarities. The Kohonen self-organizing map algorithm (Kohonen, 1990) is one such example of unsupervised learning algorithm.

4.5.4 Hybrid Training Algorithm

Supervised and unsupervised learning may coexist in a given architecture. The counter-propagation network (Hecht-Nielson, 1987) is an example in which layers from supervised and unsupervised learning paradigms are combined to construct a new type of network.

An example of hybrid training is firstly to cluster the data by unsupervised training and then apply supervised training to classify the data into their correct clusters. There are many other methods of hybrid training.

5. Overview of Existing Techniques of Protein Family Classification

The existing method for classifying protein family is the binary classification, that is, to classify target class and nontarget class. The problem can be stated as: Given an unlabelled protein sequence S and a known superfamily F: Determine whether or not sequence S belongs to superfamily F. Obviously, in practice it is too cumbersome to compare the unlabelled protein sequence with every superfamily one by one. Here, we propose two new methods for classifying multiple protein families using neural networks. The problem is defined formally as: Given an unlabelled sequence S and known superfamilies F_1, ..., F_N: Determine the superfamily of S.

5.1 Naive Approach

The idea is to compare the unlabeled sequence S with the sequences in the target class and the sequences in the nontarget class using an alignment tool such as BLAST (Altschul et al., 1997) and then assigns S to the class containing the sequences that best matches S. In this approach, the sequences in nontarget can be the sequences from various families that are totally different from the target class.

5.2 Hidden Markov Models Approach

This method for protein sequence classification is based on hidden Markov models (HMMs) (Krogh et al., 1994). HMMs are probabilistic graphical methods used to

describe sequence data or time-series. HMMs have been applied to many other molecular domains, including areas such as RNA secondary structure prediction, protein structure prediction. For protein classification, an HMM is built for each (super) family, and then the unlabeled sequence S is scored with respect to a given family HMM. If the score is more significant than a cut-off value, then S is regarded as a member of the (super) family (Sonnhammer et al., 1997).

5.3 Combination Approach

Another approach for protein sequence classification is to iteratively build a model either based on hidden Markov models or based on a position-specific weight matrix from BLAST. The unlabeled sequence S is used as a seed sequence and iteratively searched against the (super) family either by the HMM or by the position-specific weight matrix.

5.4 Neural Network Methods

5.4.1 Unsupervised Approach

In (Ferran &Ferrara, 1992), the authors employed an unsupervised approach to address the protein sequence classification problem. Self-organizing Kohonen maps was used for clustering proteins into "natural" groups. The inputs to the network were bi-peptide amino acid frequencies of protein sequences from 13 different families. After training, a 7×7 feature map of protein family clusters could be obtained, which was 96.7% accurate

according to the corresponding families. It was shown that the networks provided a good method to work with several unknown protein families. In a subsequent study, Ferran & Pflugfelder (1993) reduced the number of inputs from 400 into 20 principal components using principal component analysis (PCA). The feature map was configured to be 7×4 according to a statistical clustering analysis. The neural network-statistics hybrid system resulted in a much smaller network and faster training, with similar results. Further improvements were made in (Ferran et al., 1994), the modification was that the number of alphabet of amino acid was reduced from 20 to 11 using amino acid groups. With this method, the network classified all human protein sequences into a 15×15 feature map. The neural network approach was shown to be better than statistical non-hierarchical clustering method on this complex data.

5.4.2 Supervised Approach

Surveys on protein family classification using supervised approach can be found at the series of papers (Wu et al., 1992; 1995; 1996). Also, in these papers, the authors devised a full-scale neural network system for the automatic classification of more than 3,300 proteins from Protein Information Resource (PIR, http://pir.georgetown.edu/). The protein sequences are encoded into neural input vectors by counting the occurrences of *n*-gram words. Furthermore, SVD (singular value decomposition) method is applied to improve the generalization capability of the network. SVD compresses the long and sparse *n*-gram input vectors and captures the semantics of *n*-gram words. For example, given a data set of 100 training protein sequences, then according to the 2-gram encoding

method, a 400 (20^2)× 100 input vector matrix would be produced. With the SVD method, the matrix could be reduced to 50 × 100 neurons. In (Wu et al., 1996), a motif identification neural system (MOTIFIND) (Figure 5.1) was developed for detecting distant family relationships through the use of both global and motif sequence information.

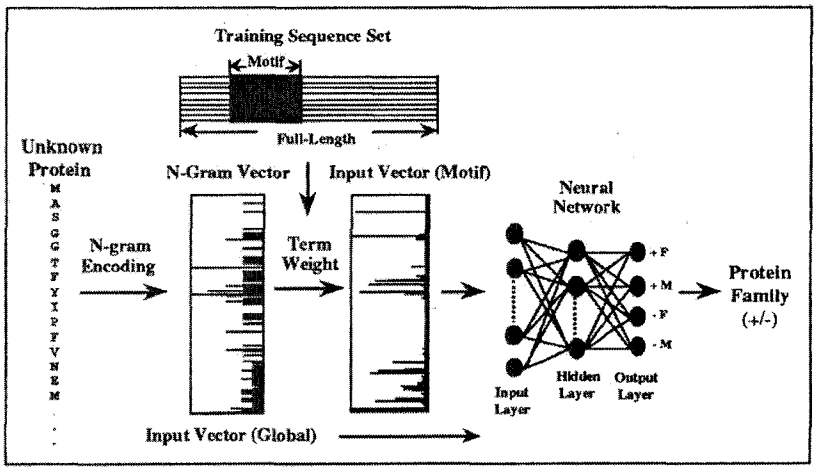


Figure 5.1 *MOTIFIND neural network for protein family classification*

(Adopted from Wu et al., 1996)

The difference with their previous methods is that an n-gram term weighting algorithm is used to extract conserved family information from local sequence motifs. An example is shown in Figure 5.2. For example, for the 2-gram 'CH', the frequency it appears in the same column is 5 and the frequency it appears in the whole data set is 5. Then, term weight can be calculated as: 5/5 = 1.00.

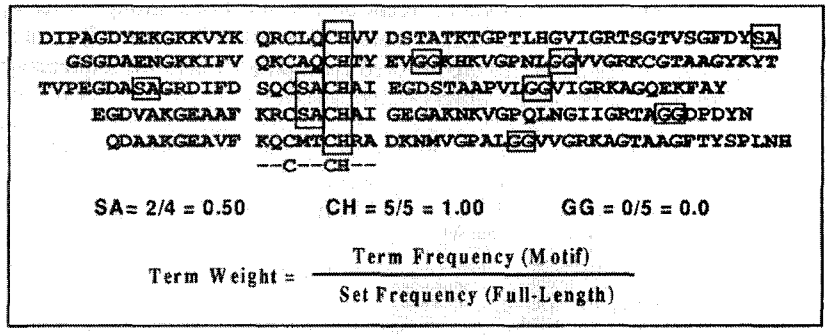


***Figure 5.2** N-Gram term weighting algorithm to extract motif information*

The final network architecture was 1696 × 20 × 4 three-layered feed-forward back-propagation neural network. The 1696 input units came from the weight of bi-grams of amino acids (20^2 units, the combinations of obtaining a 2-gram from 20 amino acids) and tetra-grams of exchange groups (6^4 units, the combinations of obtaining a 4-gram from 6 letters in the exchange group). The network has four output units to code for the global and motif pattern of both positive (member) and negative (non-member) classes. An integrated gene family identification system (GeneFIND) that combines MOTIFIND neural networks and the ProClass family database (Wu et al., 1999b) was devised for database searching against protein families in the paper (Wu et al., 1999a). With this method, protein family classification is performed at the protein level, the domain level and the motif level.

5.4.3 Supervised Way & Unsupervised Approach

A modified counter-propagation network, which employed a supervised learning vector quantizer algorithm to perform nearest-neighbor classifications, was used for molecular sequence classification (Wu et al., 1997).

6. Design Issues

6.1 Overview of Design Issues

Design issues are very important in the performance and integrity of the whole neural network system when NN techniques are applied to analyze the protein sequences. The major issues include input/output encoding, the neural network architecture, the training/prediction data set, and system evaluation mechanism (Figure 6.1).

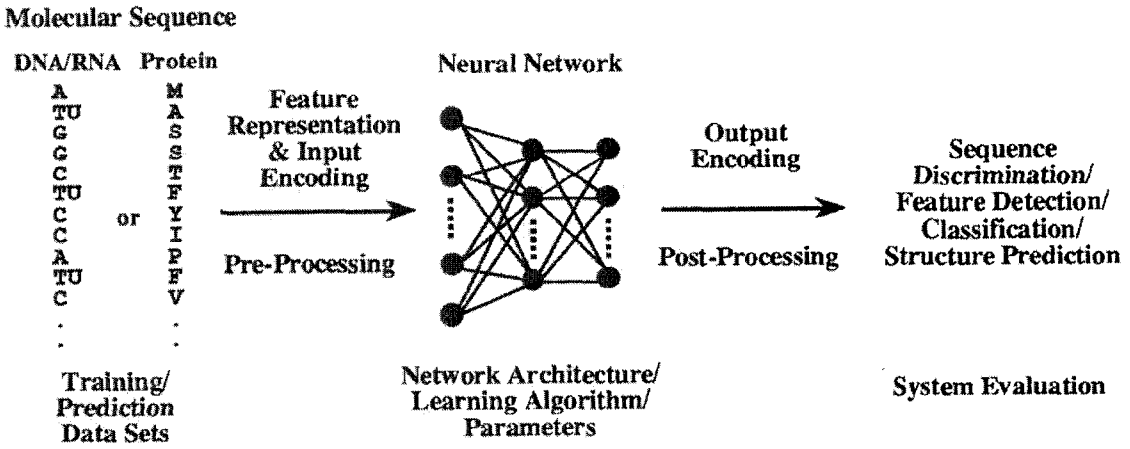


Figure 6.1 *Design issues of neural network application for protein classification*
(Adopted from Wu, 2000)

A typical design process is divided into several following steps:

- Problem Definition: determining the desired input/output mapping for the task;
- Data pre-processing: choosing appropriate feature representation and encoding methods;
- Architectures of NN and learning algorithm: selecting appropriate NN architectures, learning algorithm and parameters;

- Post-processing: choosing appropriate output encoding.

6.2 Input Sequence Encoding

6.2.1 Direct Encoding Scheme

6.2.1.1 Binary-Vector Encoding Scheme (BIN21) (Qian & Sejnowski, 1988)

In the BIN21 scheme, a binary vector is used to encode a given amino acid. Each position in the vector corresponds to a unique amino acid. The gap symbol is also encoded by the binary vector. Since there are 20 amino acids, therefore, the vector has length 21 (including the position for gap symbol). A given letter (amino acid or gap) is encoded as follow: its corresponding position in the vector is set to bit 1 and all other bits are cleared to 0. See Table 6.1 for example. Therefore, a protein of length n will be encoding using n BIN21 vectors, each representing an amino acid or gap in the protein. See Figure 6.2 for instance.

***Table 6.1** BIN21 Encoding Scheme*

Amino Acid	Vector Value
A	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
C	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
D	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0]
E	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0]
F	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0]
G	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]
H	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]

***Table 6.1** (Continued)*

J	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
K	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0]
L	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
M	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
N	[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
P	[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
Q	[0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
R	[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
S	[0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
T	[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
V	[0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
W	[0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
Y	[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
-	[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

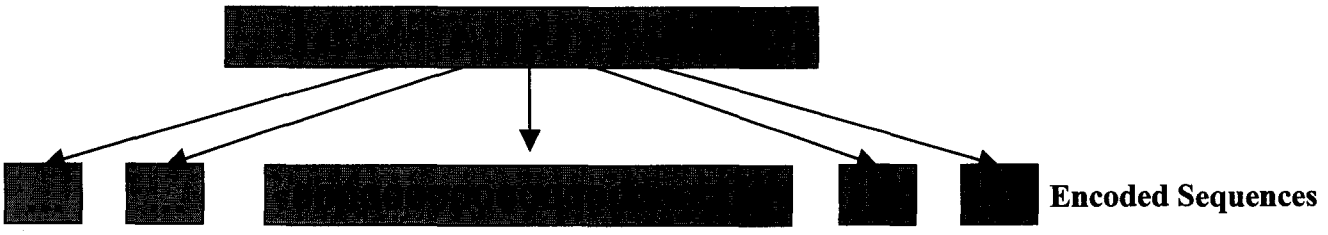


***Figure 6.2** BIN21 Encoding of a protein sequence*

6.2.1.2 Real-Vector Encoding Scheme (REAL21)

Whilst the BIN21 encoding is used to represent the amino acids of a single input sequence, the REAL21 (Rost & Sander, 1993a; Rost & Sander, 1993b; Rost & Sander, 1994a; Rost & Sander, 1994b; Rost 1996) representation is used to encode multiple sequence alignment. Each position of the multiple sequence alignment is represented by a

vector of size 21 and each entry in the vector represents the proportion of a particular amino acid at that position. An example is illustrated in the Figure 6.3, the first column of the table that lies underneath the arrow represents the number of position of the multiple alignment, the real number in each row represents the proportion of a particular amino acid at that position.

A Multiple Alignment

1	2	3	4	5	6	7	8	9	10	11	12	13
L	H	A	N	Y	K	K	E	L	D	N	L	E
L	R	A	D	F	K	K	E	L	Q	S	F	E
L	K	A	D	F	Q	E	E	L	K	A	F	D
L	K	A	N	Y	E	K	E	L	K	A	F	A
L	R	A	D	F	K	K	E	L	Q	S	F	E

Sequence Alignment							Amino Acid Profile																					
							-	A	B	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	U	V	W	Y	Z
13	E	E	D	A	E		0	0.2	0	0	0.2	0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	L	F	F	F	F		0	0	0	0	0	0	0.8	0	0	0	0	0.2	0	0	0	0	0	0	0	0	0	0	0	0
11	N	S	A	A	S		0	0.4	0	0	0	0	0	0	0	0	0	0	0	0.2	0	0	0	0.4	0	0	0	0	0	0
10	D	Q	K	K	Q		0	0	0	0	0.2	0	0	0	0	0	0.4	0	0	0	0	0.4	0	0	0	0	0	0	0	0
9	L	L	L	L	L		0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
8	E	E	E	E	E		0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	K	K	E	K	K		0	0	0	0	0	0.2	0	0	0	0	0.8	0	0	0	0	0	0	0	0	0	0	0	0	0
6	K	K	Q	E	K		0	0	0	0	0	0.2	0	0	0	0	0.6	0	0	0	0	0.2	0	0	0	0	0	0	0	0
5	Y	F	F	Y	F		0	0	0	0	0	0	0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.4	0	0
4	N	D	D	N	D		0	0	0	0	0.6	0	0	0	0	0	0	0	0	0.4	0	0	0	0	0	0	0	0	0	0
3	A	A	A	A	A		0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	H	R	K	K	R		0	0	0	0	0	0	0	0	0.2	0	0.4	0	0	0	0	0	0.4	0	0	0	0	0	0	0
1	L	L	L	L	L		0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Figure 6.3 *REAL21 Encoding of a multiple alignment*

6.2.1.1 Sliding Window for Direct Encoding

Clearly, the type of the amino acid at a given position is not sufficient to reliably predict a family, so, the prediction should take into account information about the neighbouring residues (amino acids) as well. On the other hand, the entire sequence cannot be used as input. The first reason is that most tools require that the input has a fixed size, but sequence vary in length. The second reason is that the size of the input is directly related to the amount of training examples that are necessary to train the model. Therefore, if the size of the input is too large the number of necessary examples, to train the model, will be too large, and the time needed to train the model will also be large. Generally, a fixed window size is used. In the paper (Rost & Sander, 1993a), authors used a fixed window of size 13 to predict the (2D) state for position i, not only does it considers the amino acid type at position i, but also the 6 preceding and following positions. An example is illustrated in the Figure 6.4. In this example, a window of size 3 slides through the protein sequence, and then we get the sequence triplets such as PVK, VKT, TKN. Next, these triplets are encoded by the BIN21 Encoding Scheme and used as the input to the neural networks.

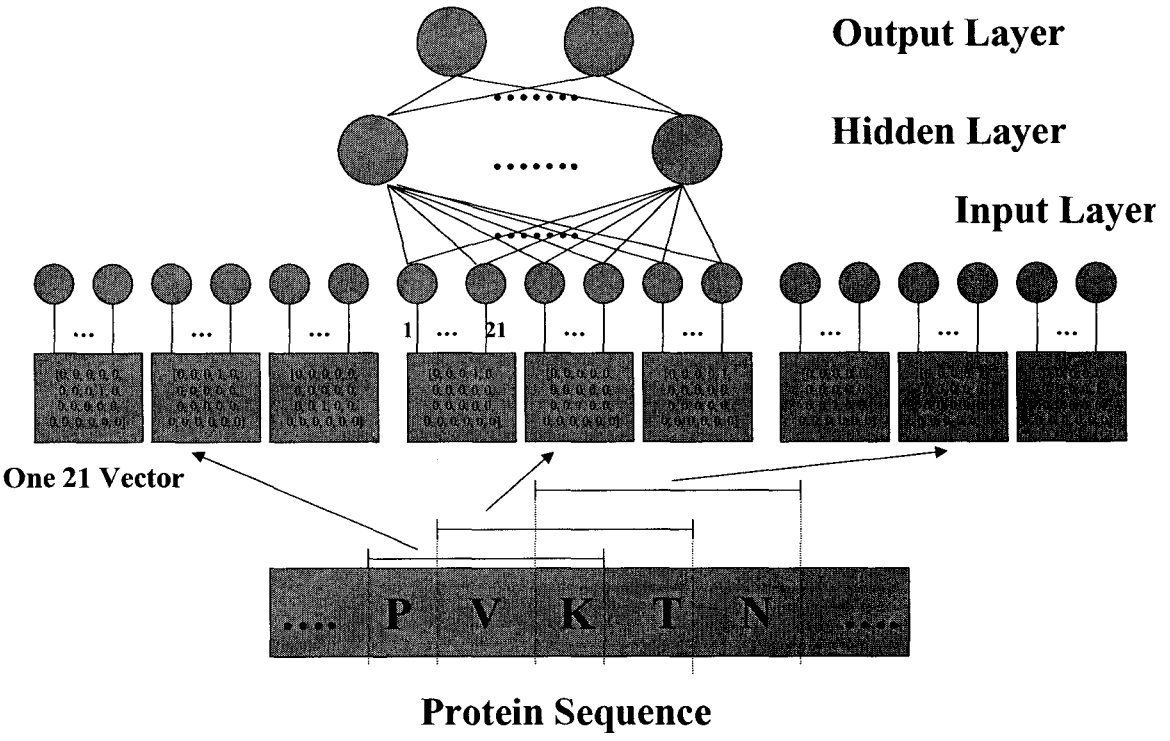


Figure 6.4 Sliding Window for BIN21 Encoding of a protein sequence

6.2.2 Gram Encoding Scheme: g-Gram (Wu et al., 1992)

In this scheme, the frequencies of g-grams are computed and then used as inputs to the neural networks. A g-gram (g > 0) is a substring of length of length g. G-gram encoding is known to be much better than BIN21 and REAL21 encodings on accuracy. They are used to extract important features of given protein families. This is due to the fact that certain g-gram appears more frequently in proteins of a given family than of other families. As an example of g-gram encoding consider the following protein, PVKTNVK.

The frequencies of the 2-grams are 1 for PV (indicating PV occurs once), 2 for VK (indicating VK occurs twice), 1 for KT, 1 for TN, and 1 for NV. See Figure 6.5.

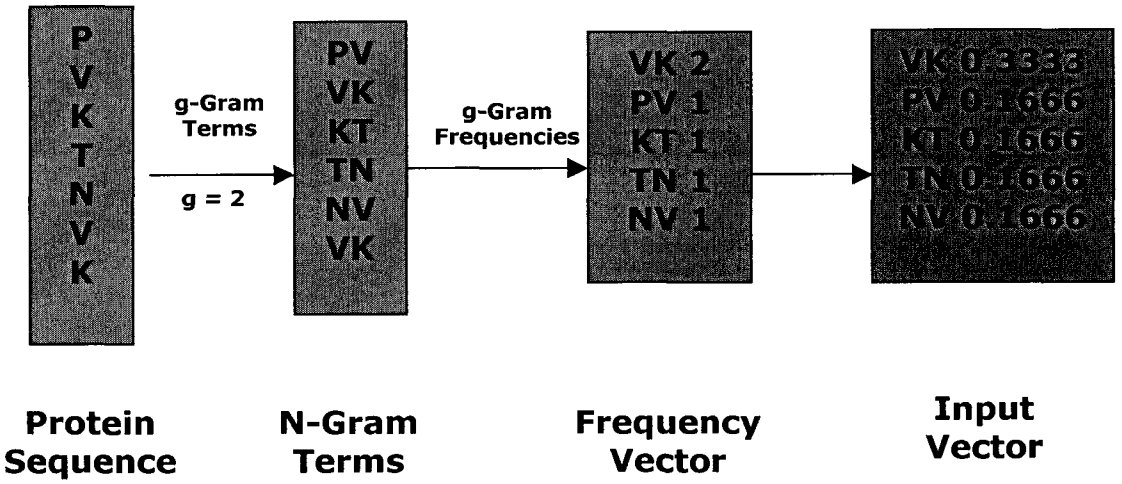


Figure 6.5 N-gram Encoding Scheme

6.3 Feature representation

A protein is a sequence of letters from the amino acids alphabet {A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y}. The alphabet has 20 letters. However these letters are not just meaningless letters. Differences in sequences and sequence length are certainly very important in classifications based on structures and functions. So, an important issue in applying neural networks to protein sequence classification is how to extract the features from protein sequences as the inputs for neural networks and connect these letters to their structures and functions. Furthermore, good feature representation makes it easier for the neural networks to recognized underlying regularities. Thus, good input representations are crucial to the success of neural network learning.

There are several methods to represent a large number of physicochemical and structural features, context features and evolutionary features containing in the protein sequences, which can be integrated into neural network systems. A simple way is to represent these features as real numbers. For example, we can represent amino acid residues in real numbers according to their mass and hydrophobicity scales or as vectors of distances or frequencies using PAM (Dayhoff et al., 1978) matrix and sequence profile. Another way is to categorize amino acid residues into classes based on certain properties. As a result, the alphabet size is reduced and the various properties of the amino acid residues are integrated, which maximizes feature extraction.

For example, in (Nakata, 1995), the author divided 20 amino acids into three group: {DENQRK}, {CSTPGHY} and {AMILVFW}. Each group represents polar, nonpolar and amphipathic based on the range of hydrophobicity in the scale:

Group1: $\geq (H_m + \sigma / 2)$;

Group 2: $\geq (H_m - \sigma /2)$ & $< (H_m + \sigma /2)$;

Group 3: $< (H_m - \sigma / 2)$;

and $H_m = \frac{1}{2}\sum_{i=1}^{20} H_i$

where H_m is the mean value of the hydrophobicity over 20 amino acids, σ is the standard deviation, and H_i is the hydrophobicity value of an amino acid in the scale.

Similarly, we can group 20 amino acids in different ways according to other physicochemical and structural properties of the amino acids. There are other groupings

in literature. For example, various amino acid groups have been described by Karlin et al. (1989) and Nakata (1995), including those for charge and polarity, hydrophobicity, hydrophilicity and secondary structure propensity. The evolutionary features of amino acids are often represented by substitution matrices, most notably the PAM (point accepted mutation) (Dayhoff et al., 1978). The PAM matrix is a frequency table representing substitution rates for closely related proteins at the particular evolutionary distance represented by multiple sequence alignments. A six-letter exchange group alphabet (Figure 6.6) derived from the PAM matrix contains information about conservative replacement in evolution and has been shown to be highly effective for protein classification (Wu et al., 1992). Examples of groupings are given in the Table 6.2.

Table 6.2 *Some examples for feature representations*

Alphabet Name	Size	Features	Membership
AAIdentity	20	Sequence Identity	A,C,D,E,F,G,H,I,K,L,M,N,P,Q,R,S,T,V,W,Y
ExchangeGroup	6	Conservative Substitution	{HRK}{DENQ}{C}{STPAG}{MILV}{FYW}
ChargePolarity	4	Charge and Polarity	{HRK}{DE}{CTSGNQY}{APMLIVFW}
Hydrophobicity	3	Hydrophobicity	{DENQRK}{CSTPGHY}{AMILVFW}
Mass	3	Mass	{GASPVTC}{DNQEHILKM}{RFWY}
Structural	3	Surface Exposure	{DENQHRK}{CSTPAGWY}{MILVF}
2DPropensity	3	2D Structure Propensity	{AEQHKMLR}{CTIVFYW}{SGPDN}

The Six-letter Exchange Group Method is illustrated in the Figure 6.6. The exchange groups consists of $\{e_1, e_2, e_3, e_4, e_5, e_6\}$ to represent a protein sequences, where e_1 corresponds to {H, R, K}, e_2 to {D, E, N, Q}, e_3 to {C}, e_4 to {S, T, P, A, G}, e_5 to {M, I, L, V}, and e_6 to {F, Y, W}. The exchange groups represent conservative replacements through evolution. For example, the given protein sequence PVKTNVK in Figure 6.6 can be represented as $e_4e_5e_1e_4e_2e_5e_1$.

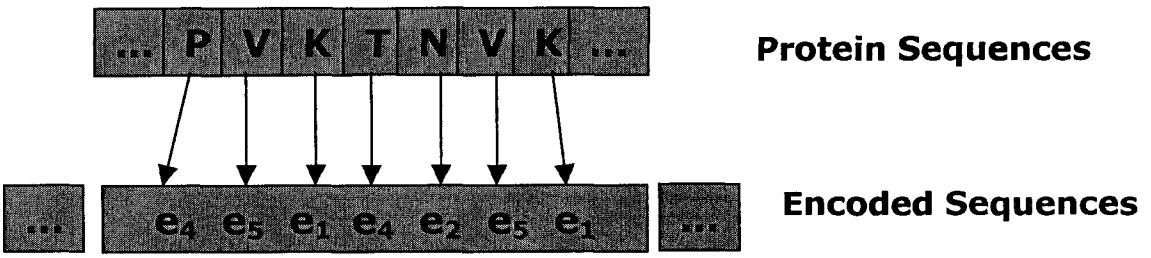


Figure 6.6 *An example of encoded sequence using the Six-letter Exchange Group Method*

6.4 Neural Networks

Compared to the traditional information-processing system, the design of a neural network system is different from its counterpart at that the design of neural networks is based directly on real data, whereas the classical one is firstly based on a mathematical model of environment observations, and then validating the models with real data, and finally building the design on the basis of model. Thus, the neural network not only provides an implicit model of the environment in which it is embedded, but also performs the information-processing functions of interest.

A neural network is characterized by:

- Its pattern of connections between the neurons (called its architecture);
- Its method of determining the weights on the connections (called its learning algorithm);
- Its activation functions.

So, there are several considerations for neural network design, including the architecture, learning algorithm, network parameters and training and test data.

6.4.1. Network Architecture

The design of network architectures is the crucial part in the design of the whole NN, its tasks include:

- The number of layers.

The number of layers depends on the dataset. If the input data is linearly separable, a two-layer architecture (one input layer and one output layer) performs well with several advantages such as its convergence properties, its simplicity and rule extraction. The network has few interconnections; thus, it requires less training time to adjust the weights, and needs fewer training patterns to achieve good generalization. The weight matrix of the network can be represented by a Hinton graph and used to decipher biological information or derive consensus sequence patterns (e.g. Qian & Sejnowski, 1988). When nonlinear input-output mapping is desired, a multi-layer network with at least one hidden layer works better. The ability of hidden neurons to extract higher-order statistics is particularly valuable when the size of the input layer is large.

- The number of processing units in each layer.

If there are a large number of neurons in each layer, consequently, the learning process becomes unconstrained and results in poor generalization with the cumbersome network. So, we have to take care of generalizability when we design a multiplayer network, although it can approximate any arbitrary mapping. On the other hand, the large network may causes data overfitting. On the other hand, a small network may not be powerful enough to learn a given task. It is still an important open problem to determine the optimal size of a network such that overfitting is minimized, generalizability is maximized and learning accuracy is maximized. See Ngom et al. (2001) for a good survey on growth/constructive algorithm for network design.

- The specific interconnections between the units.

A neural network can be fully or partially connected. Most neural networks used in the genomic sequence analysis are fully connected. In (Snyder & Stormo, 1995), the authors proposed a method of local connections. In that method, the restricted network usually has fewer free parameters available for adjustment than a fully connected network, consequently, prior information should be built into the design of a neural network, thereby simplifying the network design.

6.4.2. Network Learning Algorithm

In general, there are two ways to learn for neural networks: supervised learning and unsupervised learning.

- Supervised learning algorithm

We can consider supervised learning as "learning with a teacher". In conceptual terms, we may think of the teacher as having knowledge of the environment that is represented by a set of input-output examples. The supervised training is accomplished by presenting a sequence of training vectors, each with an associated target output vector. So, if we already know the knowledge of training vectors, we can choose the supervised learning algorithm.

- Unsupervised learning algorithm

In unsupervised learning there is no external teacher to over-see the learning process. This algorithm is used when we want to group data elements according to their similarities with each other. The training data are clustered into several groups according to some distance functions.

- Hybrid learning algorithm

Supervised and unsupervised learning may coexist in a given architecture. Some applications may benefit from using combined output from multiple neural networks trained with different learning algorithms. For example, the combination of back-propagation and counter-propagation networks resulted in better classification accuracy than that of either method alone (Wu & Shivakumar, 1994).

6.4.3. Network Parameters

There are several considerations on network parameters for optimizing the design of neural networks, including the architecture, learning the learning rate, momentum term, activation function, error function, initial weights, and termination condition. The following discussion focused on back-propagation design.

- Learning rate

A parameter that controls the amount by which weights are changed during training. In standard back-propagation, the learning rate may be constant. If its value is too small, then training progress will be too slow. A large learning rate may simply produce oscillations between solutions. In general, large rates are desirable when the network error is too large. The rate can be decreased as the error approaches zero. A sample of various approaches for selecting the proper learning rate is given in Hassoun (1995).

- Momentum

A common modification to standard back-propagation training; at each step, weight adjustments are based on a combination of the current weight adjustment, and the weight change directions of previous step. The momentum term is normally chosen between 0 and 1. Adaptive momentum rates may also be employed. Fahlman (1998) proposed a heuristic variation of back-propagation, called QuickProp, which employed a dynamic momentum rate.

- Activation function

In general, the activation for back-propagation learning algorithm should be continuous and differentiable. Common activation functions are discussed in chapter 4.

- Error function

A commonly used error function is the root-mean-square error, which is discussed in chapter 4.

- Initial weights

The back-propagation network is very sensitive to initial conditions. The choice of initial weights will influence whether the net reaches a global (or only a local) minimum of the error and, if so, how quickly it converges. In practice, the weights are usually initialized

to small zero-mean random values between –0.5 and 0.5 (or between –1 and 1 or some other suitable interval).

- Termination conditions

The training is usually stopped when the network error falls below a certain threshold, called tolerance, or a fixed number of training iterations is attained. The termination condition can also be based on the performance of the network on validation data set used to monitor the generalizability of the network during learning. In this case, learning terminates at the point when generalizability starts to decrease.

6.4.4. Training and Test Data

Generalization is concerned with how well the network classifies unseen or unknown data. Investigations have demonstrated the existence of a critical time interval in the training process where the trained network generalizes best, and after which the generalization error starts to increase. Therefore, a suitable strategy for improving generalization in networks of non-optimal size is to avoid overtraining the network by carefully monitoring the evolution of the validation error during training. Overtraining means that the network has spent to much time (iteration) on learning. Overfitting will then occurs, meaning that the network simply memorizes the training set and therefore is unable to generalize. Another solution to the overfitting problem is to reduce the complexity of numbers of units of the network to a minimum acceptability. Using more

units than is necessary is analogous to over-parameterization in statistical modeling. Noise in the input data may be learned by a network, thereby reducing its ability to generalize.

It is not particularly useful to report the results of training as the number of correctly classified training examples. Of course, results on the training data will be good if the model is reasonable; what is desired is good performance on data not used during training, i.e., generalization. For this a separate dataset, called the test dataset, is used. So, completed dataset is split into three parts: training set, validation set, and testing set.

6.4.5. Evaluation Mechanism

The evaluation of the performance of a NN system can be measured by its sensitivity, its specificity, its positive predictive value, its negative predictive value and its accuracy. They are defined as follows:

$$Sensitivity = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

$$PositivepredictiveValue = \frac{TP}{TP + FP}$$

$$NegativepredictiveValue = \frac{TN}{TN + FN}$$

$$Accuracy = \frac{TP + TN}{Total}$$

where TP is the number of true positive, TN is the number of true negative, FP is the number of false positive, and FN is the number of false negative. True positive means that the sample in the positive class is classified correctly into positive class. True negative means that the sample in the negative class is classified correctly into negative class. False positive means that the sample in the negative class is classified incorrectly into positive class. False negative means that the sample in the positive class is classified incorrectly into negative class.

7. Protein Families Classification using Multi-class Neural Networks

In this chapter, we will introduce two new NN approaches for classifying protein sequences into superfamilies. A superfamily is a group of proteins that share common properties and/or functions. As far as we know, current NN methods in literature are concerned with classifying given protein into one of two classes: a family class and a non-family class. These NN methods are therefore called as binary classification. That is given a protein sequence, they classify it as a member of a given family or not. Obviously, in practice it is too cumbersome to compare to unlabelled protein sequence with every superfamily one by one. In this thesis, we are concerned with the following multiple classification problem: given a protein sequence S and a set of superfamilies F_1, F_2, ..., F_{N-1}, F_N, determine the superfamily of S. There are many possible Neural Network approaches to the solution of this problem. Two such NN approaches are discussed in this chapter. All NN solutions to our multiple-classification problem share a common characteristic: they are multi-class neural networks, since they classify one protein into one of many families instead of one of two families.

7.1 Sequence Representation and Features Extraction

We adopt the 2-gram encoding scheme proposed by (Wu, 1996) to represent the protein sequences. The 2-gram encoding method extracts various patterns of two consecutive amino acid residues in a protein sequence and counts the number of occurrences of the

extracted residue pairs. We also adopt the 6-letter exchange group encoding scheme to represent a protein sequence. Finally, the 2-grams are used as inputs to the NN's.

We select one 2-gram amino acid pair as a feature X, let x be its value and demonstrate the possibility of each feature in the positive class and the negative class by employing a distance measure. Let P(x|Class=1) and P(x|Class=0) denote the class conditional density functions for feature X, where Class_1 represents the positive class and Class_0 is the negative class. Let D(X) denote the distance function between P(x|Class=1) and P(x|Class=0), defined as

$$D(X) = \int | P(x \mid Class = 1) - P(x \mid Class = 0) | \, dx \qquad (7.1)$$

For two features X Y and, D(X) > D(Y), means that the distance measure prefers feature X to feature Y because it is easier to distinguish between Class_1 and Class_0 by observing feature X than feature Y. That is, X appears often in the positive class and seldom in the negative class or vice versa.

The following equation (7.2) is used to calculate the feature value x of one certain feature X with respect to the sequence S:

$$x = \frac{c}{length(S) - 1} \qquad (7.2)$$

where c denotes the frequency of X in S, and length(S) represents the length of S.

For example, suppose sequence S = AMNTNMNL. Then the value of the feature MN with respect to sequence S is 2/(8-1) = 0.29.

The distance measure between the positive dataset and the negative dataset is define as:

$$D(X) = \frac{(m_1 - m_0)^2}{d_1^{\,2} + d_0^{\,2}} \tag{7.3}$$

where m_1 and d_1 (m_0 and d_0, respectively) are the mean value and the standard deviation of the feature X in the positive (negative, respectively) training data set. Intuitively, in Equation 7.3, the larger the numerator is (or the smaller the denominator is), the larger the interclass distance is, and therefore the easier it is to separate Class_1 from Class_0 (and vice versa). The mean value m and the standard deviation d of the feature X in a set Q of sequences are defined as:

$$m = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{7.4}$$

$$d = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - m)^2} \tag{7.5}$$

where x_i is the value of the feature X with respect to sequence $S_i \in Q$, and N is the total number of sequence in Q.

Both the 2-gram amino acid encoding scheme method and the 2-gram exchange group encoding scheme method are applied to represent the sequence. Thus, there are $20^2 + 6^2 = 436$ possible 2-grams in total. If all the 436 2-grams are chosen as the neural network input features, it would require many weight parameters and training data. This makes it difficult to train the neural network – a phenomenon called "curse of dimensionality". As a result, we have to reduce the numbers of input features. We let $X_1, X_2, \ldots, X_{Ng}$, $N_g << 436$, be the top N_g features (2-grams) with the largest D(X) values. Intuitively, these N_g features occur more frequently in the positive training data set and less frequently in the negative training data set. For each protein sequence S (whether it is a training or an unlabeled test sequence), we examine the N_g features in S, calculate their values as defined in Equation 7.2, and use the N_g feature values as input feature values to the neural network for the sequence S.

7.2 Proposed NN architectures

7.2.1 Pair-wise Multiple Classification Approach

Pair-wise Multiple Classification NN consists of a combination of N binary NNs, each binary NN performing a binary classification. Here, N is the number of families. If we

number our protein families as F1, F2, and F3 then network NN1 classifies F1 and F2, network NN2 classifies F1 and F3, and network NN3 classifies F2 and F3. The inputs to a given NN classifier are the top N_g 2-gram features of its two associated protein families. The individual NN classifiers are simple back-propagation networks. The output layer has two neurons. There are only 3 possible encodings for the outputs. For instance with classifier NN1 the output codes are 00 (the input sequence is not from either set F1 or F2), 01 (the input sequence is from F1) and 10 (the sequence is from F2). Each NN classifier is trained separately and the outputs are combined and correctly interpreted using win-over method on six output neurons.

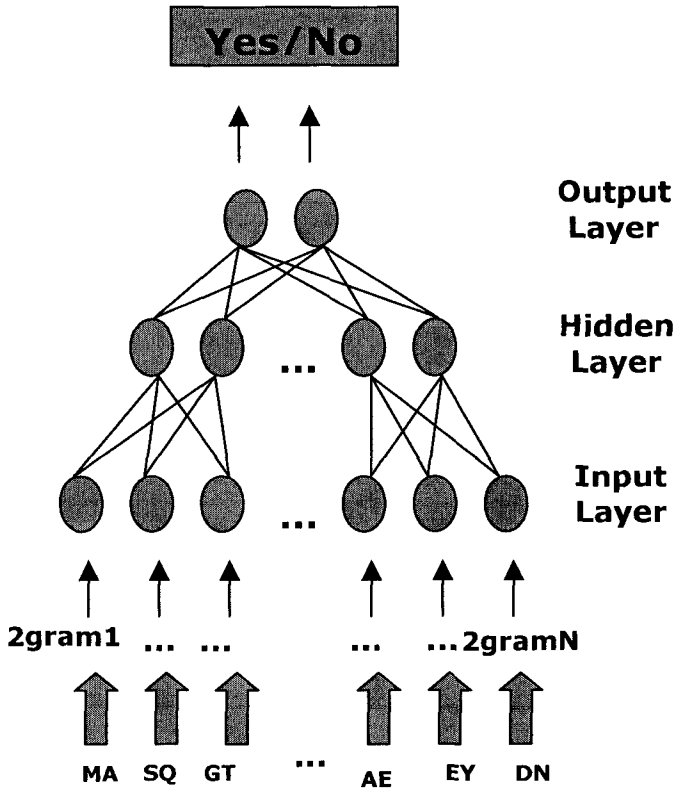


Figure 7.1 *The Architecture of a typical binary NN*

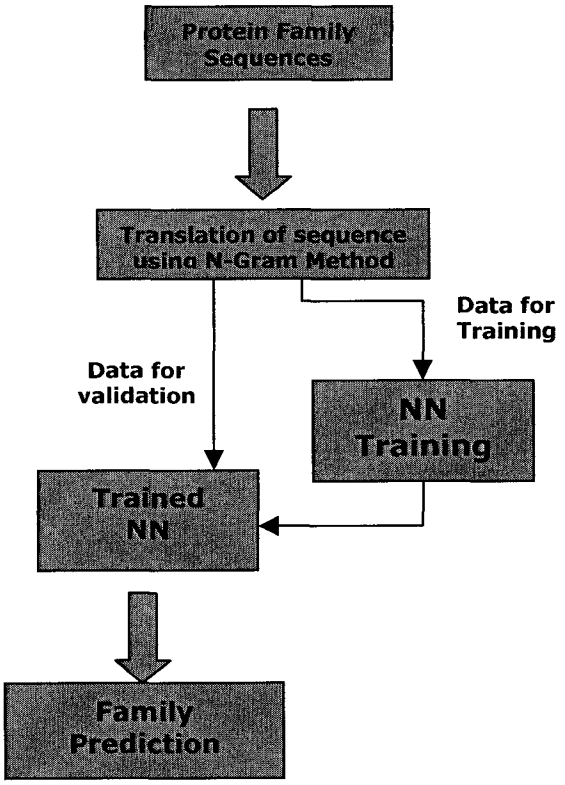


Figure 7.2 Flow chart of Binary Classification

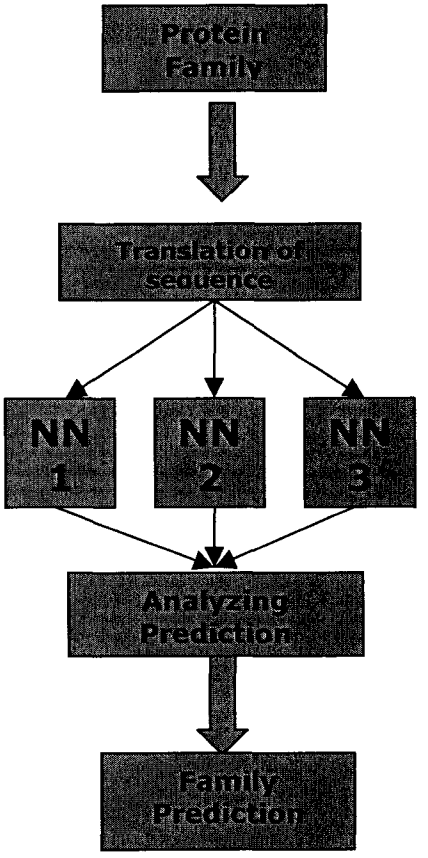


Figure 7.3 Flow chart of Pair-wise Multiple Classification Approach

In Figure 7.1 a typical binary NN classification is schematized while the flow chart of the typical binary NN classification and pair-wise multiple classification method is shown in Figure 7.2 and Figure 7.3 respectively.

7.2.2 Single Network Approach

Unlike the Pair-wise NN, the Single NN is a single compact network. Another difference with the Pair-wise NN is that we have 3 possible output layer definitions for the compact NN. Given ρ protein families $F_0, \ldots, F_\rho$, the output layer is defined as follows, either

1) It contains ρ neurons $N_0, N_1, N_2, \ldots N_\rho$, each neuron N_i associated with family F_i. The target values of a protein of class F_i is defined as follows: $(t_0, t_1, \ldots, t_\rho)$ where $t_i = 1$ and $t_{j\neq i} = 0$. That is, if a protein is correctly classified into F_i, therefore the network outputs 1 at N_i and 0 at $N_{j\neq i}$.

2) It contains $\log_2 \rho$ neurons. For instance, if $\rho = 8$ then there are 3 output neurons. Given 3 output neurons then there are 8 possible output values (N_0, N_1, N_2) each corresponding to a class. For instance value (0, 0, 0) corresponds to class F_0, value (1, 0, 1) to F_5 (the decimal value of the output specifies the class).

3) It contains any number of neurons, and the target values are decided arbitiarily. The output are error-correcting output codes (Dietterich, T. & Bakiri G., 1995).

All our compact NNs are three-layers back-propagation networks. The training set contain protein from all famalies, unlike in the pair-wise NN where an individual NN receives only proteins from 2 families only. The NN architecture of single network method is shown in Figure 7.4.

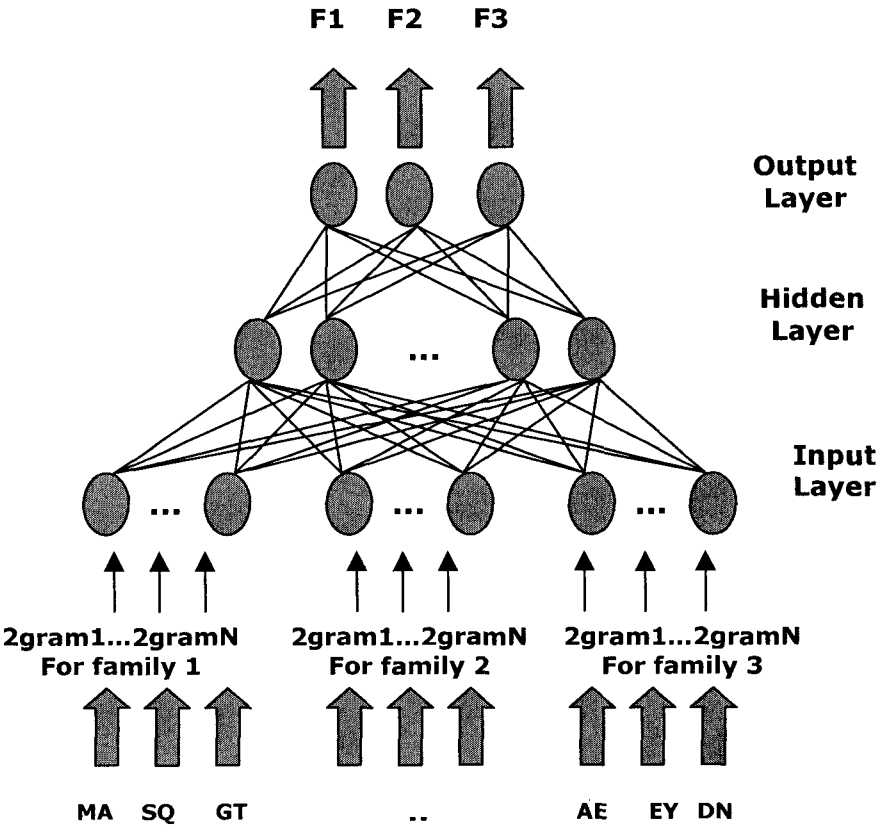


Figure 7.4 Single Network Approach NN Architecture

7.3 New Distance Measurement Function For Multiple Classification

In this thesis, we try to generalize the distance function that can be applied directly to three protein families. It is derived from the Fisher Classification.

The distance measure between three protein family datasets is defined as:

$$D(X) = \frac{\sum_{i=1}^{3}(m - m_i)^2}{d_1^{\,2} + d_2^{\,2} + d_3^{\,2}} \tag{7.6}$$

where m_i and d_i are the mean value and the standard deviation of the feature X in the whole training data set. The mean value m and the standard deviation d of the feature X in a set Q which contains families Q_1, Q_2 and Q_3 of protein are defined as:

$$m = \frac{1}{N}\sum_{i=1}^{3} n_i m_i \tag{7.7}$$

$$m_i = \frac{1}{n_i}\sum_{i=1}^{n_i} x_i \tag{7.8}$$

$$d_i = \sqrt{\frac{1}{n_i - 1}\sum_{i=1}^{n_i}(x_i - m_i)^2} \tag{7.8}$$

where x_i is the value of the feature X with respect to sequence $S_i \in Q$, and N is the total number of sequence in Q, n_i is the total number of sequence in Q_i.

7.4 Using Consensus Sequence as Inputs to The NNs

A consensus sequence is obtained by picking the most frequent base at each position in a set of aligned DNA, RNA or protein sequences. A known conserved sequence set can be

represented by a consensus sequence. Consensus sequence is a method of representing a family.

Here, we propose a new method that uses consensus sequences as inputs to the neural networks rather than the original protein sequences. The training dataset can be randomly divided into n-groups, and then do multiple alignment on each group. Finally, the consensus sequences representing each group are obtained and used as input sequence.

The reason of using consensus sequences as inputs is that it minimizes the learning time. In the normal method, each original protein sequence has to be encoded and is used as input for training the NN, the whole process is time-consuming. In the consensus method, only few consensus sequences representing the family are encoded, consequently, the smaller training dataset is needed to train for the NN, which minimizes the training time considerably.

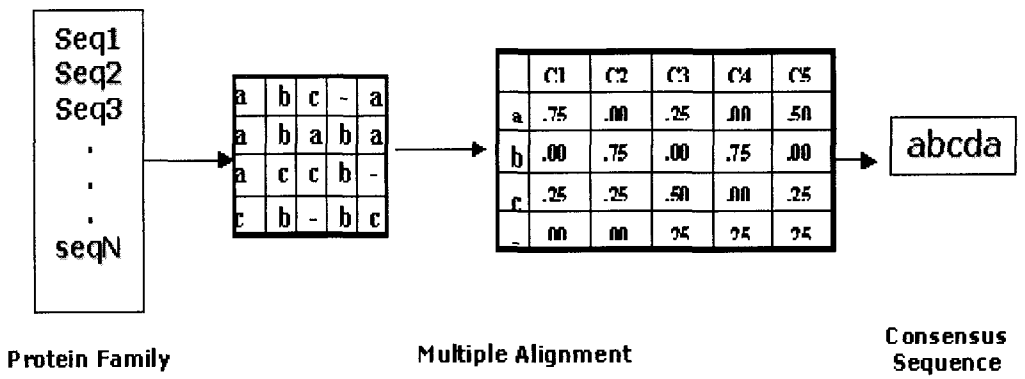


Figure 7.5 Consensus Method

8. Experiments and Results

Preliminary experiments were carried out to investigate the performance of Pair-wise Multiple Classification NN and Single Network NNs with different input encoding schemes such as BIN21, REAL21 and N-gram method. Furthermore, we also tested on different N-gram methods such as 2-gram, 3-gram and 4-gram. We also tested our generalized feature extraction method and our consensus sequence method. All experiments were conducted consistently in the same environment using the same dataset.

8.1 Experimental Environment

Our experiments were carried out using Stuttgart Neural Network Simulator (SNNS) which can be freely downloaded from the website of The Institute for Parallel and Distributed High Performance Systems (IPVR) at the University of Stuttgart (http://www-ra.informatik.uni-tuebingen.de/SNNS/) and "UOWNNS", a simulator developed by the author as a comparison.

SNNS is a software simulator for neural networks which has good performance on Unix or Windows operation system. The kernel of the SNNS is written in C. It supports arbitrary network topologies and, like Rochester Connectionist Simulator (RCS), it has a graphical user interface under X11R4 or X11R5. The SNNS includes lots of network architecture and learning procedures compared to other software in same category.

"UOWNNS" is an Artificial Neural Network Simulator (NN) written by java programming language, and which I developed for testing experiment on protein family classification. It consists of two algorithms: Back Propagation and Self Organizing Map. Some other algorithms will be added for future work. It is a beta version program now.

The hardware configuration is as follows:

- Intel Pentium III Workstation at 794 MHz;
- 384 MB RAM;
- 20GB hard drive;
- Windows XP operating system.

8.2 Dataset

The data used in the experiments were obtained from the International Protein Sequence Database in the Protein Information Resource (PIR) maintained by the National Biomedical Research Foundation (NBRF-PIR) at the Georgetown University Medical Center. This database can be accessed at http://pir.georgetown.edu. Table 8.1 summarizes the data used in the experiments. In theory, the percentage of training dataset and testing dataset can be different value. Generally, 60% sequences of dataset are used for training dataset and 40% are used for testing dataset. Different percentage will affect the performance of the NN, which will be discussed in the section 8.3.3.

Table 8.1 Protein family datasets

Family Name	Number	Training Set (60%)	Testing Set (40%)
Globin	1999	1199	800
Kinase	608	364	344
Ras	530	318	212

Globin: Globin protein family.

Kinase: Kinase-related transformation protein family.

Ras: Ras transformation protein family.

8.3 Experimental Results

The following tables and figures containing summaries of the results of the experiments which are tested with respect to different representations and feature extractions. To remind the reader what each term in these tables and figures stand for, we summarize them as follows:

- GRAM:

n-gram encoding scheme method such as 2-grams, 3-grams

- TYPE:

Type I means that Globin protein family is used for positive dataset, whereas Kinase protein family is used for negative dataset;

Type II means that Globin protein family is used for positive dataset, whereas Ras protein family is used for negative dataset;

Type III means that Kinase protein family is used for positive dataset, whereas Ras protein family is used for negative dataset;

- Accuracy: $Accuracy = \frac{TP + TN}{Total}$;
- Sensitivity: $Sensitivity = \frac{TP}{TP + FN}$;
- Specificity: $Specificity = \frac{TN}{TN + FP}$;
- TSSE: Total sum of square error;
- MSE: Mean square error;
- Training Time: CPU TIME based on Pentium III PC running WINDOWXP operating system.

The architecture of NNs used in all experiments is a three-layer topology:

- Number of input layer neurons: Depends on the representation method;
- Number of output layer neurons: Depends on the representation method;
- Number of hidden layer neurons: 10;
- Learning Algorithm: Std_Backpropagation;
- Learning parameter: epochs: 10000; learning rate η: parameter: 0.8; momentum term μ: 0.2; flat spot elimination value c: 0.1;
- The number of input neurons: 60 top N_g features.

8.3.1 Experiments based on the BIN21 and REAL21Encoding Scheme Methods

In these experiments, we used BIN21 and REAL21 encoding schemes and experimented on different sliding window sizes. The architecture of NN is a three-layer back-propagation binary classification neural network. Type I, II, III neural network is employed. Single-one Method is employed for output representation.

The data below shows that: for BIN21 and REAL21 encoding scheme, with the increase of length of the slide window, the accuracy of NN also increases, whereas the training time also increases because of the increase of input neurons for NN; to compare the accuracy the different encoding scheme for Type III NN, 2-Gram encoding scheme has the best performance.

Table 8.2 *Experiments based on the BIN21 Encoding Scheme Method*

Length	Type	Accuracy	Sensitivity	Specificity	TSSE	MSE	Training time (s)
7	I	71. 02%	70. 02%	73. 85%	401.11	0. 9453	4982
	II	71. 90%	70. 99%	74. 32%	399. 69	0. 9187	5049
	II	72. 54%	71. 22%	75. 84%	395. 77	0. 9109	5110
9	I	74. 92%	72. 55%	76. 89%	340. 48	0. 7775	6701
	II	73. 92%	72. 41%	76. 88%	340. 44	0. 8977	6891
	III	75. 44%	74. 05%	77. 45%	349. 54	0. 6821	6923
11	I	79. 60%	77. 45%	81. 08%	312. 46	0. 4993	8543
	II	79. 31%	77. 32%	80. 94%	310. 50	0. 5002	8439
	III	79. 99%	77. 78%	80. 91%	315.66	0. 5108	8442
13	I	83. 61%	81. 11%	86. 59%	240. 98	0. 4213	7902
	II	83. 25%	82. 53%	86. 32%	239. 50	0. 4573	8002
	III	84. 86%	83. 13%	86. 71%	245. 83	0. 4387	8024

Length: The size of sliding window.

Table 8.3 *Experiments based on the REAL21 Encoding Scheme Method*

Length	Type	Accuracy	Sensitivity	Specificity	TSSE	MSE	Training time (s)
7	I	73. 88%	72. 87%	76. 58%	345.98	0. 7842	4567
	II	74. 12%	73. 31%	76. 98%	340. 31	0. 7648	4781
	II	75. 55%	72. 68%	77. 04%	332. 85	0. 7222	3975
9	I	76. 29%	73. 43%	77. 78%	319. 48	0. 7211	5706
	II	76. 24%	73. 89%	78. 84%	324. 41	0. 7251	5893
	III	76. 84%	73. 65%	79. 69%	315. 74	0. 7119	4975
11	I	76. 86%	75. 45%	80. 10%	315. 99	0. 7116	8901
	II	76. 51%	75. 32%	79. 77%	317. 58	0. 7159	9329
	III	76. 89%	75. 44%	79. 80%	315.96	0. 7108	8651
13	I	77. 99%	75. 17%	80. 59%	302. 98	0. 6983	10962
	II	78. 25%	77. 80%	81. 32%	298. 76	0. 6577	10549
	III	78. 68%	77. 79%	81. 13%	294. 38	0. 6497	9876

Length: The size of sliding window.

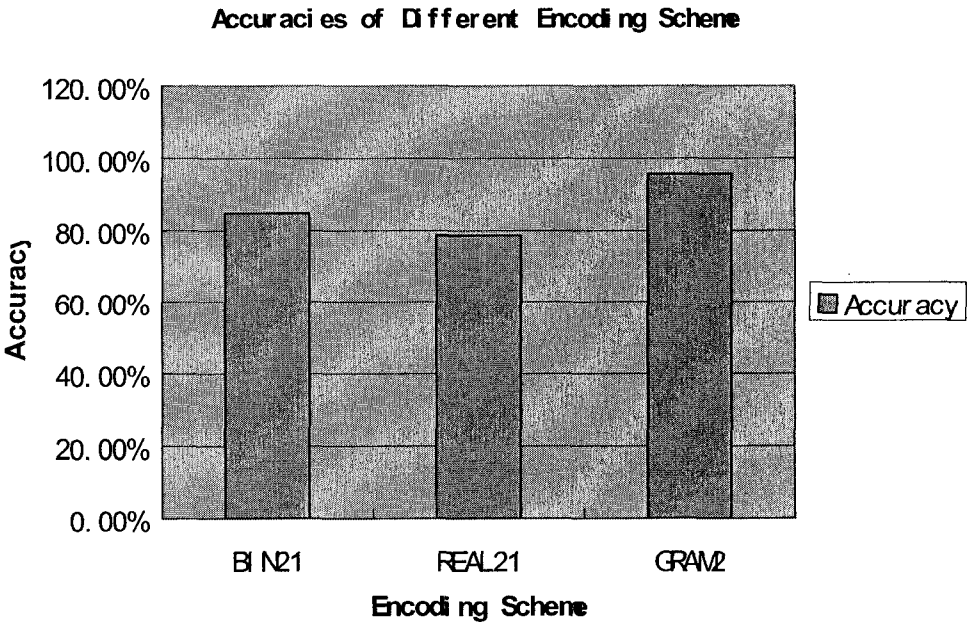


Figure 8.1 Accuracies of Different Encoding Scheme

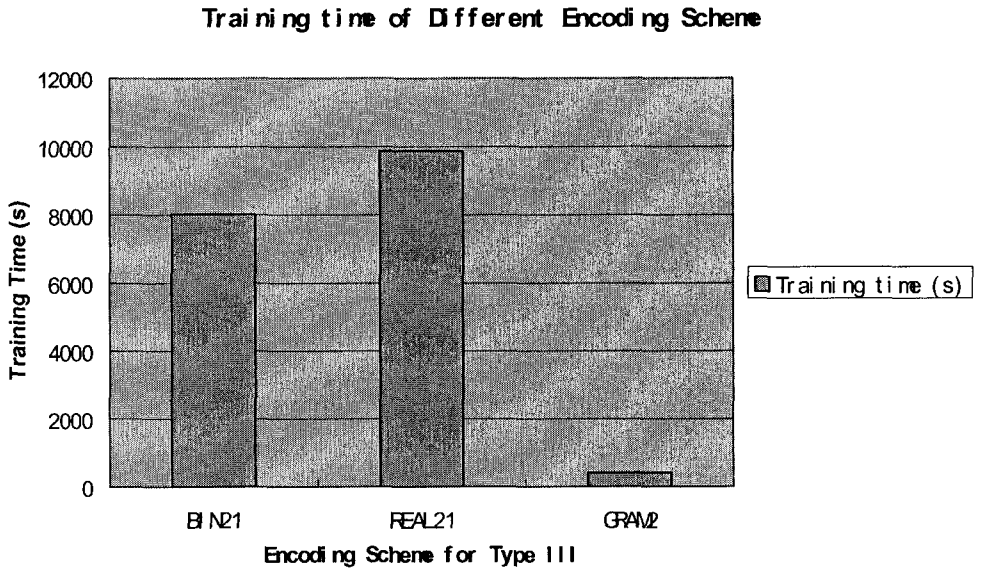


Figure 8.2 Training Time of Different Encoding Scheme for Type III

8.3.2 Experiments on Binary Classification using N-gram Method

In these experiments, we encode the input sequences with 2-gram, 3-gram and 4-gram Encoding Scheme Method and test experiments on the Type I, II, III neural network.

Table 8.4 *Experiments on 2-gram Binary Classification NN*

GRAM	Type	Accuracy	Sensitivity	Specificity	TSSE	MSE	Training time (s)
2	I	94.84%	93.81%	96.58%	30.8921	0.02582	646
	II	94.47%	93.01%	95.61%	31.5538	0.03118	611
	III	95.68%	93.88%	96.77%	14.9845	0.02691	405
3	I	95.88%	94.53%	97.03%	23.8921	0.01591	6710
	II	95.77%	94.45%	97.43%	26.5538	0.04894	7096
	III	96.98%	94.03%	97.66%	19.8601	0.00939	5434
4	I	96.50%	93.99%	97.02%	22.8686	0.01849	10,732
	II	96.85%	94.18%	97.50%	21.8658	0.01284	11,614
	III	96.46%	94.11%	97.09%	22.9658	0.02835	9,045

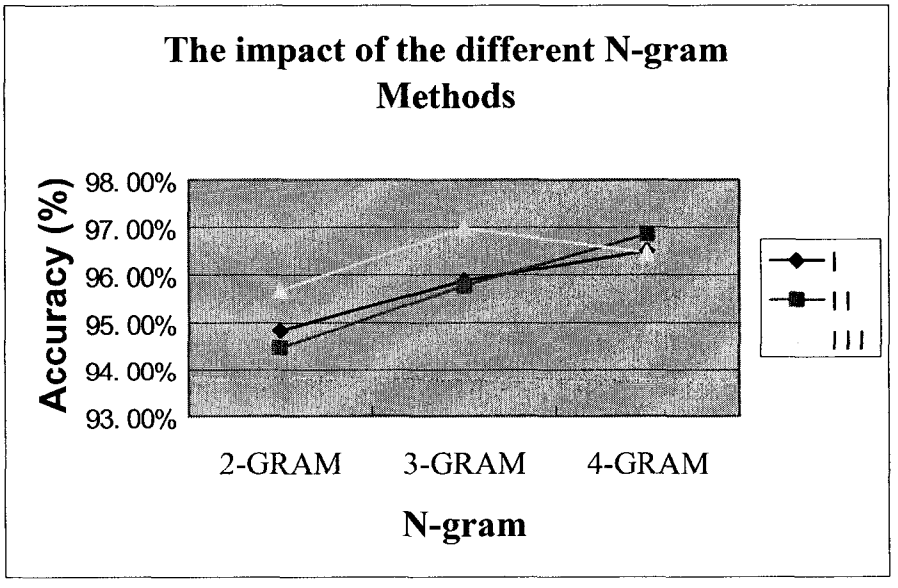


Figure 8.3 *The impact of the different N-gram Methods*

It is illustrated from Table 8.4 and Figure 8.3 that although 4-GRAM encoding scheme performs the best, its training time is the longer than 2-GRAM and 2-GRAM encoding scheme.

8.3.2 Experiments on using consensus sequences as training dataset

Rather than applying the original training dataset to train NN, we convert protein sequences in the training dataset into consensus sequences after performing multiple alignments. 2-GRAM encoding scheme is applied to these consensus sequences and used for input to binary classification NN.

Table 8.5 *Experiments on using consensus sequences as training dataset*

	Type	Accuracy	Sensitivity	Specificity	TSSE	MSE	Training time (s)	Multiple alignment time (s)
Gram2	III	95.68%	94.32%	96.89%	14.9845	0.0269054	405	0
Group5	III	91.77%	89.31%	93.54%	80.2754	0.1443802	240	72
Group10	III	81.82%	79.99%	83.84%	289.478	0.5206433	40	108
Group15	III	73.87%	69.05%	76.80%	450.384	0.7999709	21	160

Note: GroupN signifies that the consensus sequences are obtained from multiple alignments of N sequences. For instance, if the training set has 100 sequences, and if N = 20, we here randomly make 5 groups of 20 sequences, and multiply align each group to obtain 5 consensus sequences that will be used as input to the NN.

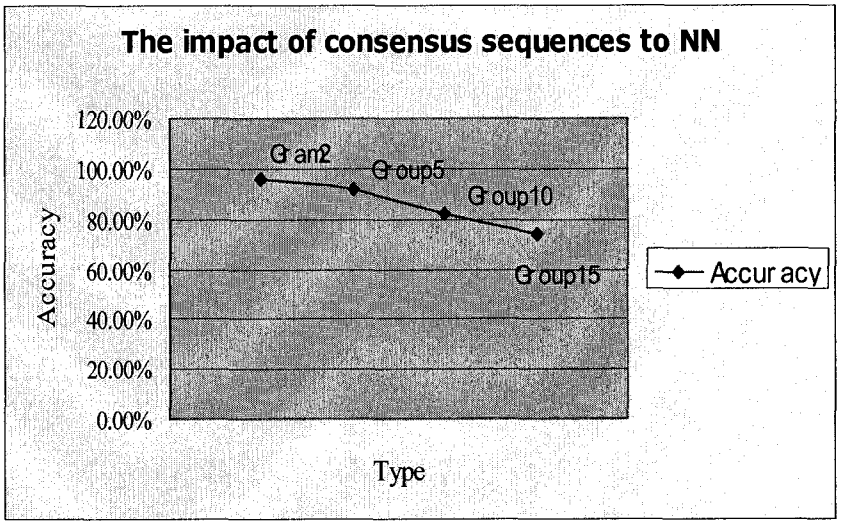


Figure 8.4 *The impact of consensus sequences on NNs' accuracy*

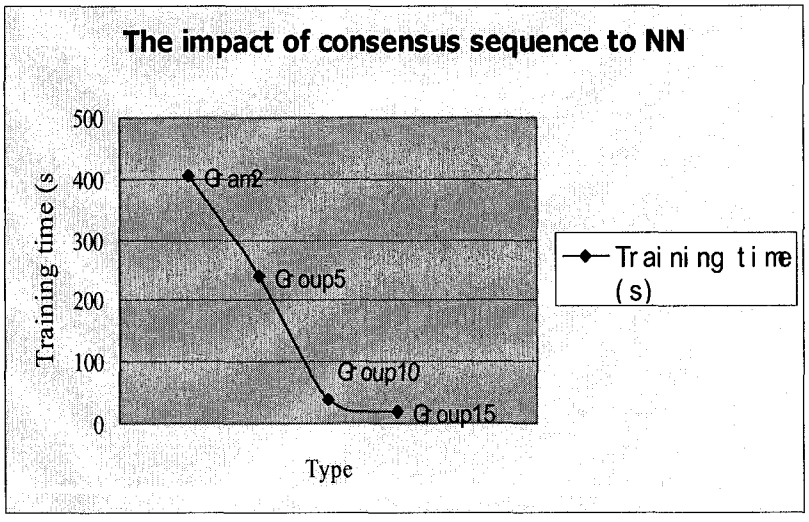


Figure 8.5 *The impact of consensus sequence on NNs' training time*

The figure above shows that: with more sequences in each multiple alignment, the training time is reduced greatly whereas the accuracy drops quickly due to loss of information during consensus sequence generation.

8.3.3 Experiments on Single Network Approach

We test experiments using Single Network Approach based on 2-gram Encoding Scheme. There are total 180 neurons in the input layer. The input to the networks are sequences drawn from the 3 protein families.

Table 8.6 *Experimental results using Single Network Approach*

GRAM	Numbers of output node	Accuracy	Sensitivity	Specificity	TSSE	MSE	Training time (s)
2	3	43.95%	40.64%	50.67%	1208.38	0.8911357	1235
	4	58.90%	55.90%	60.44%	1000.87	0.7381047	1789
	5	60.85%	57.88%	65.31%	920.61	0.6789159	2673
	6	79.49%	72.08%	82.22%	314.98	0.2322861	3599
	7	73.93%	70.13%	75.70%	503.73	0.3714823	4688
	8	74.93%	70.23%	76.59%	505.88	0.3730678	5053

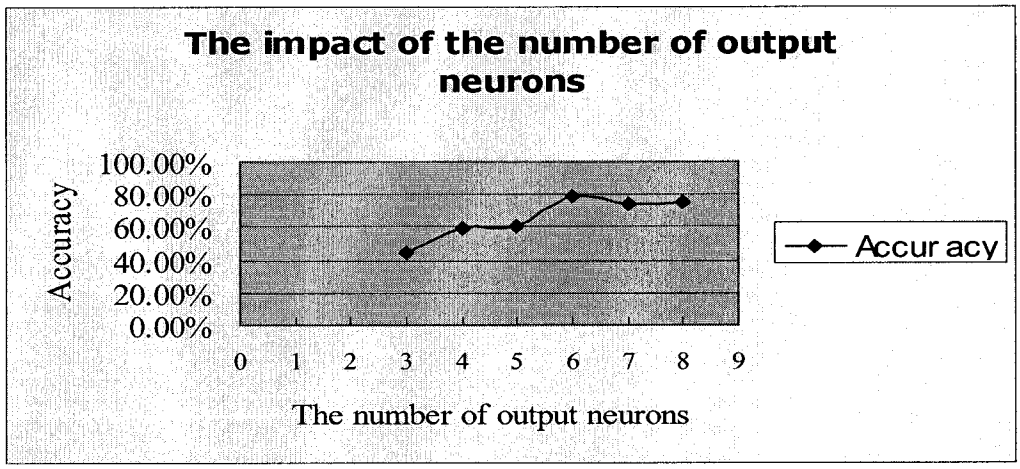


Figure 8.6 *The impact of the number of output neurons on accuracy*

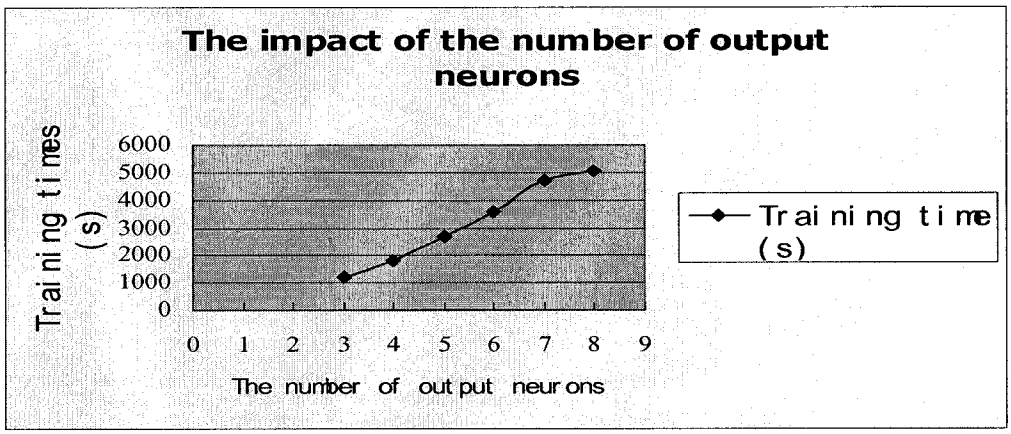


Figure 8.7 *The impact of the number of output neurons on training time*

In the experiments above, we show the performance of the network with respect to the number of output neurons. Here, for a number Q of output neurons, we arbitrarily/randomly assign an output-code for a family. The experiments above show that:

when the number of output neurons is 6, we get the highest accuracy. Also training time increases as the number of output neurons increases.

8.3.4 Experiments on Pair-wise Multiple Classification Approach with different percentage of training dataset and testing dataset

The NN may not be able to accurately predict new cases if the amount of data used for training is too small. At the same time, the quality assessment may not be accurate if the portion of data used for testing is too small. So, experiments are done on different percentages of training dataset and testing dataset.

Each sub-network in the Pair-wise Multiple Classification Approach is a typical Binary Classification NN, with 60 input neurons. The inputs are 2-grams, and we use single-one encoding for the 2 output neurons.

Table 8.7 and Figure 8.8 below show that Pair-wise Multiple Classification Approach yields excellent results for classifying three protein families. Best results are obtained when the training set has size between 60% and 70%, which confirms some investigations that suggest a split between 25%-50% of the available data as an optimal partition for testing (Efron & Gong, 1983).

Table 8.7 *Experiments on Pair-wise Multiple Classification Approach with different percentage of training dataset and testing dataset*

NN	GRAM	Percentage of Training and Testing Dataset	Accuracy	Sensitivity	Specificity	TSSE	MSE	Training time(s)
Overall	2	50% vs 50%	**89.48%**	**85.98%**	**92.53%**	**203.721**	**0.1642**	866
NN1			90.49%	88.86%	92.95%	199.584	0.1614	
NN2			89.25%	87.57%	91.49%	204.543	0.1655	
NN3			89.42%	88.05%	92.03%	204.002	0.1650	
Overall	2	60% vs 40%	**94.10%**	**92.84%**	**97.03%**	**58.65**	**0.0433**	991
NN1			93.57%	91.88%	96.77%	64.77	0.0478	
NN2			94.58%	92.49%	96.99%	54.61	0.0403	
NN3			94.22%	92.69%	96.48%	57.69	0.0425	
Overall	2	70% vs 30%	**93.47%**	**91.99%**	**96.87%**	**75.311**	**0.0555**	1312
NN1			93.49%	92.00%	95.69%	75.318	0.0555	
NN2			93.39%	91.89%	96.29%	75.677	0.0558	
NN3			93.50%	91.88%	96.49%	74.45	0.0549	
Overall	2	80% vs 20%	**92.35%**	**90.06%**	**94.55%**	**88.96**	**0.0656**	1441
NN1			92.45%	90.06%	94.69%	87.56	0.0646	
NN2			92.55%	90.10%	94.92%	89.34	0.0659	
NN3			92.35%	90.07%	94.38%	88.98	0.0656	
Overall	2	90% vs 10%	**91.66%**	**89.66%**	**93.88%**	**144.96**	**0.1494**	1554
NN1			91.56%	89.66%	92.87%	146.99	0.1399	
NN2			92.75%	90.87%	93.94%	177.68	0.1330	
NN3			91.67%	90.82%	93.33%	143.96	0.1377	

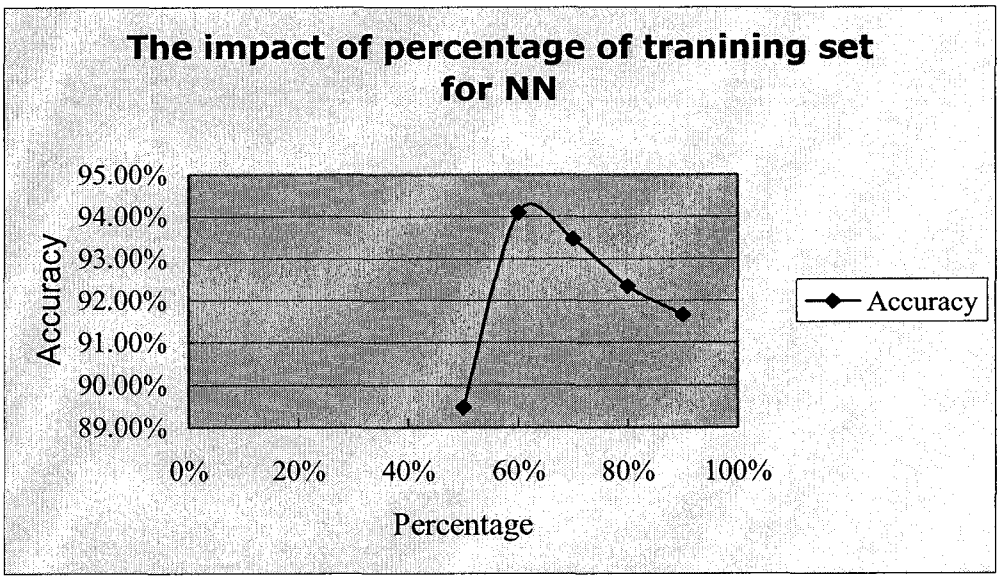


Figure 8.8 *The impact of percentage of training dataset and testing dataset for NN*

8.3.5 Experiments on the number of best features Pair-wise Multiple Classification Method

Table 8.8 *Experiments on picking different numbers of top Ng features*

GRAM	Numbers of Top Ng Features	Accuracy	Sensitivity	Specificity	TSSE	MSE	Training time(s)
2	20	89.59%	87.49%	92.58%	197.535	0.14569	1002
	30	91.99%	90.32%	94.84%	89.663	0.06612	944
	40	92.85%	92.33%	96.04%	80.779	0.05957	953
	50	93.34%	92.98%	96.89%	78.217	0.05768	1011
	60	**94.10%**	**92.84%**	**97.03%**	**58.65**	**0.0433**	**1411**
	70	93.28%	93.01%	96.99%	78.981	0.05824	982
	80	92.41%	92.11%	95.97%	81.534	0.06012	1018
	90	91.83%	90.24%	94.08%	90.537	0.06677	996
	100	90.64%	88.54%	91.88%	165.593	0.12212	958

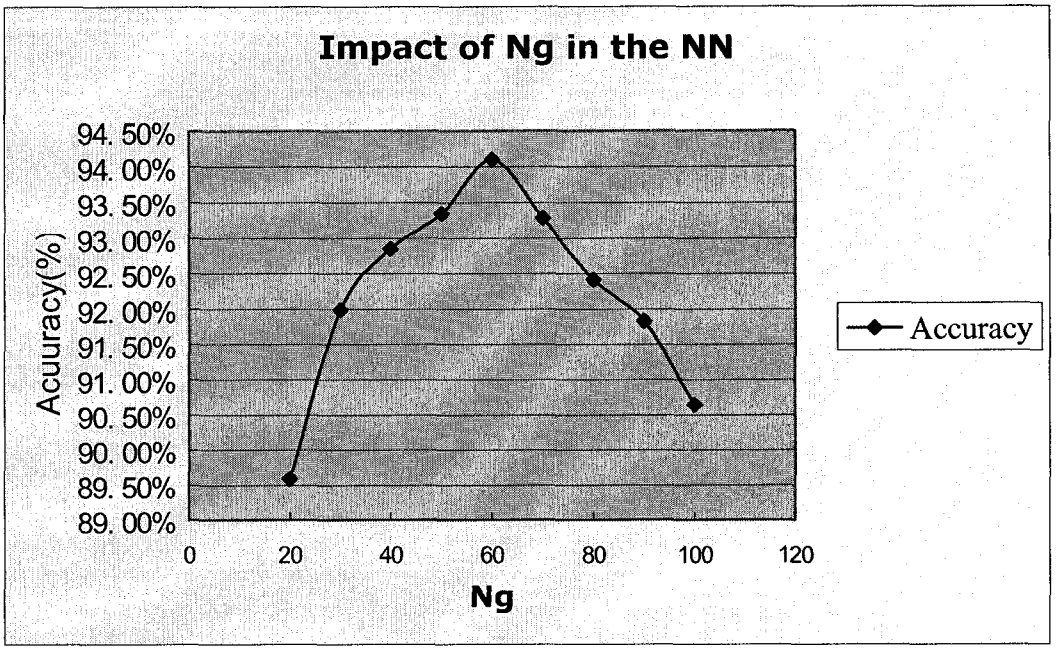


***Figure 8.9** Impact of Ng*

We use the Pair-wise Multiple Classification Approach NN with 60%-40% splitting (training dataset vs. testing dataset) to test the impact of different values of N_g on the NN performance. The data above shows that the NN has the highest accuracy when we N_g = 60 for each sub-network.

8.3.6 Experiments on using New Distance Measurement Function

We have performed experiments on using New Distance Measurement Function (NDMF) for our three-class problem. We applied the NDMF on Single Network NNs with 3 output neurons, each associated with a family. We compare this method with other two multiple classification approaches. The percentage of training dataset and testing dataset is 60%: 40%.

Table 8.9 *Comparison of the performance of three methods for multiple classification*

GRAM	Method	Accuracy	Sensitivity	Specificity	TSSE	MSE	Training time(s)
2	**M1**	**94.10%**	**92.84%**	**97.03%**	**58.65**	**0.0433**	**991**
	M2	79.49%	72.08%	82.22%	314.98	0.2322861	3599
	M3	72.29%	70.32%	75.55%	489.46	0.361	3421
	M4	56.61%	50.99%	67.73%	876.69	0.646526	421
	M5	54.55%	48.93%	59.39%	943.33	0.695671	435

Method1: Pair-wise Multiple Classification Approach;
Method2: Single Network Approach with Single-one output representation;
Method3: Single Network Approach with Logarithmic output representation;
Method4: NDMF applied to Single Network Approach with Single-one output representation;
Method5: NDMF applied to Single Network Approach with Single-one output representation.

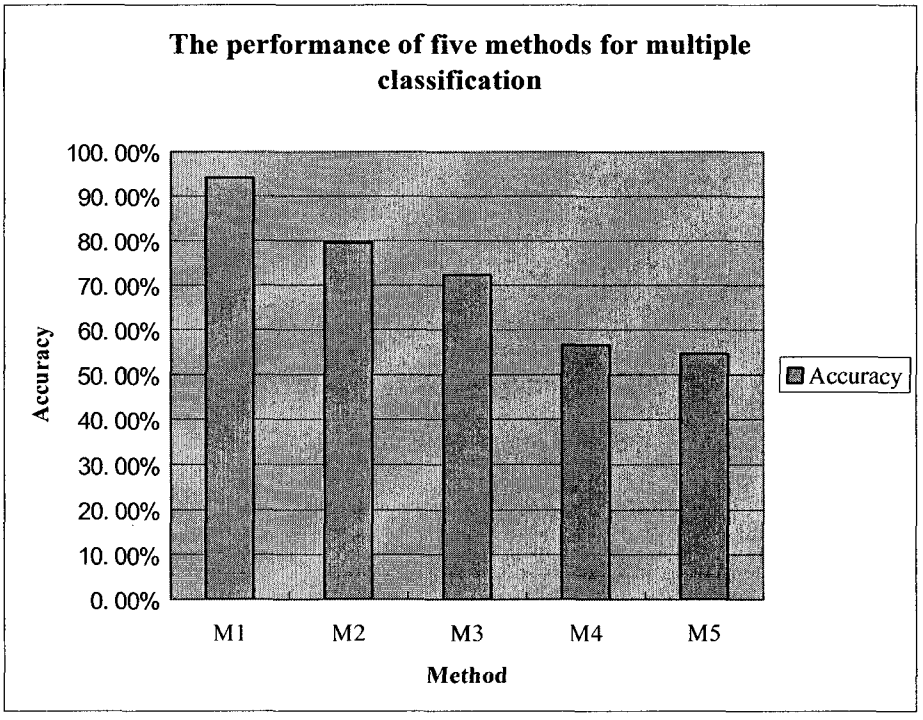


Figure 8.10 *The performance of three methods for multiple classification*

The figure above shows that: among the five methods for multiple classification, Pair-wise Binary Classification performs best; the accuracy of new method derived from Fisher Classification is not good enough although it take less time due to its minimal architecture of NN; the performance of Single Network Approach is between the other two approaches.

8.3.7 Experiments on cross-validation for Pair-wise Multiple Classification Approach

Cross-validation method (Picard & Berk, 1990) is the standard method for evaluating generalization performance with training and test sets. Because we can get the highest accuracy when we divide the dataset into 60% training set and 40% testing set, we do the holdout cross-validation method based on this splitting percentage. We divide the data based on that percentage, and then the process is repeated several times and the classification performance is the average of the individual test estimates.

Table 8.10 *Summary of cross-validation result: Mean + standard error of classification accuracies*

Train/test Runs	P1	P2	P3	P4	P5	P6
10	0.9401±0.0295	0.8993±0.0329	0.8316±0.0347	0.7743±0.0358	0.7843±0.0400	0.7666±0.0380
25	0.9118±0.0184	0.8843±0.0235	0.8231±0.0284	0.7532±0.0243	0.7730±0.0230	0.7601±0.0222
50	0.8917±0.0125	0.8802±0.0102	0.8196±0.0143	0.7544±0.0121	0.7669±0.0126	0.7532±0.0128
100	0.9099±0.0049	0.8834±0.0035	0.8199±0.0061	0.7561±0.0023	0.7678±0.0021	0.7533±0.0041

P1: Pair-wise Multiclass NN using 2-GRAM encoding scheme and Single-one output representation;
P2: Pair-wise Multiclass NN using 2-GRAM encoding scheme and Logarithmic output representation;
P3: Pair-wise Multiclass NN using BIN21 encoding scheme and Single-one output representation;
P4: Pair-wise Multiclass NN using BIN21 encoding scheme and Logarithmic output representation;

P5: Pair-wise Multiclass NN using REAL21 encoding scheme and Single-one output representation;
P6: Pair-wise Multiclass NN using REAL21 encoding scheme and Logarithmic output representation.

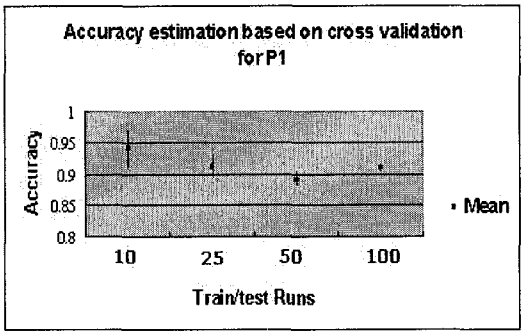


Figure 8.11 *Accuracy estimation based on cross-validation for P1*

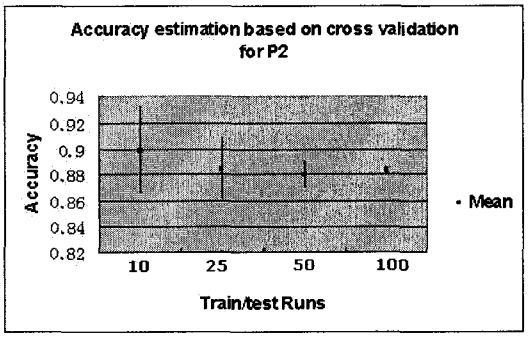


Figure 8.12 *Accuracy estimation based on cross-validation for P2*

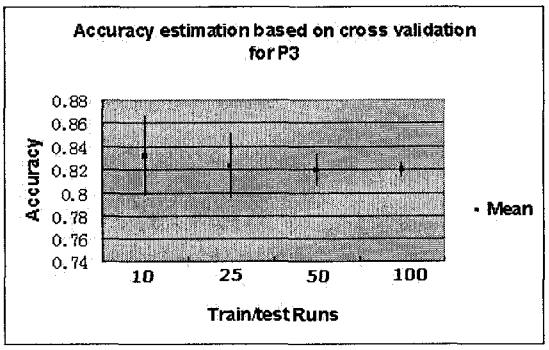


Figure 8.13 *Accuracy estimation based on cross-validation for P3*

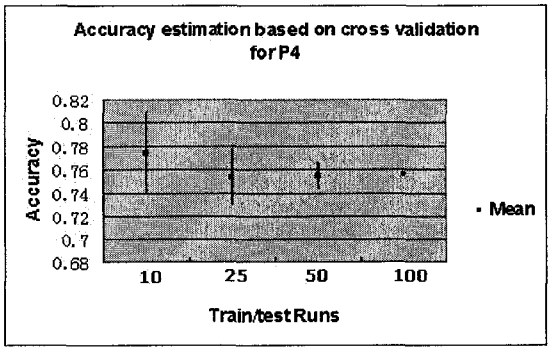


Figure 8.14 *Accuracy estimation based on cross-validation for P4*

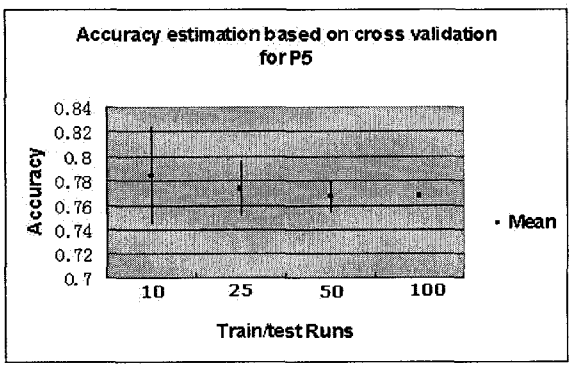


Figure 8.15 *Accuracy estimation based on cross-validation for P5*

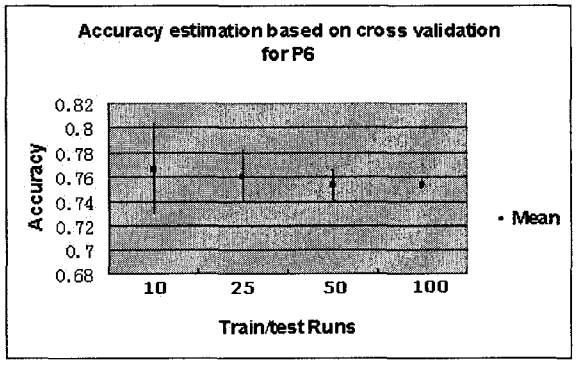


Figure 8.16 *Accuracy estimation based on cross-validation for P6*

8.3.8 Summary of Experiments on Different Input Sequence and Output Representations

We tested the impact of different input sequence encoding schemes and output representations based on the performance of the Pair-wise Multiclass NN and Single Network NN. The size of the sliding window is 13 when the BIN21 and REAL21 encoding scheme are applied. We compare these approaches based on that the percentage of training dataset and testing dataset is 60%: 40%.

Table 8.11 *Summary of experiments on different input sequence and output representations*

NN	**Pair-wise Multiclass NN**						**Single Network NN**					
Input Representation	BIN21		REAL21		2-GRAM		BIN21		REAL21		2-GRAM	
Output Representation	Single-one	Logarithmic	Single-one	Logarithmic	Single-one	Logarithmic	Single-one	Logarithmic	Single-one	Logarithmic	Single-one	Logarithmic
Accuracy	83.16%	77.43%	78.43%	76.66%	**94.10%**	89.93%	60.02%	56.43%	59.98%	57.43%	79.49%	72.29%
Sensitivity	80.31%	79.93%	79.35%	77.32%	**92.84%**	84.67%	63.56%	61.22%	63.44%	59.39%	72.08%	70.32%
Specificity	85.35%	75.64%	76.49%	75.03%	**97.03%**	91.44%	58.88%	54.44%	58.34%	55.39%	82.22%	75.55%
TSSE	241.77	356.39	296.43	360.55	**58.65**	169.34	640.05	846.98	688.77	730.54	314.98	489.46
MSE	0.1783	0.2628	0.2186	0.2659	**0.0433**	0.1249	0.472	0.6246	0.5079	0.5387	0.232286	0.361
Training time(s)	24,024	20,032	31,647	30,003	**991**	983	15,346	14,930	14,893	14,238	3,599	3,421

9. Conclusion

9.1 Summary of Work Done

Neural network techniques applied in the classification of protein family involves several aspects: input and output representation, feature extraction, architectures of NN, network learning algorithm, training and testing dataset compilation and evaluation mechanism. Three protein families are employed as datasets in this research, Globin protein family, Kinase-related transformation protein family and Ras transformation protein family. The examination of different sequence representations, feature extractions and architectures of NN are conducted for two categories: Binary Classification and Multiple Classification. Furthermore, based on the n-gram Encoding Scheme proposed by Wu (1992) and feature extraction by Wang (2002), we construct two new NN architectures to classify protein into one among many families. Experiments were carried out and results were analyzed.

The experiments illustrate that Pair-wise Multiple Classification NN Approach gives the highest accuracy than Single Network Approach. Meanwhile, N-gram Encoding Scheme performs best among different encoding schemes and feature extraction.

There are many factors involved in the classification of protein families using neural networks issue. Though what we have shown are very crude experiments, they are sufficient to indicate that neural network application in protein family classification is a good area for further study.

9.2 Limitations and Future Work

There are a number of limitations that need further improvement.

- Training time

In the Pair-wise Binary Classification Approach, three sub-networks are trained separately, the process of training the whole neural network is time-consuming. Furthermore, in the Single Network Approach, the top Ng features of all three families are used as the input neurons, as a result, the training process is also time-consuming.

- Feature extraction

The calculation of distance function, D(X), between positive dataset and negative dataset is time-consuming when there are a lot of sequences in one protein family and each sequence's length is very long.

- Consensus Method for classification

The purpose of using consensus sequences rather than original sequences as the input sequence is to reduce training time and computation of the D(X) values. In our experiment, it is illustrated that it indeed can reduce the time but it also reduces the accuracy for larger multiple alignments.

- Crude experiment

We need to do test with larger number of families, and also design more robust and powerful input/output representation.

These limitations lead to some directions for future work. We summarize as follows:

- Parallel method to training individual neural network in the Pair-wise Multiple Classification Approach.
- Parallel method to calculate the D(X) values.
- Adding some other protein sequence information such as structural information to compensate the lost local similarity information in the sequences when consensus sequences are used as inputs.
- More experiments on combinations of different input/output encoding method, feature extraction methods and learning algorithms.
- Optimize the output representation using Error-Correcting Output Codes Method.
- Experiments on more than three protein families.
- Employing Fisher Linear Discriminant Analysis method (FLDA).
- Employing Support Vector Machines (SVM) to solve protein family multi-classification problem.

9.3 Conclusions

In this thesis, the neural networks are applied to classify multiple protein families. The main contributions of our work include:

- Two different NN methods: Pair-wise Multiple Classification NN and Single NN.
- The development of a neural network simulator using Java programming language. Two learning algorithms are implemented: back-propagation and self-organizing map.
- Generalized Fisher Method for 3-class.
- Our experimental results indicate that the Pair-wise Multiple Classification Approach and Single Network Approach achieve good performance on three protein superfamilies from PIR protein database with accuracy of 94.10% and 79.49% respectively.

Bibliography

1. Alberts B., Bray D., Lewis J., Raff M., Roberts K., Watson J.D. (1994), *Molecular biology of the cell,* Chapter 3, 112-113, Garland Publishing, Inc., New York.

2. Altschul, S. F., Gish, W., Miler, W., Myers, E. W. & Lipman, D. J. (1990). *"Basic local alignment search tool"*, J Mol Biol, 215, 403-410.

3. Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z, Miller, W. & Lipman, D. J. (1997). *"Gapped BLAST and PSI-BLAST: a new generation of protein database search programs"*, Nucletic Acids Res, 25, 3389-3402.

4. Arrigo, P., Giuliano, F., Scalia, F., Rapallo, A. & Damiani, G. (1991). *"Identification of a new motif on nucleic acid sequence data using Kohonon's self organizing map"*, Comput Appl Biosci, 7, 353-357.

5. Attwood, T. K., Flower, D. R., Lewis, A. P., Mabey, J. E., Morgan, S. R., Scordis, P., Selley, J. N. & Wright, W. (1999). *"PRINTS prepares for the new millennium"*, Nucleic Acid Res, 27, 220-225.

6. Barker, W. C., Garavelli, J. S., McGarvey, P. B., Marzec, C. R., Orcutt, B. C., Srinivasaro, G. Y., Yeh, L. S., Ledley, R. S., Mewes, H. W., Pfeiffer, F., Tsugita, A. & Wu C. (1999). *"The PIR-Ineternational Protein Sequence Database"*, Nucleic Acids Res, 27, 39-43.

7. Bateman, A., Birney, E., Durbin, R., Eddy, S. R., Finn, R. D. & Sonnhammer, E. L. (1999). *"Pfam 3.1: 1313 multiple alignments and profile HMMs match the majority of proteins"*, Nucleic Acids Res, 27, 260-262.

8. Beale, R. & Jackson, T. (1991). Neural Computing: An Introduction, Adam Hilger, Bristol.

9. Blom, N., Hansen, J., Blaas, D. & Brunak, S. (1996). *"Cleavage site analysis in picornaviral polyproteins: discovering cellular targets by neural networks"*, Protein Sci, 5, 2203-2216.

10. Bucher, P. (1990). *"Weight matrix descriptions of four eukaryotic RNA polmerase II promoter elements derived from 502 unrelated promoter sequences"*, J Mol Biol, 220, 49-65.

11. Burge, C. & Karlin, S. (1997). *"Prediction of complete gene structures in human genomic DNA"*, J Mol Biol, 268, 78-94.

12. Bult, C.J., White, O., Olsen, G.J., Zhou, L., Fleischmann, R.D., Sutton, G.G., Blake, J.A.,FitzGerald, L.M., Clayton, R.A., Gocayne, J.D., Kerlavage, A.R., Dougherty, B.A., Tomb, J.F., Adams, M.D., Reich, C.I., Overbeek, R., Kirkness, E.F., Weinstock, K.G., Merrick, J.M., Glodek, A., Scott, J.L., Geoghagen, N.S. and Venter, J.C. (1996). "*Complete genome sequence of the methanogenic archaeon, Methoanococcus jannaschii*", Science, 273, 1058-1073.

13. Campitelli, L., Delledonne, L. & Salvini, A. (1999). *"A neural network approach to protein sequence processing"*, Euro Biosci, 6, 78-79.

14. Che A. (2000). *"Classification of transmembrane helices and signal peptides using neural networks"*, Comput Appl Biosci, 10, 63-66.

15. Cheng, B. & Titterington, D. M. (1994). *"Neural Networks: a review from a statistical perspective"*, Statistical Sciences, 9, 2-54.

16. Corpet, F., Gouzy, J. & Kahn D. (1999). *"Recent improvements of the ProDom database of protein domain families"*, Nucleic Acids Res, 27, 263-267.

17. Day, W. H. E. & McMorris, F. R. (1993). *"A consensus program for molecular sequences"*, Comput Appl Biosci, 9, 653-656.

18. Dayhoff, M. O., Schwartz, R. M. & Orcutt, B. C. (1978). *"A model of evolutionary change in proteins. Matrices for detecting distant relationship"*. In Atlas of Protein Sequence and Structure, vol. 15, suppl 3 (ed. Dayhoff, M. O.), pp.345-358. National Research Foundation, Washington, D. C.

19. Dietterich, T. G. & Bakiri, G. (1995). *"Solving multiclass learning problems via error-correcting output codes"*, Journal of Artificial Intelligence Research 2, 263-286.

20. Ding CH, Dubchak I. (2001). *"Multi-class protein fold recognition using support vector machines and neural networks"*, Bioinformatics, 17(4), 349-358.

21. Dosztanyi, Z., Fiser, A. & Simon, I. (1997). *"Stabilization centers in proteins: identification, characterization and prediction"*, J Mol Biol, 272, 597-612.

22. Dubchak, I., Holbrook, S. R. & Kim, S. H. (1993a). *"Prediction of protein folding class from amino acid composition"*, Proteins, 16, 79-91.

23. Dubchak, I., Holbrook, S. R. & Kim, S. H. (1993b). *"Comparison of two variations of neural network approach to the prediction of protein folding pattern"*, Ismb, 1, 118-126.

24. Dubchak, I., Muchinik, I., Holbrook, S. R. & Kim, S. H. (1995). *"Prediction of protein folding class using global description of amino acid sequence"*, Proc Natl Acad Sci USA 92, 8700-8704.

25. Dubchak, I., Muchinik, I. & Kim, S. H. (1997). *"Protein folding class predictor for SCOP: approach base on global descriptors"*, Ismb, 5, 104-107.

26. Duh, M. S., Walker, A. M., Pagano, M. & Kronlund, K. (1998). *"Prediction and cross-validation of neural networks versus logistic regression: using hepatic disorders as an example"*, Am J Epidemiol, 147, 407-413.

27. Eddy, S. R., Mitchison, G. & Durbin, R. (1995). *"Maximum Discrimination hidden Markov models of sequence consensus"*, J Comp Biol, 2, 9-23.

28. Emanuelsson, O., Nielson, H., Brunak, S. & von Heijne, G. (1999*). "CholroP, a neural network-based method for predicting chloroplast transit peptides and their cleavage sites"*, Protein Sci, 8, 978-984.

29. Emanuelsson, O., Nielson, H., Brunak, S. & von Heijne, G. (2000). *"Predicting subcelluar localization of protein based on their N-terminal amino acid sequence"*, J Mol Biol, 300, 1005-1016.

30. Fahlman, S. E. (1988). *"Fast learning variations on back-propagation: An empirical study"*, In Processings of the 1988 Connectionist Models Summer School (ed. Hinton, G. E., Sejnowski, T. J. & Touretzkey, D. S.), pp.38-51. Morgan Kaufmann, San Mateo, CA.

31. Fahlman, S. & Lebiere, C. (1990). *"The cascade-correlation learning architecture"*, Adv Neural Inf Process Syst, 2, 524-532.

32. [Fariselli et al., 93] Fariselli, P., Compiani, M. & Casadio, R. (1993). *"Predicting secondary structure of membrane proteins with neural networks"*, Euro Biophys J, 22, 41-51.

33. Fariselli P, Casadio R. (2000). *"Prediction of the number of residue contacts in proteins"*, Proc Int Conf Intell Syst Mol Biol, 8, 46-51.

34. Fausett, Laurene V., (1994). Fundamentals of neural networks: architectures, algorithms, and applications, Prentice-Hall Inc, New Jersey.

35. Ferran, E. A. & Ferrara, P. (1992). *"Clustering proteins into families using artificial neural networks"*, Comput Appl Biosci, 8, 39-44.

36. Ferran, E. A. & Pflugfelder, B. (1993). *"A hybrid method to cluster protein sequences based on statistics and artificial neural networks"*, Comput Appl Biosci, 9. 671-680.

37. Ferran, E. A., Pflugfelder, B. & Ferrara, P. (1994). *"Self-organized neural maps of human protein sequences"*, Protein Sci, 3, 507-521.

38. Fleischmann, R.D., Adams, M.D., White, O., Clayton, R.A., Kirkness, E.F., Kerlavage, A.R., Bult, C.J., Tomb, J.F., Dougherty, B.A., Merrick, J.M., et al. (1995). *"Whole-genome random sequencing and assembly of Haemophilus influenzae Rd."*, Science, 269, 495-512.

39. Fraser, C.M., Gocayne, J.D., White, O., Adams, M.D., Clayton, R.A., Fleischmann, R.D., Bult, C.J., Kerlavage, A.R., Sutton, G., Kelley, J.M., et al. (1995). *"The minimal gene complement of Mycoplasma genitalium"*, Science, 270, 397-403.

40. Giuliano, F., Arrigo, P., Scalia, F., Cardo, P. P. & Daminani, G. (1993). *"Potentially functional regions of nucleic acids recognized by a Kohonen's self-organizing map"*, Comput Appl Biosci, 9, 687-693.

41. Gracy, J. & Argos, P. (1998). *"Automated protein sequence database classification. I. Integration of compositional similarity search, local similarity search, and multiple sequence alignment"*, Bioinformatics, 14, 164-173.

42. Granjeon, E. & Tarroux, P. (1995). *"Detection of compositional constraints in nucleic acid sequences using neural networks"*, Comput Appl Biosci, 11, 29-37.

43. Goffeau, A., Barrell, B.G., Bussey, H., Davis, R.W., Dujon, B., Feldmann, H., Galibert, F., Hoheisel, J.D., Jacq, C., Johnston, M., Louis, E.J., Mewes, H.W., Murakami, Y., Philippsen, P., Tettelin, H. and Oliver, S.G. (1996). *"Life with 6000 genes"*, Science, 274, 563-567.

44. Guigo, R., Knudsen, S., Drake, N. & Smith, T. (1992). *"Prediction of gene structure"*, J Mol Biol, 226, 141-157.

45. Gulukota, K., Sidney, J., Sette, A. & Delisi C. (1997). *"Two complementary methods for predicting peptides binding major histocompatibility complex molecules"*, J Mol Biol, 267, 1258-1267.

46. Gutteridge A, Bartlett GJ, Thornton JM. (2003). *"Using a neural network and spatial clustering to predict the location of active sites in enzymes"*, J Mol Biol, 330(4), 719-734.

47. Hassoun, M. H. (1995). Fundamentals of Artificial Neural Networks. MIT P, Cambridge.

48. Haykin, S. (1999). Neural Networks: a comprehensive foundation, 2nd ed, Prentice Hall, New Jersey

49. Hecht-Nielsen, R. (1987). Counterpropagation networks. Applied Optics, 26, 4949-4984.

50. Henikoff, S. & Henikoff, J. G. (1994). *"Protein family classification based on searching a database of blocks"*, Genomics, 19, 97-107.

51. Henikoff, J. G., Henikoff, S. & Pietrokovski, S. (1999). *"New features of the Blocks Database servers"*, Nucleic Acids Res, 27, 226-228.

52. Hinton, G. E. (1992). *"How neural networks learn from experience"*, Sci Am, 267, 144-151.

53. Hofmann, K., Bucher, P., Falquet, L. & Bairoch, A. (1999). *"The PROSITE database, its status in 1999"*, Nucleic Acids Res, 27, 215-219.

54. Hornik, K., Stinchcombe, M. & White, H (1989). *"Multilayer feedforward networks are universal approximators"*, NEUNET, 2, 359-366.

55. Hughey, R. & Krogh, A. (1996). *"Hidden Markov Models for Sequence Analysis: Extension and Analysis of the Basic Method"*, Computer Applications in the Biosciences, 12, No. 2, 95-107.

56. Jagla, B. & Schuchhardt, J. (2000). *"Adaptive encoding neural networks for the recognition of human signal peptide cleavage sites"*, Bioinformatics, 16(3), 245-250.

57. Jones, D. T. (1999). *"GenTHREADER: An efficient and reliable protein fold recognition method for genomic sequences"*, J Mol Biol, 287, 797-815.

58. Kanaya S, Kinouchi M, Abe T, Kudo Y, Yamada Y, Nishi T, Mori H, Ikemura T. (2001). *"Analysis of codon usage diversity of bacterial genes with a self-*

organizing map (SOM): characterization of horizontally transferred genes with emphasis on the E. coli O157 genome", Gene, 276(1-2), 89-99.

59. Karlin, S., Ost, F. & Blasisdell, B. E. (1989). "Pattern in DNA and amino acid sequences and their statistical significance", In Mathematical Method for DNA Sequence (ed. Waterman, M. S.), pp. 133-157, CRCP, Boca Raton.

60. Kaur, H & Raghava, GP (2003). *"A neural-network based method for prediction of gamma-turns in proteins from multiple sequence alignment"*, Protein Science, 12(5), 923-929.

61. Kneller, D. G., Cohen, F. E. & Langridge, R. (1990). "Improvements in protein secondary structure prediction by an enhanced neural network", J Mol Biol, 214, 171-182.

62. Kohonen, T. (1990). "The Self-organizing map", Proc., IEEE 78, 1464-1480.

63. Konopka, A. K. (1994). *"Sequences and codes: Fundamentals of biomolecular cryptology."* In: Biocomputing: Informatics and Genome Projects. (Ed., Smith, D.), Academic Press, San Diego, CA, pp. 119-174.

64. Krogh, A., Brown, M., Mian, I.S., Sjolander, K., Haussler, D. (1994). *"Hidden Markov Models in Computational Biology: Applications to Protein Modeling"*, Journal of Molecular Biology, 235, No. 5, 1501-1531.

65. Ladunga, I., Czako, F., Casbai, I. & Geszti, T. (1991). *"Improving signal peptide prediction accuracy by simulated neural network"*, Comput Appl Biosci, 7, 485-487.

66. Lundstrom, J., Rychlewski, L., Bujnicki, J. & Elofsson, A. (2001). *"Pcons: a neural-network-based consensus predictor that improves fold recognition"*, Protein Sci, 10(11), 2354-2362.

67. Milik, M., Kolinski, A. & Skolnick, J. (1995). *"Neural network system for the evaluation of side-chain packing in protein stuructures"*,Protein Eng, 8, 225-36.

68. Milutinovic, V., Ngom, A. & Stojmenovic, I. (2001). "STRIP --- *A Strip-Based Neural Netwrok Growth Algorithm for Learning Multiple-Valued Functions"* , IEEE Transactions on Neural Networks, 12, 212-227.

69. Murray, A. F. (1995). Applications of Neural Networks, Kluwer Academic Publishers, Boston.

70. Nair, T. M. Tambe, S. S. & Kulkarni, B. D. (1994). *"Application of artificial neural networks for prokaryotic transcription terminator prediction"*, FEBS Lett, 346, 273-277.

71. Nakata, K. (1995). *"Prediction of zinc finger DNA binding protein"*, Comput Appl Biosci, 11, 125-131.

72. Needleman, S. B. & Wunsch, C. D. (1970). *"A general method applicable to the search for similarities in the amino acid sequences of two proteins"*, J Mol Biol, 48, 443-453.

73. Nielsen, H., Engelbrecht, J., von Heijne, G. & Brunak, S. (1996). *"Defining a similarity threshold for a functional protein sequence pattern: the signal peptide cleavage site"*, Proteins, 24, 165-177.

74. Nielsen, H., Engelbrecht, J., Brunak, S. & von Heijne, G. (1997). *"Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites"*, Protein Eng, 10, 1-6.

75. Nielsen H, Brunak S, von Heijne G. (1999). *"Machine learning approaches for the prediction of signal peptides and other protein sorting signals"*, Protein Eng, 12(1), 3-9.

76. Pasquier, C., Promponas V. J. & Hamodrakas S. J. (2001). *"PRED-CLASS: Cascading Neural Networks for generalized protein classification and genome-wide applications"*, Protein: Structure, Function, and Genetics 44, 361-369.

77. Pearson, W. R. & Lipman, D. J. (1988). *"Improved tools for biological sequence comparsions"*, Proc Nat Acad Sci USA, 85, 2444-2448.

78. Pearson, W. R (1991). *"Searching protein sequence libraries: comparison of the sensitivity and the selectivity of the Smith-Waterman and FASTA algorithms"*, Genomics, 11, 635-650.

79. Pedersen, A. G. & Nielson, H. (1997). *"Neural network prediction of translation initiation sites in eukaryotes: perspectives for EST and genome analysis"*, Ismb, 5, 226-233.

80. Pollastri G, Baldi P, Fariselli P, Casadio R. (2002). *"Prediction of coordination number and relative solvent accessibility in proteins"*, Proteins, 47(2), 142-153.

81. Qian, N. & Seijnowski, T. J. (1988). "Predicting the secondary structure of globular proteins using neural network models", J Mol Biol, 202, 865-684.

82. Reczko, M., Hatzigeorgiou, A., Mache, N., Zell, A. & Suhai, S. (1995). *"A parallel neural network simulator on the connection machine CM-5"*, Comput Appl Biosci, 11, 309-315.

83. Reed, R. (1993). *"Pruning algorithems – A survey"*, IEEE Trans. Neural Networks, 4, 740-747.

84. Rice DW, Eisenberg D. (1997). *"A 3D-1D substitution matrix for protein fold recognition that includes predicted secondary structure of the sequence"*, J Mol Biol, 267(4), 1026-1038.

85. Ritchie, M. D., White, B. C., Parker, J. S., Hahn, L. W. & Moore, J. H. (2003). *"Optimization of neural network architecture using genetic programming improves detection and modeling of gene-gene interactions in studies of human diseases"*, BMC Bioinformatics, 4(1), 28.

86. Richter, S. and Lamppa, G. K. (1998). *"A chloroplast processing enzyme functions as the general stromal processing peptidase"*, Proc Natl Acad Sci USA, 95, 7463-7468.

87. Riedmiller, M. & Braun, H. (1993). *"A direct adaptive method for faster backpropagation learning: the Rprop algorithm"*, In Proceedings of the IEEE International Conference on Neural Networks (ICNN 93), pp. 586-591.

88. Rost, B. & Sander, C. (1993a). *"Improved prediction of protein secondary structure by use of sequence profiles and neural networks"*, Proc Natl Acad Sci USA, 90, 7558-7562.

89. Rost, B. & Sander, C. (1993b). *"Prediction of protein secondary structure at better than 70% accuracy"*, J Mol Biol, 232, 584-599.

90. Rost, B. & Sander, C. (1994a). *"Combining evolutionary information and neural networks to predict protein secondary structure"*, Protein, 19, 55-72.

91. Rost, B. & Sander, C. (1994b). *"Conservation and prediction of solvent accessibility in protein families"*, Protein, 20, 216-226.

92. Rost, B. (1996). *"PHD: predicting one-dimensional protein structure by profile-based neural networks"*, Methods Enzymol, 266, 525-539.

93. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1986). *"Learning representations by back-propagation errors"*, Nature (London), 323, 533-536.

94. Rumelhart, D. E. & McClelland, J. L. (1986). Parallel Distributed Processing, Vols. 1 and 2, MIT P, Cambridge.

95. Sasagawa, F. & Tajiman, K. (1993). *"Prediction of protein secondary structures by a neural network"*, Comput Appl Biosci, 9, 147-152.

96. Salzberg, S., Delcher, A. L., Fasman, K. H. & Henderson, J. (1997). *"A decision tree system for finding genes in DNA"*, J Comput Biol, 5, 667-680.

97. Sellers, P. H. (1974). *"On the theory and computation of evolutionary distances"*, SIAM J Appl Math, 26, 787-793.

98. Smith, T. F. & Waterman, M. S. (1981). *"Comparison of bio-sequences"*, Adv Appl Math, 2, 482-489.

99. Snyder, E. E. & Stormo, G. D. (1995). *"Identification of protein coding regions in genomic DNA"*, J Mol Biol, 248, 1-18.

100. Sonnhammer, E.L., Eddy, S. R., Durbin, R. (1997). *"PFAM: A Comprehensive Database of Protein of Protein Domain Families Based on Seed Alignments"*, Protein, 28, No. 3, 405-420.

101. Staden, R. (1984). *"Computer methods to locate signals in nucleic acid sequences"*, Nucleic Acids, Res 12, 505-519.

102. Stolorz, P., Lapedes, A. & Xia, Y. (1992). *"Predicting protein secondary structures using neural net and statistical methods"*, J Mol Biol, 225, 363-377.

103. Stormo, G. D., Schneider, T. D., Gold, L. & Ehrenfeucht, A. (1982). *"Use of the 'Perceptron' algorithm to distinguish translational initiation sites in E. Coli"*, Nucleic Acids Res, 10, 2997-3011.

104. Stormo, G. D. (1990a). *"Identifying regulatory sites from DNA sequence data"*, In: Structure and Methods, Adenine Press, Guiderland, NY, pp. 103-112.

105. Stormo, G. D. (1990b). *"Consensus patterns in DNA"*, Methods Enzymol, 183, 211-221.

106. Turner, P.C., Mclennan, A. G., Bates, A. D., White, M.R.H. (2000). Instant Notes in Molecular Biology, BIOS Scientific Publishers Limited.

107. Von Heijne, G. (1986). *"A new method for predicting signal sequence cleavage sites"*, Nucleic Acids Res, 14, 4683-4690.

108. Wang, J. T. L., Ma, Q., Shasha, D., and Wu C. H (2001). *"New techniques for extracting features from protein sequences"*, IBM Systems Journal (Special Issue on Deep Computing for the Life Sciences) 40(2), 426-441.

109.Waterman, M. S. & Jones, R. (1990). *"Consensus methods for DNA and protein sequence alignment"*, Methods Enzymol, 183, 221-237.

110.Wu, C., Whitson, G., McLarty, J., Ermongkonchai, A. & Chang, T. C. (1992). *"Protein classification artificial neural system"*, Protein Sci, 1, 667-677.

111.Wu, C. & Shivakumar, S. (1994). *"Back-propagation and counter-propagation neural networks for phylogenetic classification of ribosomal RNA sequences"*, Nucleic Acids Res, 22, 4291-2299.

112.Wu, C. H., Berry, M., Shivakumar, S. & McLarty, J. (1995). *"Neural networks for full-scale protein sequence classification: Sequence encoding with singular value decomposition"*, Machine Learning, 21, 177-193.

113.Wu, C. H. (1996). *"Gene Classification Artificial Neural System"*, Methods Enzymol, 266, 71-78.

114.Wu, C. H., Zhao, S., Chen, H. L., Lo, C. J. & McLarty, J. (1996). *"Motif identification neural network design for rapid and sensitive protein family search"*, Comput Appl Biosci, 12, 109-118.

115.Wu, C. H., Chen, H. L. & Chen, S. C. (1997). *"Counter-propagation neural networks for molecular sequence classification: supervised LVQ and dynamic node allocation"*, Applied Intelligence, 7, 27-38.

116.Wu, C. H., Huang, H. & McLarty, J. (1999a). "Gene family identification network design", International Journal on Artificial Intelligence Tools, 8(4), 419-432.

117.Wu, C. H., Shivakumar, S. & Huang, H. (1999b). *"ProClass Protein Family Database"*, Nucleic Acids Res, 27, 272-274.

118.Wu, C. H. & McLarty, J. (2000). Neural networks and genome informatics, First edition, Elsevier Science Ltd.

119.Xin, Y., Carmeli, T. T., Liebman, M. N. & Wilcox, G. L. (1993). "Use of the backpropagation neural network algorithm for prediction of protein folding patterns", In proceedings of the Second International Conference on Bioinformatics, Supercomputing, and Complex Genome Analysis (ed. Lim, H. A., Fickett, J. W., Cantor, C. R. & Robbins, R. J.), pp.359-375, World Scientific, New Jersey.

120.Xu, Y., Mural, R. J. & Uberbacher, E. C. (1994). "Constructing gene models from accurately predicted exons: an application of dynamic programming", Comput Appl Biosci, 10, 613-623.

Appendix A: Part of Dataset

>NF00246442 Globin (Myoglobin) [Isoparorchis hypselobagri]
VLTKDEFDSLLHELDPKIDTEEHRMELGLGAYTELFAAHPEYIKKFSRLQEATPANVMAQ
DGAKYYAKTLINDLVELLKASTDEATLNTAIARTATKDHKPRNVSGAEFQTGEPIFIKYF
SHVLTTPANQAFMEKLLTKIFTGVAGQL

>NF00867444 Globin I (HbI) [Scapharca broughtonii]
PSVQGAAAQLTADVKKDLRDSWKVIGSDKKGNGVALMTTLFADNQETIGYFKRLGNVSQG
MANDKLRGHSITLMYALQNFIDQLDNTDDLVCVVEKFAVNHITRKISAAEFGKINGPIKK
VLASKNFGDKYANAWAKLVAVVQAAL

>NF00244223 Hemoglobin alpha-2 chain (HB 2) [Catharacta maccormicki]
MLTADDKKLILQIWEKVLGHQEDFGAEALERMFITYPQTKTYFPHFDLQHGSDQIRSHGK
KVVTALGNAVKSLDNLSQALSELSNLHAYNLRVDPVNFKLLSQCFQVVLAVHLGKDYTPE
VHAAFDKFLSAVAAVLSEKYR

>NF00244221 Hemoglobin beta chain [Catharacta maccormicki]
VHWSAEEKQLITGLWGKVNVADCGAEALARLLIVYPWTQRFFSSFGNLSSPTAIIGNPMV
RAHGKKVLTSFGEAVKNLDNIKNTFAQLSELHCDKLHVDPENFRLLGDILIIVLAAHFAK
EFTPDCQAAWQKLVRVVAHALARKYH

>NF00244220 Hemoglobin alpha-1 chain (HB 1) [Catharacta maccormicki]
VLSGSDKNNVKGVFGKIGGHAEEYGAETLERMFATYPQTKTYFPHFDLQHGSAQVKAHGK
KVAAALVEAANHIDDISGALSKLSDLHAQKLRVDPVNFKLLGQCFLVVVAIHHPSVLTPE
VHASLDKFLCAVGNVLTAKYR

>NF01102503 myoglobin [Elaenia flavogaster]
QISGVHF

>NF00233719 Hemoglobin alpha-D chain [Coturnix japonica]
MLTAEDKKLIQQAWEKASSHQEDFGAEALLRMFTAYPQTKTYFPHFDLSPGSDQIRGHGK
KVLAALGNAVKNIDDLSQAMAELSNLHAFNLRVDPVNFKLLSQCIQVVLAAHMGKDYSPE
VHAAFDKFLSAVAAVLAGKYR

>NF00233674 Hemoglobin alpha-A chain [Coturnix japonica]
MVLSAADKTNVKGIFAKIAGHAEEYGAEALDRMFTTYPQTKTYFPHFDVSHGSAQIKGHG
KKVAAALVEAANHIDDIAGTLSKLSDLHAQKLRVDPVNFKLLGQCFLVVVAIHHPAALTP
EVHASLDKFLCAVGTVLTAKYR

>NF00233658 Hemoglobin beta chain [Coturnix japonica]
VHWSAEEKQLITGLWGKVNVAECGAEALARLLIVYPWTQRFFASFGNLSSPTAILGNPMV
RAHGKKVLTSFGDAVKNLDNIKNTFSQLSELHCDKLHVDPENFRLLGDILIIVLAAHFTK
DFTPECQAAWQKLVRVVAHALARKYH

>NF00231585 hemoglobin beta chain, major [Cricetinae gen. sp.]
MVHLTDAEKALVTGLWGKVNADAVGAEALGRLLVVYPWTQRFFEHFGDLSLPVAVMNNPQ
VKAHGKKVIHSFADGLKHLDNLKGAFSSLSELHCDKLHVDPENFKLLGNMIIIVLSHDLG
KDFTPSAQSAFHKVVAGVANALAHKYH

>NF00231056 Putative globin [Burkholderia pseudomallei]
MKRAHAVSSGDPPGRSPMGLRDLFFDSTDSTDSRMTDVTDDAPSPPTAFELVGGEARVRE
LVDRFYDLMDLEPEFAGIRALHPPTLEGSRDKLFWFLCGWLGGPDHYIERFGHPRLRARH
LPFPIASSERDQWLRCIAWAMQDVGLDEPLRERLMHSFHDTADWMRNRPG

>NF01101863 Hemoglobin beta chain [Procolobus badius]
VHLTPDEKNAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFDSFGDLSTADAVMGNPKV
KAHGKKVLGAFSDGLAHLDNLKGTFAQLSELHCDKLHVDPENFKLLGNVLVCVLAHHFGK
EFTPQVQAAYQKVVAGVANALAHKYH

>NF01101862 Hemoglobin alpha chain [Procolobus badius]
VLSPADKTNVKTAWGKVGGHGGEYGAEALERMFLSFPTTKTYFPHFDLSHGSAQVKGHGK
KVADALTLAAAHVDDMPSALSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHHPAEFTPA
VHASLDKFLASVSTVLTSKYR

>NF01257625 alpha globin [Hydrophis melanocephalus]
MVLTAEDRRLLQASVGKLGCRLEDIGADRLNRLFIVFPQSKTYFSHYNLSPGSKDIIHQG
EKVGKALDSALKHLDDIRGTLSQLSDLHAYNLRVDPVNFKLLAKCFQVSLATHLRTEYSA

LVCLAWDKFFEQVSDVLSEKYR

>NF00226726 hemoglobin V [Tokunagayusurika akamusi]
AFVGLSDSEEKLVRDAWAPIHGDLQGTANTVFYNYLKKYPSNQDKFETLKGHPLDEVKDT
ANFKLIAGRIFTIFDNCVKNVGNDKGFQKVIADMSGPHVARPITHGSYNDLRGVIYDSMH
LDSTHGAAWNKMMDNFFYVFYECLDGRCSQFS

>NF00226725 hemoglobin VII [Tokunagayusurika akamusi]
DPTWVDMEAGDIALVKSSWAQIHDKEVDILYNFFKSYPASQAKFSAFAGKDLESLKDTAP
FALHATRIVSVINEAIALMGVAENRPALKNVLKQQGINHKGRGVTAAHFEEFETALEAFL
ESHASGYNAGTKKAWDSAFNNMYSVVFPEL

>NF00224572 Larval alpha-globin [Hynobius retardatus]
MTLTAADKALIVSLWGKIAGHTEALGAEALERMFESFPQSKTYFSHFDLHHGSADVKSHG
AKVLNAIGLAAQHVDDLDHALSKLSDLHAYNLRVDPGNFKLLSHTIQVVLANHFPAEFTP
EAHAAFDKFLAAVSTVLTSKYR

>NF00224571 Larval beta-globin [Hynobius retardatus]
MVHWTAEEKAAISSVWKQVNVESDGQEALARLLIVYPWTQRYFSSFGDLSSPAAICANAK
VRAHGKKVLSALGAGANHLDDIKGNFADLSKLHADTLHVDPNNFLLLANCLVIVLARKLG
AAFNPQVHAAWEKFLAVSTAALSRNYH

>NF00224570 Adult beta-globin [Hynobius retardatus]
MVFTNDERKDIHEVWGKVKADKLGADALARVLIVNPWTRKHFSSFGDLSTPEAILHNPKI
AAHGAKVVHSIIEASKHLDDLKGYYADLSNVHCLKLHVDPNNFHLLAGIIVVMLGITLRE
DFTPHRQASVEKYLEAVCDALSHGYH

>NF00993701 2-domain hemoglobin protein subunit (Fragment) [Daphnia exilis]
ILSSHERSLIRKTWDQAKKDGDVAPQVLFRFVKAHPEYQKMFSKFANVPQSELLGDGNFL
AQAYTILAGLNVVIQSLFSQELMANQLNALGGAHQARGATPIMFEQFGGILEEVLAEELG
SAFTAEARQAWKNGLAALVAGIAKNLKKSEDLADPQTKLTPHQIRDVQRSWENIRNGRNA
LVSSIFVKLFKETPRIQKFFAKFGNVAVDSLAGNADYEKQVALVADRLDTIISAMDDKLQ
LLGNINYMRYTHTERGIPRGPWEDFSRLLLDV

>NF00993700 2-domain hemoglobin protein subunit (Fragment) [Daphnia exilis]
ILSSHERSLIRKTWDQAKKDGDVAPQVLXRFVKAHPEYQKMFSKFANVPQSELLGNGNFL
AQAYTILAGLNVVIQSLFSQELMANQLNALGGAHQARGATPIMFEQFGGILEEVLAEELG
STFTAEARQAWKNGLAALVAGIAKNLKKSEDLADPQTKLTPHQIRDVQRSWENIRNGRNA
LVSSIFVKLFKETPRIQKFFAKFGNVAVDSLAGNADYEKQVALVADRLDTIISAMDDKLQ
LLGNIGYMRYTHTERGIPRGPWG

>NF00224426 Myoglobin (Fragment) [Eopsaltria australis]
CQISXVDF

>NF00224424 Myoglobin (Fragment) [Erythrura gouldiae]
CQISGVHF

>NF00224421 Myoglobin (Fragment) [Manorina melanocephala]
CQISGVHF

>NF00224386 Hemoglobin [Trema virgata]
MSSSEVDKVFTEEQEALVVKSWAVMKKNSAELGLKFFLKIFEIAPSAKNLFSYLKDSPIP
LEQNPKLKPHAMTVFVMTCESAVQLRKAGKVTVRESNLKRLGAIHFKNGVVNEHFETRFA
LLETIKEAVPEMWSAEMKNAWGEAYDQLVAAIKFEVKPSST

>NF00224385 Hemoglobin [Trema virgata]
MSSSEVDKVFTEELEALVVKSWAVMKKNSAELGLKFFLKIFEIAPSAKNLFSYLKDSPIP
LEQNPKLKPHAMTVFVMTCESAVQLRKAGKVTVRESNLKRLGAIHFKNGVVNEHFETRFA
LLETIKEAVPEMWSAEMKNAWGEAYDQLVAAIKFEVKPSST

>NF00224384 Hemoglobin [Trema virgata]
MSSSEVDKVFTEEQEALVVKSWAVMKKNSAELGLKFFLKIFEIAPSAKNLFSYLKDSPIP
LEQNPKLKPHAMTVFVMTCESAVQLRKAGKVTVRESNLKRLGAIHFKNGVVNEHFETRFA
LLETIKEAVPEMWSAEMKNAWGEAYDLLVAAIKFEVKPSST

>NF00224091 Linker protein 3 [Sabella spallanzanii]
MRALLLMCIGAALVAAAVSDNCPECDKPDPATMXKLDENAARLNRLVARAVDVSDTIKRR
QSEYEDAEVQAQNMLQRLEKRNGVLCKGQFVCGGAAQQCADNLVVCDGVNDCDNGADEAT
DICNIKAKVGDSWVGSFVYHNCSSAKFSNLRFELKKFERKPNMMSLADVSAMLFWDDNTD
ADVTFIDSLPYDGTYKFYSRSAHLRSIMVVNSSRKCSTSYPERLTSCITMSTLLTAKRLC

>NF00246837 myosin-light-chain kinase (EC 2.7.1.117) A [Dictyostelium discoideum]
MTEVEKIYEFKEELGRGAFSIVYLGENKQTKQRYAIKVINKSELGKDYEKNLKMEVDILK
KVNHPNIIALKELFDTPEKLYLVMELVTGGELFDKIVEKGSYSEADAANLVKKIVSAVGY
LHGLNIVHRDLKPENLLLKSKENHLEVAIADFGLSKIIGQTLVMQTACGTPSYVAPEVLN
ATGYDKEVDMWSIGVITYILLCGFPPFYGDTVPEIFEQIMEVNYEFPEEYWGGISKEAKD
FIGKLLVVDVSKRLNATNALNHPWLKSNNSNNTIDTVKMKEYIVERRKNSNENWLTKRIF
Q

>NF00246787 Cell division control protein 2 homolog (EC 2.7.1.-) (p34 protein kinase) [Dictyostelium discoideum]
MESDGGLSRYQKLEKLGEGTYGKVYKAKEKATGRMVALKKIRLEDDGVPSTALREISLLK
EVPHPNVVSLFDVLHCQNRLYLVFEYLDQDLKKYMDSVPALCPQLIKSYLYQLLKGLAYS
HGHRILHRDLKPQNLLIDRQGALKLADFGLARAVSIPVRVYTHEIVTLWYRAPEVLLGSK
SYSVPVDMWSVGCIFGEMLNKKPLFSGDCEIDQIFRIFRVLGTPDDSIWPGVTKLPEYVS
TFPNWPGQPYNKIFPRCEPLALDLIAKMLQYEPSKRISAKEALLHPYFGDLDTSFF

>NF00243448 mitogen-activated protein kinase 1 homolog (clone Aspk9) [Avena sativa]
MDGAPVAEFRPTMTHGGRFLLYNIFGNQFEITSKYQPPIMPIGRGAYGIVCSVMNFETRE
MVAIKKIANAFDNNMDAKRTLREIKLLRHLDHENIVGLRDVIPPSIPQSFNDVYIATELM
DTDLHHIIRSNQELSEEHCQYFLYQLLRGLKYIHSANVIHRDLKPSNLLLNANCDLKICD
FGLARPSSESDMMTEYVVTRWYRAPELLLNSTDYSAAIDVWSVGCIFMELINRAPLFPGR
DHMHQMRLITEVIGTPTDDDLGFIRNEDARRYMRHLPQFPRRPFPGQFPKVQPAALDLIE
RMLTFNPLQRITVEEALEHPYLERLHDVADEPICTDPFSFDFEQHPLTEDQMKQLIFNEA
LELNPNFRY

>NF00222395 protein kinase PK1 (EC 2.7.1.-) [Craterostigma plantagineum]
MEKYELVKDIGSGNFGVARLMRNKETKELVAMKYIERGHKIDENVAREIINHRSLRHPNI
IRFKEVVLTPTHLAIVMEYAAGGELFERICNAGRFSEDEARYFFQQLICGVHYCHALQIC
HRDLKLENTLLDGSPAPRLKSCDFGYSKSSLLHSRPKSTVGTPAYIAPEVLSRREYDGKL
ADVWSCGVTLYVMLVGAYPFEDQEDPFFFRKTILWIMAVQYKIPDYVRISQSCRHLLSRI
FVANPLRRITIKEIKSHLWFLKNLPRELTEVAQAAYFRRDNPTYSLQSEEEIMKIVEEAK
VPPQASRSIGGFGWGTEEDEENVEEEEEGESEGEEDEYEKQVKAAHASGEVPPLT

>NF00222323 Cell division control protein 2 homolog B (EC 2.7.1.-) (Fragment) [Antirrhinum majus]
AYGVVYKARDIETNEDIALKKIRLEQEDEGVPSTAIREISLLKEMHHENIVNLKDVVHRE
KRLYLVFEYLDLDLKKHMDSCPEFSQDLHMVKMFLCQILRGVAYCHSHRVLHRDLKPQNL
LIDRGSNTIKLADFGLARAFGIPVRTFTHEVVTLWYRAPEVLLGSRHYSTPVDVWSVGCI
FAEMVNQKPLFPGDSEIDELHKIFRIIGTPNEDIWPGVTSLPDFKSSFPKWPPKELATIV
PNLGATGLDLLCKMLQLDPSKRITAKKALEHEYFKDIVLP

>NF00222242 Cell division control protein 2 homolog C (EC 2.7.1.-) [Antirrhinum majus]
MEKYEKLEKVGEGTYGKVYKALEKSTGQVVALKKTRLEMDEEGVPPTALREVSLLQMLSQ
SLYVVRLLSVEHVDCAKNGKPLLYLVFEYLDTDLKKFIDSHRKGPNPRPLPPQQIQSFLF
QLCKGVSHCHAHGVLHRDLKPQNLLLDKDKGVLKIADLGLARAFTVPLKSYTHEIVTLSY
RAPEVLLGSSHYSTAVDMSSVGCIFAEMVRRQALFPGDSEFQQLLHIFRLLGTPSDEQWP
GVSSLRDWHVYPQWEPQNSAPAVPSLGPDGLDLLTKTLKYDPADRISAKAALDHPYFDTL
DKSQF

>NF00222227 Cell division control protein 2 homolog D (EC 2.7.1.-) [Antirrhinum majus]
MAEEKSKSSAMDAFVKLEKVGEGTYGKVYRAMEKSTGKIVALKKTRLHEDEEGVPPTTLR
EVSLLRMLSRDPHVVRLLDVKQGQNKEGKTVLYLVFEYMDTDLKKYIRSFKQTGESIAPM
NVKSLMYQLCKGVAFCHGHGVLHRDLKPHNLLMDRKTMMLKIADLGLARAYTLPIKKYTH
EILTLWYRAPEVLLGATHYSPAVDMWSVACIFAELVTQKALFPGDSELQQLLHIFRLLGT
PNEEIWPGVSTLVDWHEYPQWTAQPISSAVPGLDEKGLNLLSEMLHYEPSRRISAKKAME
HPYFDELDKSGL

>NF00222226 protein kinase cdc2a (EC 2.7.1.-), cyclin-dependent [Antirrhinum majus]
MVSSHRSLMEQYEKVEKIGEGTYGVVYKARDRVTNETIALKKIRLEQEDEGVPSTAIREI
SLLKEMQHGNIVRLQDVVHSEKRLYLVFEYLDLDLKKHMDSCPEFSQDPRLVKMFLYQIL
RGIAYCHSHRVLHRDLKPQNLLIDRRTNALKLADFGLARAFGIPVRTFTHEVVTLWYRAP
EILLGSRHYSTPVDVWSVGCIFAEMVNQRPLFPGDSEIDELFKIFRVMGTPNEETWPGVT
SLPDFKSAFPKWPAKELAAVVPNLDASGLDLLDKMLRLDPSKRITARNALQHEYFKDIGF
VP

>NF00220509 shaggy protein kinase 6 (EC 2.7.1.-) [Petunia x hybrida]
MASTSKDTTTASTSTMDTRPENSEVDELPKELHEMKIKDERTDSHEDNLKDMEPAVVSGN
GAETGQIIVTTVSGRNGQQKQTLSYMAERVVGTGSFGVVFQAKCLETGESVAIKKVLQDR
RYKNRELQIMRTLDHPNVVKLRHCFYSTTEKNEVYLNLVLEYVSETVYRVSRHYSRMNQH
MPIIYVQLYTYQICRALNYMHSVLHVCHRDIKPQNLLVNPHTHQLKLCDFGSAKMLVPGE
PNISYICSRYYRAPELIFGATEYTTAIDMWSAGCVMAELLLGQPLFPGESGVDQLVEIIK
ILGTPTREEIRCMNPNYTEFKFPEIKAHPWHKIFQKKMPPEAVDLVSRLLQYSPTLRCTA
LEACAHPFFDALREPNACLPNGRPLPHLFNFTAQELSGAPADLRKALCLSTCASEFCEDS
SLGFVNGFREV

>NF00220455 Cell division control protein 2 homolog (EC 2.7.1.-) (p34cdc2) (Fragment) [Petunia x hybrida]
EGVPSTAIREISLLKEMQHANIVRLQDVVHSEKRLYLVFEYLDLDLKKHMDSSPEFSKDP
RLVKMFLYQILRGIAYCHSHRVLHRDLKPQ

>NF00220362 Mitogen-activated protein kinase homolog 1 (EC 2.7.1.37) (PMEK1) [Petunia x hybrida]
MATPVEPPNGIRTPGKHYYSMWQSLFEIDTKYVPIKPIGRGAYGIVCSSVNRETNEKVAI
KKINNAFENRIDALRTLRELKLLRHLRHENVIALKDVMMPIQRRSFKDVYLVYELMDTDL
HQIIKSSQTLSNDHCQYFLFQLLRGLKYLHSANILHRDLKPGNLLINANCDLKICDFGLA
RTSSGKDQFMTEYVVTRWYRAPELLLCCDNYGTSIDVWSVGCIFADVLGRKPVFPGTECL
NQLKLIINILGSQREEDIEFIDNPKARKYIKSLPYSPGTPFSRLYPNAHPLAIDLLQRML
VFDPSKRISVMEALQHPYMSPLYDPNTDPPAQVPINLDIDEDLVEETIREMMWEEILHYH
PEAAVALLWKLFYELLLVPSRHRP

>NF00220326 shaggy protein kinase 4 (EC 2.7.1.-) [Petunia x hybrida]
MASGIMPSAGGKHRTDAMLVDKLPEEINEMKIRDDKAEKEMEAAVVDGNGTEKGHIIVTT
IGGKNGEPKQTISYMAERVVGQGSFGIVFQAKCLETGETVAIKKVLQDKRYKNRELQTIR
LLDHPNVVALRHCFFSTTEKDELYLNLVLEYVPETVYRVLRHYSKANQQMPMIYVKLYTY
QIFRALAYIHGIGVCHRDIKPQNLLVNPHTHQLKLCDFGSAKVLVKGEPNISYICSRYYR
APELIFGATEYTFAIDIWSVGCVLAELLLGQPLFPGESGVDQLVEIIKVLGTPTREEIKS
MNPNYTEFKFPQIKAHPWHKIFHKRMPPEAVDLVSRLLQYSPNLRSTALEACTHTFFDEL
RDPKTRLPNGRPLPPLFNFRPQELKGASADLLNKLIPEHAKKQCTFLGV

>NF00217987 Mitogen-activated protein kinase homolog NTF6 (EC 2.7.1.37) (P43) [Nicotiana tabacum]
MENETNEKLEIKGIPTHEGKYVEYNVLGNFFEVTSKYIPPIQPVGRGAYGMVCCATNSET
KEEVAIKKIGNAFENRIDAKRTLREIKLLSHMDHENIIKIKDIVRPPDREEFNDVYIVYE
LMDTDLHQIIRSSQALTDDHCQYFLYQLLRGLKYVHSANVLHRDLKPSNLLLNANCDLKI
CDFGLARTTSEADFMTEYVVTRWYRAPELLLNCTEYTAAIDIWSVGCILMELIKREPLFP
GRDYAQQLGLIIALLGSPEDSDLGFLRSDNARKYVKHLPRVPRHPFSQKFPDVSPLALDL
AERMLVFDPAKRITVEDALNHPFLISLHEINEEPVCDSPFNFDFEQASLSEDDIKELIWN
EALKFDPNTMK

>NF00217967 protein kinase (EC 2.7.1.37) cdc2 homolog 2 [Nicotiana tabacum]
MDQYEKVEKIGEGTYGVVYKARDRVTNETIALKKIRLEQEDEGVPSTAIREISLLKEMQH
ANIVRLQDVVHSEKRLYLVFEYLDLDLKNTWITTPEFSEDPRLVKMFLYQILRGIAYCHS
HRVLHRDLKPQNLLIDRRTNALKLADFGLARAFGIPVRTFTHEVVTLWYRAPEILLGSRH
YSTPVDVWSVGCIFAEMVTQRPLFPGDSEIDELSRFRVMGTPNEDTWPGVTTLPDFKSAF
PKWPSKDLATIVPNLDGAGLDLLDKIVRLDPSKRITARNALEHEYFKDIGYVP

>NF00217900 mitogen-activated protein kinase (EC 2.7.1.-) WIPK [Nicotiana tabacum]
MADANMGAGGGQFPDFPSVLTHGGQYVQFDIFGNFFEITTKYRPPIMPIGRGAYGIVCSV
LNTELNEMVAVKKIANAFDIYMDAKRTLREIKLLRHLDHENVIGLRDVIPPPLRREFSDV
YIATELMDTDLHQIIRSNQGLSEDHCQYFMYQLLRGLKYIHSANVLHRDLKPSNLLVNAN
CDLKICDFGLARPNIENENMTEYVVTRWYRAPELLLNSSDYTAAIDVWSVGCIFMELMNR
KPLFGGKDHVHQIRLLTELLGTPTEADLGFLQNEDAKRYIRQLPQHPRQQLAEVFPHVNP
LAIDLVDKMLTFDPTRRITVEEALDHPYLAKLHDAGDEPICPVPFSFDFEQQGIGEEQIK
DMIYQEALSLNPEYA

>NF00217755 shaggy protein kinase (EC 2.7.1.-) 111 [similarity] [Nicotiana tabacum]
MNVMRRLKSIASGRSSVSDPGGDFSLKKVTVEQEVDHRVDGETQLEEQCTIAPKQDVTST
SEESTVCTLNVDTRPEKSRYEELPKEMNEMKIRDEKTNGHEDDIKDMEPAVVSGNGTETG
QIIVTTLSGRNGRQKKTLSYMAERVVGTGSFGVVFQAKCLETGESVAIKKVLQDRRYKNR

>NF00246832 GTP-binding protein SAS2 [Dictyostelium discoideum]
MTSPATNKPAAYDFLVKLLLIGDSGVGKSCLLLRFSDGSFTPSFIATIGIDFKIRTIELE
GKRIKLQIWDTAGQERFRTITTAYYRGAMGILLVYDVTDEKSFGSIRNWIRNIEQHASDS
VNKMLIGNKCDMTEKKVVDSSRGKSLADEYGIKFLETSAKNSVNVEEAFIGLAKDIKKRM
IDTPNDPDHTICITPNNKKNTCC

>NF00246813 Ras-like protein RASC [Dictyostelium discoideum]
MSKLLKLVIVGDGGVGKSALTIQLTQNQFIAEYDPTIENSYRKQVNIDEEVYMLDILDTA
GQEEYSAMRDQYIRSGRGFLIVYSIISRASFEAVTTFREQILRVKDLSTYPIVIIGNKAD
LPDKDRKVPPMEGKELAKSFGAPFLETSAKSRVNVEEAFFTLVREIKRWNQNPQNEEMLP
PKKRGCIIL

>NF00246525 Ras2 protein [Dictyostelium discoideum]
MTEYKLVIVGGGGVGKSALTIQLIQNHFIDEYDPTIEDSYRKQVSIDEETCLLDILDTAG
QEEYSAMRDQYMRTGQGFLCVYSITSRSSFDEINSFREQILRVKDKDRVPMILIGNKCDL
DTERVVSIAEGGRKGKSIGCPFLETSAKIRVNVEEAFYSLVREIRNDLKGNNDKPLKKGG
KKGLMKACIIF

>NF00231579 gene c-Ki-ras protein [Cricetinae gen. sp.]
MTEYKLVVVGAGGVG

>NF00220507 GTP-binding protein (clone PCRGP2) [Petunia x hybrida]
RYRAITSAYYRGAVGALLVYDVTRHVTFENVERWLKELRDHTDSNIVIMLVGNKADLRHL
RAVWSEDAKAFAEKESTF

>NF00220471 GTP-binding protein (clone PCRGP3) [Petunia x hybrida]
RFRTITSSYYRGAHGIIIVYDVTDQESFNNVKQWLSEIDRYASDNVNKLLVGNKCDLADN
RAVSYDTAKAFADEIGIP

>NF00220450 GTP-binding protein (clone PCRGP2) [Petunia x hybrida]
RYRAITSAYYRGAVGALLVYDVTRHVSFENVERWLKELRDHTDSNIVIMLVGNKADLRHL
RAVWSEDAKAFAEKESTF

>NF00220291 GTP-binding protein RAB1 [Petunia x hybrida]
MSNEYDYLFKLLLIGDSSVGKSCLLLRFADDSYVDSYISTIGVDFKIRTVELDGKTIKLQ
IWDTAGQERFRTITSSYYRGAHGIIIVYDVTEMESFNNVKQWLSEIDRYANESVCKLLVG
NKCDLVENKVVDTQTGKALADELGIPFLETSAKDSINVEQAFLTMAGEIKKKMGNQPAGA
KKTGSTVQIKGQPIEQKSNCCG

>NF00220262 GTP-binding protein (clone PCRGP1) [Petunia x hybrida]
RYRAITSAYYRGAVGALIVYDITRHVTFENVERWLKELRDHTDQNIVIMLVGNKADLRHL
RAVSTEDAKAFAERESTF

>NF00217917 Ras-related protein Rab11A [Nicotiana tabacum]
MANRVDHEYDYLFKMVLIGDSGVGKSNILSRFTRNEFCLESKSTIGVEFATRTLQVEGKT
VKAQIWDTAGQERYRAITSAYYRGAVGALLFYDITKRQTFDNVQRWLRELRDHRDSNIVI
ILAGNKSDLKHLRAVSEQDDQALVKKEGLSFLETSALEALNVDKAFQTILTDIYHIISKK
ALAAQEAAASTALPGQGTTINVSDNSANVKRGCCST

>NF00217861 Ras-related protein Rab11C [Nicotiana tabacum]
MASGYGDASQKIDYVFKVVLIGDSAVGKTQILARFARNEFSLDSKATIGVEFQTRTLVIQ
HKSVKAQIWDTAGQERYRAVTSAYYRGAVGAMLVYDITKRQTFDHIPRWLEELRAHADRN
IVIMLTGNKTDLEDQRAVPTEDAKEFAQKEGLFFLETSAMEATKLEDAFLTVLTEIFNIV
NKKNLAADENQSNSNPASLTGKKILVPGPGQVIPGKKACCSS

>NF00217763 Ras-related protein Rab5 [Nicotiana tabacum]
MASRRHNNLNAKLVLLGDMGAGKSSLVIRFVKGQFLEFQESTIGAAFFSSTVSVNNATVK
FEIWDTAGQERYHSLAPMYYRGAAAAIIVYDITSTESLARAKKWVQELQKQGNPNMVMAL
AGNKADLEDKRKVTAEEARLYAEENGLFFMETSAKTATNVNDIFYEIAKRLPRAQPAQNP

AGMVLEDKPAQGSQAASCCT

>NF00217708 Ras-related protein Rab11D [Nicotiana tabacum]
MASGYGDASQKIDYVFKVVLIGDSAVGKSQILARFARNEFSLDSKATIGVEFQTRTLAIQ
HKSVKAQIWDTAGQERYRAVTSAYYRGAVGAMLVYDITKRQTFDHIPRWLEELRAHADRN
IVIMLIGNKTDLEDQRAVPTEDAKEFAQKEGLFFLETSAMEATNLEDAFLTVLTEIFNIV
NKKNLAADDNQSNGNPASLTGKKILVPGPGQVIPEKKACCSS

>NF00217650 GTP-binding protein Rab7b [Nicotiana tabacum]
MPSPANVLKVIILGDSGVGKTSLMNQYVNKKFSNQYKATIGADFLTKEVQFEDRLFTLQI
WDTAGQERFQSLGVAFYRGADCCVLVYDVNSMKSFENLNNWREEFLIQASPSDPENFPFV
VLGNKVDIDGGNSRVVSEKKARAWCASKGNIPYFETSAKEGTFVEEAFQCIAKNALKSGE
EEEIYLPDTLDVGTSSQPRTGGCEC

>NF00217465 GTP-binding protein Rab6 [Nicotiana tabacum]
MAPVSALAKYKLVFLGDQSVGKTSIITRFMYDKFDNTYQATIGIDFLSKTMYLEDRTVAL
QLWDTAGQERFRSLIPSYIRDSSVAVIVYDVASRQSFLNTSKWIEEVRTERGSDVIIVLV
GNKTDLVEKRQVSIEEAEAKARELNVMFIETSAKAGFNIKPLFRKIAAALPGMETLSSAK
QEDMVDVNLKSSNANASQSQAQSGGCAC

>NF00217437 GTP-binding protein, ras-related [Nicotiana tabacum]
MATSGNKNMNAKLVLLGDVGAGKSSLLLRFVKGQFIEFQESTIGAAFFSQTVAVNDATVK
FEIWDTAGQERYHSLPPMYYRGAAAAIIVFDITNQASFDRAKKWVQELQAQGNPNMVMAL
AGNKADLLDARKVAAEEAQTYAQENGLFFMETSAKTASNVNDIFYEIANRLPRLQPAQNP
SGMVLMDRPAQTPASASCCS

>NF00217406 GTP-binding protein Rab7a [Nicotiana tabacum]
MAARRRMLLKVIILGDSGVGKTSLMNQYVNRKFSNQYKATIGADFLTKEIQFEDRLYTLQ
IWDTAGQERFQSLGVAFYRGADCCVLVYYVNVMKSFENLNNWREEFLIQASPSDPENFPF
IVLGNKIDVDGGNSRVVSEKKVKAWCASKGNIPYFETSAKEGFNVDAAFQCIAKNALKNE
PEDEIYLPDTIDVAGGSQSRSTGCES

>NF00217277 GTP-binding protein, ras-related [Nicotiana tabacum]
MNPEYDYLFKLLLIGDSGVGKSCLLLRFADDTYLESYISTIGVDFKIRTVEQDGKTIKLQ
IWDTAGQERFRTITSSYYRGAHGIIVVYDVTDQESFNNVKQWLSEIDRYASDGVNKILVG
NKSDLTANRVVSYETAKAFADEIGIPFLETSAKDATNVEQAFMAMTSAIKNRMASQPANN
SAKPPTVNIRGQPVTQSGGCCSS

>NF00217122 Ras-related protein Rab11B [Nicotiana tabacum]
MAGGYRAEDDYDYLFKLVLIGDSGVGKSNLLSRFSRNEFNLESKSTIGVEFATRSIRVDD
KIVKAQIWDTAGQERYRAITSAYYRGAVGALVVYDITRHVTFENVERWLKELRDHTDQNI
VIMLVGNKADLRHLRAVSTEDAKAFAERENTFFMETSALESLNVENAFTEVLTEIYKVVC
RKALEVGDDPAALPKGQTINVGKDDVSAVKKVGCCSS

>NF00216997 GTP-binding protein Rab11e [Nicotiana tabacum]
EEYLFKIVIIGDSAVGKSNLLTRYARNEFNLHSKATIGVEFQTQTLEIDGKEVKAQIWDT
AGQERFRAVTSAYYRGAFGALVVYDITRRTTFDSIPRWLDELKTHSDTTVARMLVGNKCD
LDNIRAVSVEEGKSLAESEGMFFMETSALDATNVNKAFDMVIREIYNSVSRKVLNSDSYK
AELSVNRVSLVDNGTDGSKQNQGYSCCSR

>NF00216853 GTP-binding protein Rab7c [Nicotiana tabacum]
MSMRRRTLLKVIVLGDSGVGKTSLMNRYVHKKFSQQYKATIGADFVTKELQIDDRLVTLQ
IWDTAGQERFQSLGVAFYRGADCCVLVYDVNVMRSFDNLDNWHEEFLKQANPPDPKTFPF
ILLGNKIDIDGGNSRVVSEKKAKEWCSSKGIPYFETSAKEDINVDAAFLFYCKTRLANEH
RQDIYFQGIPEAVSETEQRSGCAC

>NF00216609 Ras-related protein RHN1 [Nicotiana plumbaginifolia]
MASSSHNNLNAKLVLLGDMGAGKSSLVIRFVKGQFLEFQESTIGAAFFSSTLAVNNATVK
FEIWDTAGQERYHSLAPMYYRGAAAAIIVYDITSSDSFARAKKWVQELQKQGNPNMVMAL
AGNKADLEDRRKVTAEEARLYAEENGLFFMETSAKTAVNVNAIFYEIAKRLPRAQPAQNP

AGMVLVDRAAEGTRATSCCT

>NF00216545 Ras-related protein YPT3 [Nicotiana plumbaginifolia]
MAGYRADDEYDYLFKLVLIGDSGVGKSNLLSRFTKNEFNLESKSTIGVEFATKSLNIDNK
VIKAQIWDTAGQERYRAITSAYYRGAVGALLVYDVTRHVTYENVTRWLKELRDHTDPNIV
VMLIGNKSDLRHLVAVSTDEAKGLAEREGLYFMETSALEATNVENAFTEALTQIYRIVSK
KAVEAGDEGATSSAPPKGETINIKDEGSSWKKFGCCSS

>NF00215917 GTP-binding protein ypt2 [Lycopersicon esculentum]
MAAPPARARADYDYLIKLLLIGDTGVGKSCLLLRFSDGSFTTSFITTIGIDFKIRTIELD
GKRIKLQIWDTAGQERFRTITTAYYRGAMGILLVYDVTDESSFNNIRNWIRNIEQHASDN
VNKILVGNKADMDESKRAVPTSKGQALADEYGIKFFETSAKTNLNVEEVFFSIGKDIKQR
LSESDSKTEPQSIRINQSDQAGTAGQGAQKSSCCGS

>NF00200175 GTP-binding protein era homolog [Xylella fastidiosa]
MTQTVPYRCGRIAVIGRPNVGKSTLTNALVGTKISIVSNRPQTTRHRLLGIATFPEGQIV
LVDTPGLHREQKHPMNRLMNRTARGSLEDVDAALLVTESTHWNEEDTLAYNLLNDTGIPV
VLVINKIDRFKDKSALLPFLTHINENHTFTTIHPVSALKRKGLETLVSDLLALLPEGDPM
FSEDEITDRSQRFLASELVREQVMRQLGEELPYATTVEIEYFTENTGLFRIGALIWVERE
SQKAIVIGKGGARLKEIGVKARQQMERLFQTKVFLETWVRVRKDWSNNEAALKTFGYE

>NF00199210 GTP-binding protein era homolog [Thermotoga maritima]
MSIKSGFVALAGKPNVGKSTFINAVMGRKVVIVSDKPQTTRNRINCIYTDKDSQIIFVDT
PGIHKPLHRLGEYMVKAAVQALKGVDLVLFMLDAADGFTKTDEHVAKIVNDSGTKTIIAV
NKIDVAGEEKAKAVGELAKSMVENVVSVHYISALKGEGVFEVLEKIKEELPEGPQYYPED
MVTDRPLSFMAAEIIREKIFHLTRQEVPHSTAVVIEEIKDRPNGVLYIRANIYVERDSQK
GILIGKNGSMIKKIGTLAREELEFLVGRKVYLDLNVKVKEKWREKDFIILQEIGLKDDIK

>NF00818777 hypothetical protein all3030 [imported] [Nostoc sp. PCC 7120]
MIQKKICMVGAFATGKTSLVARFVYSIFSEKYQTTVGVKVDKKLVHLPENSINLIIWDIY
GEDELQKLQMSYMRGCSGYLLVVDGTRKNTLETAYRLQNSLEDNLGQIPFVLVMNKWDMT
DEWEIDPAELNSLVEKGWNVVKTSAKTGIGVEDVFQILAQKIMET

>NF00818361 GTP-binding protein [imported] [Nostoc sp. PCC 7120]
MSKEKTKLSQLENMTAELKVASIDNHISSLSGEVSIPQAPPEFKSGFIGIIGRPNVGKST
LMNQLVGQKIAITSPVAQTTRNRLRGIVTTPEAQLIFVDTPGIHKPHHQLGEVLVKNAKL
AIESVDVVLFVVDGAVACGAGDRFIADLLIHSKTPVILGINKVDQQPPDSQKIDESYQQL
ASAYQWPTVKFSAKTGAELPQLQELLVEHLEHGPYYYPPDLVTDQPERFIMGELIREQIL
LLTREEVPHSVAIAIDLVEETPTITRVLATIHVERDSQKGILIGKGGSMLKSIGSAAREQ
IQKLIAGKVYLELFVKVQPKWRHSRVRLAELGYRVEE

>NF00179855 GTP-binding protein UU491 [imported] [Ureaplasma urealyticum]
MVLEMIKKYGIVAIVGKPNVGKSTLINAIMKKKVSIISNKPQTTRNAVKEIYEDDESAII
FTDTPGFHEPSNKLDLFLNHEIEISYKEANVILFVTTMDKELDANDFEIINLIKEANKEN
IILVISKAEMAKNQDQIDERIHFLKKHIAFKDVVQISALHVINIDKLINTIKNYLHKDVV
TDYFRQKAEKEDKFVITEIIREQCLLNLNHEVPHGVGVEIDESKYNQEANHWIIKASIII
EKNSHKPIIIGQNGTMIKKISMGARKQLHEIYDCHISLTIFVKVENNWRDNNNIIKSLGY
KIKK

>NF00178584 GTP-binding protein era homolog [Mycoplasma pulmonis]
MKILFSTIIGRPNVGKSTLLNNILEYDLAIVSSKPQATRDQIMGIYSDDDYQLIFTDTPG
IYKTKTKFGENLNAQSYESLKDIDLVIFLSPANEEIGPGDEFICEKIKNFTNKIALITKI
DLENSEEVLKKKAEKLKSLGFKEIFAISSKDHGSIKKLINEIKKYSYEGERQFDEDMITE
KSEKFIAKESIREACIDLLEQELPHSILVEIDSFSEEEREEKNLVEIHSTIYVNKESQKG
ILIGKGGSKIKKIGISARKKIQRKLGVNVKLFLKIKVKKNWVNQEKIFKKFDN

>NF00177908 GTP-binding protein era homolog [Mycoplasma pneumoniae]
MESIRIGVLGLTNAGKSTLVNQLHKANNLLVSPMNNTTLLAVSTNTITHEKQNITFIDVP
GFSEKRHSSYELISQEIRKALSGIDVLLLVVRSDQQQKLPLLQTQLQPLKRYRDLTRVLL
INNFFDVVLQAEDKQAIVLDFKPQAVLETDLLHFDATSFWNQFNEVKLQANEFRKDVEFL
DADTDNFKILEALREQIIKYCSEEIPHVVRLEIVDKSFDQAKNLLKLHLSISVPKLSQKK

Appendix B: Part of Training Set of Single Network Approach

SNNS pattern definition file V4.2
generated at Thu Dec 18 21:28:56 2003

No. of patterns: 1693
No. of input units: 180
No. of output units: 3

pattern 1
0.0272108843537415 0.0 0.00680272108843537 0.0 0.0 0.0 0.0
0.00680272108843537 0.0 0.0 0.0612244897959184 0.0 0.0 0.0204081632653061
0.0 0.00680272108843537 0.0 0.0204081632653061 0.00680272108843537
0.0136054421768707 0.0 0.00680272108843537 0.0 0.00680272108843537 0.0
0.0 0.0476190476190476 0.0 0.0 0.0 0.054421768707483 0.00680272108843537
0.0136054421768707 0.0 0.0 0.0680272108843537 0.0 0.054421768707483 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.00680272108843537 0.0136054421768707 0.0136054421768707
0.00680272108843537 0.0 0.0 0.0 0.0272108843537415 0.0
0.00680272108843537 0.0 0.0 0.0 0.0 0.00680272108843537 0.0
0.0 0.0612244897959184 0.0 0.0 0.0204081632653061 0.0 0.00680272108843537
0.0 0.0204081632653061 0.00680272108843537 0.0136054421768707 0.0
0.00680272108843537 0.0 0.00680272108843537 0.0 0.0 0.0476190476190476
0.0 0.0 0.0 0.054421768707483 0.00680272108843537 0.0136054421768707 0.0
0.0 0.0680272108843537 0.0 0.054421768707483 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.00680272108843537 0.0136054421768707 0.0136054421768707 0.00680272108843537 0.0
0.0 0.0 0.0272108843537415 0.0 0.00680272108843537 0.0 0.0 0.0
0.0 0.00680272108843537 0.0 0.0 0.0612244897959184 0.0 0.0
0.0204081632653061 0.0 0.00680272108843537 0.0 0.0204081632653061
0.00680272108843537 0.0136054421768707 0.0 0.00680272108843537 0.0
0.00680272108843537 0.0 0.0 0.0476190476190476 0.0 0.0 0.0
0.054421768707483 0.00680272108843537 0.0136054421768707 0.0 0.0 0.0680272108843537
0.0 0.054421768707483 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.00680272108843537
0.0136054421768707 0.0136054421768707 0.00680272108843537 0.0 0.0 0.0
0 0 1
pattern 2
0.0206896551724138 0.0 0.0 0.0137931034482759 0.0 0.00689655172413793 0.0 0.0
0.0 0.0 0.0896551724137931 0.0206896551724138 0.0 0.00689655172413793 0.0
0.00689655172413793 0.0 0.0137931034482759 0.0 0.00689655172413793 0.0
0.0 0.0 0.00689655172413793 0.0 0.0137931034482759 0.0551724137931034
0.00689655172413793 0.0 0.00689655172413793 0.0413793103448276 0.0 0.0
0.00689655172413793 0.00689655172413793 0.0827586206896552 0.0 0.0551724137931034
0.0 0.0 0.0137931034482759 0.0 0.00689655172413793 0.0206896551724138 0.0
0.0 0.0 0.0 0.00689655172413793 0.0 0.0 0.0 0.0137931034482759
0.0 0.0137931034482759 0.0 0.00689655172413793 0.0 0.0 0.0 0.0
0.0206896551724138 0.0 0.0 0.0137931034482759 0.0 0.00689655172413793 0.0
0.0 0.0 0.0 0.0896551724137931 0.0206896551724138 0.0 0.00689655172413793
0.0 0.00689655172413793 0.0 0.0137931034482759 0.0 0.00689655172413793
0.0 0.0 0.0 0.00689655172413793 0.0 0.0137931034482759 0.0551724137931034
0.00689655172413793 0.0 0.00689655172413793 0.0413793103448276 0.0 0.0
0.00689655172413793 0.00689655172413793 0.0827586206896552 0.0 0.0551724137931034
0.0 0.0 0.0137931034482759 0.0 0.00689655172413793 0.0206896551724138 0.0
0.0 0.0 0.0 0.00689655172413793 0.0 0.0 0.0 0.0137931034482759
0.0 0.0137931034482759 0.0 0.00689655172413793 0.0 0.0 0.0
0.0206896551724138 0.0 0.0 0.0137931034482759 0.0 0.00689655172413793 0.0
0.0 0.0 0.0 0.0896551724137931 0.0206896551724138 0.0 0.00689655172413793
0.0 0.00689655172413793 0.0 0.0137931034482759 0.0 0.00689655172413793
0.0 0.0 0.0 0.00689655172413793 0.0 0.0137931034482759 0.0551724137931034
0.00689655172413793 0.0 0.00689655172413793 0.0413793103448276 0.0 0.0
0.00689655172413793 0.00689655172413793 0.0827586206896552 0.0 0.0551724137931034
0.0 0.0 0.0137931034482759 0.0 0.00689655172413793 0.0206896551724138 0.0
0.0 0.0 0.0 0.00689655172413793 0.0 0.0 0.0 0.0137931034482759
0.0 0.0137931034482759 0.0 0.00689655172413793 0.0 0.0 0.0
0 0 1
pattern 3
0.0142857142857143 0.0 0.0 0.0142857142857143 0.0428571428571429 0.0 0.0
0.00714285714285714 0.0 0.0 0.1 0.0214285714285714 0.0 0.0142857142857143

0.0 0.0 0.0142857142857143 0.0142857142857143 0.0 0.0214285714285714 0.0 0.0
0.0142857142857143 0.00714285714285714 0.0 0.0 0.0357142857142857 0.0 0.0
0.0 0.0642857142857143 0.00714285714285714 0.00714285714285714
0.00714285714285714 0.0 0.0785714285714286 0.00714285714285714 0.0428571428571429
0.0 0.00714285714285714 0.0285714285714286 0.0 0.0 0.0142857142857143 0.0
0.0 0.0 0.00714285714285714 0.00714285714285714 0.0 0.0 0.0
0.00714285714285714 0.0 0.0 0.0 0.0 0.0
0.0142857142857143 0.0 0.0 0.0142857142857143 0.0428571428571429 0.0 0.0
0.00714285714285714 0.0 0.0 0.1 0.0214285714285714 0.0 0.0142857142857143
0.0 0.0 0.0142857142857143 0.0142857142857143 0.0 0.0214285714285714 0.0 0.0
0.0142857142857143 0.00714285714285714 0.0 0.0 0.0357142857142857 0.0 0.0
0.0 0.0642857142857143 0.00714285714285714 0.00714285714285714
0.00714285714285714 0.0 0.0785714285714286 0.00714285714285714 0.0428571428571429
0.0 0.00714285714285714 0.0285714285714286 0.0 0.0 0.0142857142857143 0.0
0.0 0.0 0.00714285714285714 0.00714285714285714 0.0 0.0 0.0
0.00714285714285714 0.0 0.0 0.0 0.0 0.0
0.0142857142857143 0.0 0.0 0.0142857142857143 0.0428571428571429 0.0 0.0
0.00714285714285714 0.0 0.0 0.1 0.0214285714285714 0.0 0.0142857142857143
0.0 0.0 0.0142857142857143 0.0142857142857143 0.0 0.0214285714285714 0.0 0.0
0.0142857142857143 0.00714285714285714 0.0 0.0 0.0357142857142857 0.0 0.0
0.0 0.0642857142857143 0.00714285714285714 0.00714285714285714
0.00714285714285714 0.0 0.0785714285714286 0.00714285714285714 0.0428571428571429
0.0 0.00714285714285714 0.0285714285714286 0.0 0.0 0.0142857142857143 0.0
0.0 0.0 0.00714285714285714 0.00714285714285714 0.0 0.0 0.0
0.00714285714285714 0.0 0.0 0.0 0.0 0.0 0.0
0 0 1
pattern 4
0.0275862068965517 0.0 0.0206896551724138 0.0137931034482759 0.0137931034482759 0.0 0.0
0.0137931034482759 0.0 0.0 0.0758620689655172 0.0137931034482759 0.0 0.0
0.00689655172413793 0.0 0.00689655172413793 0.0137931034482759
0.00689655172413793 0.0137931034482759 0.0 0.0 0.00689655172413793 0.0
0.00689655172413793 0.00689655172413793 0.0206896551724138 0.0
0.00689655172413793 0.0 0.0551724137931034 0.0 0.0137931034482759 0.0
0.00689655172413793 0.0827586206896552 0.0 0.0206896551724138 0.00689655172413793
0.0 0.00689655172413793 0.0137931034482759 0.0 0.00689655172413793 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0275862068965517 0.0 0.0206896551724138 0.0137931034482759
0.0137931034482759 0.0 0.0 0.0137931034482759 0.0 0.0 0.0758620689655172
0.0137931034482759 0.0 0.0 0.00689655172413793 0.0 0.00689655172413793
0.0137931034482759 0.00689655172413793 0.0137931034482759 0.0 0.0
0.00689655172413793 0.0 0.00689655172413793 0.00689655172413793
0.0206896551724138 0.0 0.00689655172413793 0.0 0.0551724137931034 0.0
0.0137931034482759 0.0 0.00689655172413793 0.0827586206896552 0.0 0.0206896551724138
0.00689655172413793 0.0 0.00689655172413793 0.0137931034482759 0.0
0.00689655172413793 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0275862068965517 0.0
0.0206896551724138 0.0137931034482759 0.0137931034482759 0.0 0.0 0.0137931034482759 0.0
0.0 0.0758620689655172 0.0137931034482759 0.0 0.0 0.00689655172413793 0.0
0.00689655172413793 0.0137931034482759 0.00689655172413793 0.0137931034482759 0.0
0.0 0.00689655172413793 0.0 0.00689655172413793 0.00689655172413793
0.0206896551724138 0.0 0.00689655172413793 0.0 0.0551724137931034 0.0
0.0137931034482759 0.0 0.00689655172413793 0.0827586206896552 0.0 0.0206896551724138
0.00689655172413793 0.0 0.00689655172413793 0.0137931034482759 0.0
0.00689655172413793 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0 0 1
pattern 5
0.00714285714285714 0.0 0.00714285714285714 0.00714285714285714 0.0214285714285714
0.0 0.0 0.00714285714285714 0.0 0.0 0.0785714285714286 0.0142857142857143
0.0 0.0142857142857143 0.0 0.00714285714285714 0.0142857142857143 0.0142857142857143
0.0 0.0214285714285714 0.0 0.0 0.0214285714285714 0.0 0.0
0.00714285714285714 0.0214285714285714 0.0 0.0 0.0 0.05 0.0
0.00714285714285714 0.00714285714285714 0.0 0.0785714285714286 0.0 0.05
0.0 0.0 0.00714285714285714 0.0 0.0 0.0142857142857143
0.00714285714285714 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.00714285714285714 0.00714285714285714 0.00714285714285714 0.0 0.0
0.0 0.0 0.0 0.00714285714285714 0.0 0.00714285714285714
0.00714285714285714 0.0214285714285714 0.0 0.0 0.00714285714285714 0.0
0.0 0.0785714285714286 0.0142857142857143 0.0 0.0142857142857143 0.0
0.00714285714285714 0.0142857142857143 0.0142857142857143 0.0 0.0214285714285714 0.0

```
        0.0 0.0214285714285714 0.0 0.0 0.00714285714285714 0.0214285714285714 0.0
        0.0 0.0 0.05 0.0 0.00714285714285714 0.00714285714285714 0.0
        0.0785714285714286 0.0 0.05 0.0 0.0 0.00714285714285714 0.0 0.0
        0.0142857142857143 0.00714285714285714 0.0 0.0 0.0 0.0 0.0 0.0
        0.0 0.00714285714285714 0.00714285714285714 0.00714285714285714 0.0
        0.0 0.0 0.0 0.0 0.00714285714285714 0.0 0.00714285714285714
        0.00714285714285714 0.0214285714285714 0.0 0.0 0.00714285714285714 0.0
        0.0 0.0785714285714286 0.0142857142857143 0.0 0.0142857142857143 0.0
        0.00714285714285714 0.0142857142857143 0.0142857142857143 0.0 0.0214285714285714 0.0
        0.0 0.0214285714285714 0.0 0.0 0.00714285714285714 0.0214285714285714 0.0
        0.0 0.0 0.05 0.0 0.00714285714285714 0.00714285714285714 0.0
        0.0785714285714286 0.0 0.05 0.0 0.0 0.00714285714285714 0.0 0.0
        0.0142857142857143 0.00714285714285714 0.0 0.0 0.0 0.0 0.0 0.0
        0.0 0.00714285714285714 0.00714285714285714 0.00714285714285714 0.0
        0.0 0.0 0.0 0.0
0       0       1
# pattern 6
0.0136986301369863 0.0 0.0136986301369863 0.0205479452054795 0.0205479452054795 0.0 0.0
        0.0136986301369863 0.0 0.0 0.0958904109589041 0.0205479452054795 0.0 0.0
        0.00684931506849315 0.0 0.00684931506849315 0.0136986301369863
        0.00684931506849315 0.00684931506849315 0.0 0.0 0.0 0.0
        0.00684931506849315 0.0 0.0205479452054795 0.0 0.0 0.0
0       0       1
.
.
.
.
.

# pattern 1685
0.0319634703196347 0.0 0.0 0.0045662100456621 0.0 0.0 0.0045662100456621 0.0 0.0
        0.0045662100456621 0.0730593607305936 0.0 0.0 0.0182648401826484 0.0 0.0045662100456621
        0.0 0.0228310502283105 0.0 0.0045662100456621 0.0 0.0 0.0 0.0045662100456621
        0.0 0.0045662100456621 0.0684931506849315 0.0045662100456621 0.0091324200913242 0.0045662100456621
        0.0502283105022831 0.0 0.0045662100456621 0.0 0.0 0.0730593607305936 0.0045662100456621
        0.0319634703196347 0.0045662100456621 0.0045662100456621 0.0091324200913242 0.0 0.0
        0.0045662100456621 0.0 0.0 0.0045662100456621 0.0045662100456621 0.0091324200913242
        0.0045662100456621 0.0 0.0 0.0 0.0 0.0045662100456621 0.0045662100456621
        0.0045662100456621 0.0 0.0045662100456621 0.0045662100456621 0.0319634703196347 0.0 0.0
        0.0045662100456621 0.0 0.0 0.0045662100456621 0.0 0.0 0.0045662100456621
        0.0730593607305936 0.0 0.0 0.0182648401826484 0.0 0.0045662100456621 0.0
        0.0228310502283105 0.0 0.0045662100456621 0.0 0.0 0.0 0.0045662100456621 0.0
        0.0045662100456621 0.0684931506849315 0.0045662100456621 0.0091324200913242 0.0045662100456621
        0.0502283105022831 0.0 0.0045662100456621 0.0 0.0 0.0730593607305936 0.0045662100456621
        0.0319634703196347 0.0045662100456621 0.0045662100456621 0.0091324200913242 0.0 0.0
        0.0045662100456621 0.0 0.0 0.0045662100456621 0.0045662100456621 0.0091324200913242
        0.0045662100456621 0.0 0.0 0.0 0.0 0.0045662100456621 0.0045662100456621
        0.0045662100456621 0.0 0.0045662100456621 0.0045662100456621 0.0319634703196347 0.0 0.0
        0.0045662100456621 0.0 0.0 0.0045662100456621 0.0 0.0 0.0045662100456621
        0.0730593607305936 0.0 0.0 0.0182648401826484 0.0 0.0045662100456621 0.0
        0.0228310502283105 0.0 0.0045662100456621 0.0 0.0 0.0 0.0045662100456621 0.0
        0.0045662100456621 0.0684931506849315 0.0045662100456621 0.0091324200913242 0.0045662100456621
        0.0502283105022831 0.0 0.0045662100456621 0.0 0.0 0.0730593607305936 0.0045662100456621
        0.0319634703196347 0.0045662100456621 0.0045662100456621 0.0091324200913242 0.0 0.0
        0.0045662100456621 0.0 0.0 0.0045662100456621 0.0045662100456621 0.0091324200913242
        0.0045662100456621 0.0 0.0 0.0 0.0 0.0045662100456621 0.0045662100456621
        0.0045662100456621 0.0 0.0045662100456621 0.0045662100456621
1       0       0
# pattern 1686
0.0451127819548872 0.0 0.0 0.0 0.0 0.00751879699248120 0.00751879699248120 0.0
        0.00751879699248120 0.00751879699248120 0.0601503759398496 0.00751879699248120 0.0 0.00751879699248120
        0.0 0.0 0.00751879699248120 0.00751879699248120 0.0 0.0 0.0 0.0
        0.00751879699248120 0.0 0.0 0.0 0.0526315789473684 0.0150375939849624 0.0
        0.0150375939849624 0.0676691729323308 0.00751879699248120 0.0 0.0 0.0 0.037593984962406
        0.0 0.0676691729323308 0.0225563909774436 0.0 0.0 0.0 0.0 0.0150375939849624
        0.0 0.0 0.0 0.00751879699248120 0.0 0.00751879699248120 0.0 0.0
        0.00751879699248120 0.0150375939849624 0.00751879699248120 0.00751879699248120 0.00751879699248120 0.0
        0.0 0.00751879699248120 0.0451127819548872 0.0 0.0 0.0 0.0 0.00751879699248120
        0.00751879699248120 0.0 0.00751879699248120 0.00751879699248120 0.0601503759398496 0.00751879699248120
        0.0 0.00751879699248120 0.0 0.0 0.00751879699248120 0.00751879699248120 0.0 0.0
```

0.0 0.0 0.0075187969924812 0.0 0.0 0.0 0.0526315789473684 0.0150375939849624
0.0 0.0150375939849624 0.0676691729323308 0.0075187969924812 0.0 0.0 0.0
0.037593984962406 0.0 0.0676691729323308 0.0225563909774436 0.0 0.0 0.0 0.0
0.0150375939849624 0.0 0.0 0.0 0.0075187969924812 0.0 0.0075187969924812 0.0
0.0 0.0075187969924812 0.0150375939849624 0.0075187969924812 0.0075187969924812 0.0075187969924812
0.0 0.0 0.0075187969924812 0.0451127819548872 0.0 0.0 0.0 0.0
0.0075187969924812 0.0075187969924812 0.0 0.0075187969924812 0.0075187969924812 0.0601503759398496
0.0075187969924812 0.0 0.0075187969924812 0.0 0.0 0.0075187969924812 0.0075187969924812
0.0 0.0 0.0 0.0 0.0075187969924812 0.0 0.0 0.0 0.0526315789473684
0.0150375939849624 0.0 0.0150375939849624 0.0676691729323308 0.0075187969924812 0.0 0.0
0.0 0.037593984962406 0.0 0.0676691729323308 0.0225563909774436 0.0 0.0 0.0
0.0 0.0150375939849624 0.0 0.0 0.0 0.0075187969924812 0.0 0.0075187969924812
0.0 0.0 0.0075187969924812 0.0150375939849624 0.0075187969924812 0.0075187969924812
0.0075187969924812 0.0 0.0 0.0075187969924812
1 0 0
pattern 1687
0.0204081632653061 0.00510204081632653 0.0 0.0 0.0102040816326531 0.0
0.00510204081632653 0.0 0.0 0.00510204081632653 0.0714285714285714 0.0
0.0153061224489796 0.0102040816326531 0.0 0.00510204081632653 0.0 0.0153061224489796
0.00510204081632653 0.00510204081632653 0.0 0.0 0.0
0.00510204081632653 0.0 0.00510204081632653 0.0459183673469388
0.00510204081632653 0.0 0.0 0.0714285714285714 0.0 0.00510204081632653
0.0 0.0 0.0663265306122449 0.0 0.0510204081632653 0.0 0.00510204081632653
0.0 0.00510204081632653 0.0 0.00510204081632653 0.0
0.00510204081632653 0.0 0.0 0.0 0.0 0.0 0.0 0.0102040816326531
0.0 0.0102040816326531 0.00510204081632653 0.0 0.0 0.0
0.00510204081632653 0.0204081632653061 0.00510204081632653 0.0 0.0
0.0102040816326531 0.0 0.00510204081632653 0.0 0.0 0.00510204081632653
0.0714285714285714 0.0 0.0153061224489796 0.0102040816326531 0.0 0.00510204081632653
0.0 0.0153061224489796 0.00510204081632653 0.00510204081632653 0.0 0.0
0.0 0.00510204081632653 0.0 0.00510204081632653 0.0459183673469388
0.00510204081632653 0.0 0.0 0.0714285714285714 0.0 0.00510204081632653
0.0 0.0 0.0663265306122449 0.0 0.0510204081632653 0.0 0.00510204081632653
0.0 0.00510204081632653 0.0 0.00510204081632653 0.0
0.00510204081632653 0.0 0.0 0.0 0.0 0.0 0.0 0.0102040816326531
0.0 0.0102040816326531 0.00510204081632653 0.0 0.0 0.0
0.00510204081632653 0.0204081632653061 0.00510204081632653 0.0 0.0
0.0102040816326531 0.0 0.00510204081632653 0.0 0.0 0.00510204081632653
0.0714285714285714 0.0 0.0153061224489796 0.0102040816326531 0.0 0.00510204081632653
0.0 0.0153061224489796 0.00510204081632653 0.00510204081632653 0.0 0.0
0.0 0.00510204081632653 0.0 0.00510204081632653 0.0459183673469388
0.00510204081632653 0.0 0.0 0.0714285714285714 0.0 0.00510204081632653
0.0 0.0 0.0663265306122449 0.0 0.0510204081632653 0.0 0.00510204081632653
0.0 0.00510204081632653 0.0 0.00510204081632653 0.0
0.00510204081632653 0.0 0.0 0.0 0.0 0.0 0.0 0.0102040816326531
0.0 0.0102040816326531 0.00510204081632653 0.0 0.0 0.0
0.00510204081632653
1 0 0
pattern 1688
0.033175355450237 0.004739336492891 0.004739336492891 0.004739336492891 0.0 0.0 0.004739336492891
0.0 0.004739336492891 0.00947867298578199 0.0663507109004739 0.004739336492891 0.0
0.00947867298578199 0.0 0.0 0.0 0.018957345971564 0.0
0.00947867298578199 0.004739336492891 0.0 0.0 0.004739336492891 0.0
0.014218009478673 0.0616113744075829 0.00947867298578199 0.00947867298578199
0.004739336492891 0.0616113744075829 0.004739336492891 0.0 0.0 0.004739336492891
0.0758293838862559 0.0 0.0521327014218009 0.004739336492891 0.004739336492891 0.0
0.004739336492891 0.004739336492891 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.004739336492891 0.004739336492891 0.004739336492891 0.0 0.0 0.0
0.0 0.004739336492891 0.033175355450237 0.004739336492891 0.004739336492891 0.004739336492891
0.0 0.0 0.004739336492891 0.0 0.004739336492891 0.00947867298578199
0.0663507109004739 0.004739336492891 0.0 0.00947867298578199 0.0 0.0 0.0
0.018957345971564 0.0 0.00947867298578199 0.004739336492891 0.0 0.0
0.004739336492891 0.0 0.014218009478673 0.0616113744075829 0.00947867298578199
0.00947867298578199 0.004739336492891 0.0616113744075829 0.004739336492891 0.0 0.0
0.004739336492891 0.0758293838862559 0.0 0.0521327014218009 0.004739336492891 0.004739336492891
0.0 0.004739336492891 0.004739336492891 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.004739336492891 0.004739336492891 0.004739336492891 0.0 0.0
0.0 0.0 0.004739336492891 0.033175355450237 0.004739336492891 0.004739336492891
0.004739336492891 0.0 0.0 0.004739336492891 0.0 0.004739336492891
0.00947867298578199 0.0663507109004739 0.004739336492891 0.0 0.00947867298578199

0.0 0.0 0.0 0.018957345971564 0.0 0.00947867298578199 0.004739336492891
0.0 0.0 0.004739336492891 0.0 0.014218009478673 0.0616113744075829
0.00947867298578199 0.00947867298578199 0.004739336492891 0.0616113744075829
0.004739336492891 0.0 0.0 0.004739336492891 0.0758293838862559 0.0 0.0521327014218009
0.004739336492891 0.004739336492891 0.0 0.004739336492891 0.004739336492891 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.004739336492891 0.004739336492891
0.004739336492891 0.0 0.0 0.0 0.0 0.004739336492891
1 0 0
pattern 1689
0.0233644859813084 0.00467289719626168 0.0 0.0 0.00467289719626168
0.00467289719626168 0.00467289719626168 0.00467289719626168
0.00467289719626168 0.00467289719626168 0.0700934579439252 0.00467289719626168
0.0 0.0280373831775701 0.0 0.0 0.0 0.0233644859813084 0.0 0.0
0.00934579439252336 0.00934579439252336 0.0 0.00467289719626168 0.0
0.014018691588785 0.0373831775700935 0.00467289719626168 0.00467289719626168 0.0
0.0467289719626168 0.0 0.0186915887850467 0.0 0.0 0.088785046728972
0.00467289719626168 0.0514018691588785 0.00467289719626168 0.00467289719626168
0.00467289719626168 0.0 0.014018691588785 0.0 0.0 0.00934579439252336
0.00467289719626168 0.0 0.0 0.0 0.00467289719626168 0.0 0.0
0.00467289719626168 0.00934579439252336 0.0 0.0 0.0 0.0 0.0
0.0233644859813084 0.00467289719626168 0.0 0.0 0.00467289719626168
0.00467289719626168 0.00467289719626168 0.00467289719626168
0.00467289719626168 0.00467289719626168 0.0700934579439252 0.00467289719626168
0.0 0.0280373831775701 0.0 0.0 0.0 0.0233644859813084 0.0 0.0
0.00934579439252336 0.00934579439252336 0.0 0.00467289719626168 0.0
0.014018691588785 0.0373831775700935 0.00467289719626168 0.00467289719626168 0.0
0.0467289719626168 0.0 0.0186915887850467 0.0 0.0 0.088785046728972
0.00467289719626168 0.0514018691588785 0.00467289719626168 0.00467289719626168
0.00467289719626168 0.0 0.014018691588785 0.0 0.0 0.00934579439252336
0.00467289719626168 0.0 0.0 0.0 0.00467289719626168 0.0 0.0
0.00467289719626168 0.00934579439252336 0.0 0.0 0.0 0.0 0.0
0.0233644859813084 0.00467289719626168 0.0 0.0 0.00467289719626168
0.00467289719626168 0.00467289719626168 0.00467289719626168
0.00467289719626168 0.00467289719626168 0.0700934579439252 0.00467289719626168
0.0 0.0280373831775701 0.0 0.0 0.0 0.0233644859813084 0.0 0.0
0.00934579439252336 0.00934579439252336 0.0 0.00467289719626168 0.0
0.014018691588785 0.0373831775700935 0.00467289719626168 0.00467289719626168 0.0
0.0467289719626168 0.0 0.0186915887850467 0.0 0.0 0.088785046728972
0.00467289719626168 0.0514018691588785 0.00467289719626168 0.00467289719626168
0.00467289719626168 0.0 0.014018691588785 0.0 0.0 0.00934579439252336
0.00467289719626168 0.0 0.0 0.0 0.00467289719626168 0.0 0.0
0.00467289719626168 0.00934579439252336 0.0 0.0 0.0 0.0 0.0
1 0 0
pattern 1690
0.0136363636363636 0.0 0.0 0.00454545454545455 0.0 0.0 0.00909090909090909
0.0 0.00454545454545455 0.00454545454545455 0.0772727272727273
0.00454545454545455 0.00909090909090909 0.0181818181818182 0.0
0.00454545454545455 0.0 0.0181818181818182 0.0 0.00454545454545455 0.0
0.00454545454545455 0.0 0.0136363636363636 0.0 0.0136363636363636 0.0545454545454545
0.0 0.00909090909090909 0.0 0.05 0.00454545454545455 0.0 0.0
0.0 0.0863636363636364 0.00454545454545455 0.0545454545454545 0.0 0.0
0.00454545454545455 0.0 0.0 0.00909090909090909 0.00454545454545455
0.0 0.00454545454545455 0.0 0.0 0.0 0.0 0.0
0.00909090909090909 0.00909090909090909 0.00909090909090909 0.0 0.0
0.0 0.00454545454545455 0.00454545454545455 0.0136363636363636 0.0 0.0
0.00454545454545455 0.0 0.0 0.00909090909090909 0.0
0.00454545454545455 0.00454545454545455 0.0772727272727273 0.00454545454545455
0.00909090909090909 0.0181818181818182 0.0 0.00454545454545455 0.0
0.0181818181818182 0.0 0.00454545454545455 0.0 0.00454545454545455 0.0
0.0136363636363636 0.0 0.0136363636363636 0.0545454545454545 0.0 0.00909090909090909
0.0 0.05 0.00454545454545455 0.0 0.0 0.0 0.0863636363636364
0.00454545454545455 0.0545454545454545 0.0 0.0 0.00454545454545455 0.0
0.0 0.00909090909090909 0.00454545454545455 0.0 0.00454545454545455
0.0 0.0 0.0 0.0 0.0 0.00909090909090909 0.00909090909090909
0.00909090909090909 0.0 0.0 0.0 0.00454545454545455
0.00454545454545455 0.0136363636363636 0.0 0.0 0.00454545454545455 0.0
0.0 0.00909090909090909 0.0 0.00454545454545455 0.00454545454545455
0.0772727272727273 0.00454545454545455 0.00909090909090909 0.0181818181818182 0.0
0.00454545454545455 0.0 0.0181818181818182 0.0 0.00454545454545455 0.0
0.00454545454545455 0.0 0.0136363636363636 0.0 0.0136363636363636 0.0545454545454545

0.0 0.00909090909090909 0.0 0.05 0.00454545454545455 0.0 0.0
0.0 0.0863636363636364 0.00454545454545455 0.0545454545454545 0.0 0.0
0.00454545454545455 0.0 0.0 0.00909090909090909 0.00454545454545455
0.0 0.00454545454545455 0.0 0.0 0.0 0.0 0.0
0.00909090909090909 0.00909090909090909 0.00909090909090909 0.0 0.0
0.0 0.00454545454545455 0.00454545454545455
1 0 0
pattern 1691
0.0491803278688525 0.0109289617486339 0.0 0.0 0.0 0.00546448087431694
0.00546448087431694 0.0 0.00546448087431694 0.0109289617486339 0.0491803278688525
0.0109289617486339 0.0109289617486339 0.0218579234972678 0.0 0.0 0.0 0.0218579234972678
0.0 0.00546448087431694 0.00546448087431694 0.0 0.0 0.0 0.0
0.0 0.0655737704918033 0.00546448087431694 0.00546448087431694 0.0
0.0819672131147541 0.0 0.0 0.0 0.0163934426229508 0.0437158469945355 0.0
0.0765027322404372 0.0 0.0 0.0 0.00546448087431694 0.0 0.0 0.0
0.00546448087431694 0.00546448087431694 0.0 0.0 0.00546448087431694
0.0 0.00546448087431694 0.0109289617486339 0.00546448087431694
0.00546448087431694 0.0 0.0109289617486339 0.0109289617486339 0.0109289617486339
0.0109289617486339 0.0491803278688525 0.0109289617486339 0.0 0.0 0.0
0.00546448087431694 0.00546448087431694 0.0 0.00546448087431694
0.0109289617486339 0.0491803278688525 0.0109289617486339 0.0109289617486339 0.0218579234972678 0.0
0.0 0.0 0.0218579234972678 0.0 0.00546448087431694 0.00546448087431694
0.0 0.0 0.0 0.0 0.0 0.0655737704918033 0.00546448087431694
0.00546448087431694 0.0 0.0819672131147541 0.0 0.0 0.0 0.0163934426229508
0.0437158469945355 0.0 0.0765027322404372 0.0 0.0 0.0 0.00546448087431694
0.0 0.0 0.0 0.00546448087431694 0.00546448087431694 0.0 0.0
0.00546448087431694 0.0 0.00546448087431694 0.0109289617486339
0.00546448087431694 0.00546448087431694 0.0 0.0109289617486339 0.0109289617486339
0.0109289617486339 0.0109289617486339 0.0491803278688525 0.0109289617486339 0.0 0.0 0.0
0.00546448087431694 0.00546448087431694 0.0 0.00546448087431694
0.0109289617486339 0.0491803278688525 0.0109289617486339 0.0109289617486339 0.0218579234972678 0.0
0.0 0.0 0.0218579234972678 0.0 0.00546448087431694 0.00546448087431694
0.0 0.0 0.0 0.0 0.0 0.0655737704918033 0.00546448087431694
0.00546448087431694 0.0 0.0819672131147541 0.0 0.0 0.0 0.0163934426229508
0.0437158469945355 0.0 0.0765027322404372 0.0 0.0 0.0 0.00546448087431694
0.0 0.0 0.0 0.00546448087431694 0.00546448087431694 0.0 0.0
0.00546448087431694 0.0 0.00546448087431694 0.0109289617486339
0.00546448087431694 0.00546448087431694 0.0 0.0109289617486339 0.0109289617486339
0.0109289617486339 0.0109289617486339
1 0 0
pattern 1692
0.0478723404255319 0.0106382978723404 0.00531914893617021 0.0 0.0 0.00531914893617021
0.00531914893617021 0.0 0.0106382978723404 0.00531914893617021 0.0585106382978723
0.00531914893617021 0.00531914893617021 0.0212765957446809 0.0
0.00531914893617021 0.0 0.0106382978723404 0.0 0.00531914893617021
0.0106382978723404 0.0 0.0 0.00531914893617021 0.0 0.0 0.0585106382978723
0.00531914893617021 0.0106382978723404 0.00531914893617021 0.0531914893617021 0.0
0.0 0.0 0.00531914893617021 0.0478723404255319 0.0 0.0691489361702128 0.0
0.0 0.0 0.00531914893617021 0.0106382978723404 0.0 0.00531914893617021
0.00531914893617021 0.00531914893617021 0.0 0.00531914893617021
0.00531914893617021 0.0 0.0 0.0 0.0 0.0 0.0106382978723404
0.00531914893617021 0.00531914893617021 0.0106382978723404 0.0478723404255319
0.0106382978723404 0.00531914893617021 0.0 0.0 0.00531914893617021
0.00531914893617021 0.0 0.0106382978723404 0.00531914893617021 0.0585106382978723
0.00531914893617021 0.00531914893617021 0.0212765957446809 0.0
0.00531914893617021 0.0 0.0106382978723404 0.0 0.00531914893617021
0.0106382978723404 0.0 0.0 0.00531914893617021 0.0 0.0 0.0585106382978723
0.00531914893617021 0.0106382978723404 0.00531914893617021 0.0531914893617021 0.0
0.0 0.0 0.00531914893617021 0.0478723404255319 0.0 0.0691489361702128 0.0
0.0 0.0 0.00531914893617021 0.0106382978723404 0.0 0.00531914893617021
0.00531914893617021 0.00531914893617021 0.0 0.00531914893617021
0.00531914893617021 0.0 0.0 0.0 0.0 0.0 0.0106382978723404
0.00531914893617021 0.00531914893617021 0.0106382978723404 0.0478723404255319
0.0106382978723404 0.00531914893617021 0.0 0.0 0.00531914893617021
0.00531914893617021 0.0 0.0106382978723404 0.00531914893617021 0.0585106382978723
0.00531914893617021 0.00531914893617021 0.0212765957446809 0.0
0.00531914893617021 0.0 0.0106382978723404 0.0 0.00531914893617021
0.0106382978723404 0.0 0.0 0.00531914893617021 0.0 0.0 0.0585106382978723
0.00531914893617021 0.0106382978723404 0.00531914893617021 0.0531914893617021 0.0
0.0 0.0 0.00531914893617021 0.0478723404255319 0.0 0.0691489361702128 0.0

0.0 0.0 0.00531914893617021 0.0106382978723404 0.0 0.00531914893617021
0.00531914893617021 0.00531914893617021 0.0 0.00531914893617021
0.00531914893617021 0.0 0.0 0.0 0.0 0.0 0.0 0.0106382978723404
0.00531914893617021 0.00531914893617021 0.0106382978723404
1 0 0
pattern 1693
0.0109289617486339 0.0 0.0 0.0109289617486339 0.0 0.0 0.00546448087431694
0.00546448087431694 0.00546448087431694 0.00546448087431694 0.0655737704918033
0.00546448087431694 0.0 0.0218579234972678 0.0 0.0109289617486339
0.00546448087431694 0.0163934426229508 0.0 0.0 0.0109289617486339
0.00546448087431694 0.0 0.00546448087431694 0.0 0.0 0.0382513661202186
0.00546448087431694 0.0 0.0 0.0382513661202186 0.0 0.00546448087431694
0.0 0.00546448087431694 0.0491803278688525 0.00546448087431694 0.087431693989071
0.0 0.0 0.00546448087431694 0.0 0.0 0.0 0.00546448087431694
0.0 0.0163934426229508 0.0 0.0 0.0 0.0 0.0
0.00546448087431694 0.0109289617486339 0.00546448087431694 0.00546448087431694
0.0 0.0109289617486339 0.0 0.0109289617486339 0.0 0.0 0.0109289617486339 0.0
0.0 0.00546448087431694 0.00546448087431694 0.00546448087431694
0.00546448087431694 0.0655737704918033 0.00546448087431694 0.0 0.0218579234972678
0.0 0.0109289617486339 0.00546448087431694 0.0163934426229508 0.0 0.0
0.0109289617486339 0.00546448087431694 0.0 0.00546448087431694 0.0 0.0
0.0382513661202186 0.00546448087431694 0.0 0.0 0.0382513661202186 0.0
0.00546448087431694 0.0 0.00546448087431694 0.0491803278688525
0.00546448087431694 0.087431693989071 0.0 0.0 0.00546448087431694 0.0
0.0 0.0 0.00546448087431694 0.0 0.0163934426229508 0.0 0.0 0.0
0.0 0.0 0.0 0.00546448087431694 0.0109289617486339 0.00546448087431694
0.00546448087431694 0.0 0.0109289617486339 0.0 0.0109289617486339 0.0 0.0
0.0109289617486339 0.0 0.0 0.00546448087431694 0.00546448087431694
0.00546448087431694 0.00546448087431694 0.0655737704918033 0.00546448087431694
0.0 0.0218579234972678 0.0 0.0109289617486339 0.00546448087431694 0.0163934426229508
0.0 0.0 0.0109289617486339 0.00546448087431694 0.0 0.00546448087431694
0.0 0.0 0.0382513661202186 0.00546448087431694 0.0 0.0 0.0382513661202186
0.0 0.00546448087431694 0.0 0.00546448087431694 0.0491803278688525
0.00546448087431694 0.087431693989071 0.0 0.0 0.00546448087431694 0.0
0.0 0.0 0.00546448087431694 0.0 0.0163934426229508 0.0 0.0 0.0
0.0 0.0 0.0 0.00546448087431694 0.0109289617486339 0.00546448087431694
0.00546448087431694 0.0 0.0109289617486339 0.0
1 0 0

Appendix C: Part of Testing Set of Single Network Approach

SNNS pattern definition file V4.2
generated at Mon Nov 15 14:30:51 2003

No. of patterns: 1356
No. of input units: 180
No. of output units: 3

pattern 1
0.0402843601895735 0.0023696682464455 0.0023696682464455 0.0023696682464455 0.004739336492891 0.0
0.004739336492891 0.0 0.00947867298578199 0.0023696682464455 0.0758293838862559
0.004739336492891 0.014218009478673 0.023696682464455 0.0023696682464455 0.0023696682464455 0.0
0.023696682464455 0.004739336492891 0.0023696682464455 0.0 0.00710900473933649 0.0
0.0 0.0 0.0023696682464455 0.0687203791469194 0.00710900473933649 0.004739336492891
0.0 0.0687203791469194 0.0 0.00710900473933649 0.0023696682464455 0.0023696682464455
0.0663507109004739 0.0023696682464455 0.0663507109004739 0.00710900473933649 0.0
0.0023696682464455 0.0023696682464455 0.0023696682464455 0.004739336492891 0.0 0.0023696682464455
0.00710900473933649 0.0023696682464455 0.0023696682464455 0.0 0.0 0.0
0.0023696682464455 0.0023696682464455 0.00710900473933649 0.0 0.0118483412322275
0.0118483412322275 0.004739336492891 0.0 0.0402843601895735 0.0023696682464455 0.0023696682464455
0.0023696682464455 0.004739336492891 0.0 0.004739336492891 0.0 0.00947867298578199
0.0023696682464455 0.0758293838862559 0.004739336492891 0.014218009478673 0.023696682464455
0.0023696682464455 0.0023696682464455 0.0 0.023696682464455 0.004739336492891 0.0023696682464455
0.0 0.00710900473933649 0.0 0.0 0.0023696682464455 0.0687203791469194
0.00710900473933649 0.004739336492891 0.0 0.0687203791469194 0.0
0.00710900473933649 0.0023696682464455 0.0023696682464455 0.0663507109004739 0.0023696682464455
0.0663507109004739 0.00710900473933649 0.0 0.0023696682464455 0.0023696682464455
0.0023696682464455 0.004739336492891 0.0 0.0023696682464455 0.00710900473933649
0.0023696682464455 0.0023696682464455 0.0 0.0 0.0 0.0023696682464455 0.0023696682464455
0.00710900473933649 0.0 0.0118483412322275 0.0118483412322275 0.004739336492891 0.0
0.0402843601895735 0.0023696682464455 0.0023696682464455 0.0023696682464455 0.004739336492891 0.0
0.004739336492891 0.0 0.00947867298578199 0.0023696682464455 0.0758293838862559
0.004739336492891 0.014218009478673 0.023696682464455 0.0023696682464455 0.0023696682464455 0.0
0.023696682464455 0.004739336492891 0.0023696682464455 0.0 0.00710900473933649 0.0
0.0 0.0 0.0023696682464455 0.0687203791469194 0.00710900473933649 0.004739336492891
0.0 0.0687203791469194 0.0 0.00710900473933649 0.0023696682464455 0.0023696682464455
0.0663507109004739 0.0023696682464455 0.0663507109004739 0.00710900473933649 0.0
0.0023696682464455 0.0023696682464455 0.0023696682464455 0.004739336492891 0.0 0.0023696682464455
0.00710900473933649 0.0023696682464455 0.0023696682464455 0.0 0.0 0.0
0.0023696682464455 0.0023696682464455 0.00710900473933649 0.0 0.0118483412322275
0.0118483412322275 0.004739336492891 0.0
0 0 1
pattern 2
0.0272108843537415 0.0 0.00680272108843537 0.0 0.0 0.0 0.0
0.00680272108843537 0.0 0.0 0.0612244897959184 0.0 0.0 0.0204081632653061
0.0 0.00680272108843537 0.0 0.0204081632653061 0.00680272108843537
0.0136054421768707 0.0 0.00680272108843537 0.0 0.00680272108843537 0.0
0.0 0.0476190476190476 0.0 0.0 0.0 0.054421768707483 0.00680272108843537
0.0136054421768707 0.0 0.0 0.0680272108843537 0.0 0.054421768707483 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.00680272108843537 0.0136054421768707 0.0136054421768707
0.00680272108843537 0.0 0.0 0.0 0.0272108843537415 0.0
0.00680272108843537 0.0 0.0 0.0 0.0 0.00680272108843537 0.0
0.0 0.0612244897959184 0.0 0.0 0.0204081632653061 0.0 0.00680272108843537
0.0 0.0204081632653061 0.00680272108843537 0.0136054421768707 0.0
0.00680272108843537 0.0 0.00680272108843537 0.0 0.0 0.0476190476190476
0.0 0.0 0.0 0.054421768707483 0.00680272108843537 0.0136054421768707 0.0
0.0 0.0680272108843537 0.0 0.054421768707483 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.00680272108843537 0.0136054421768707 0.0136054421768707 0.00680272108843537 0.0
0.0 0.0 0.0272108843537415 0.0 0.00680272108843537 0.0 0.0 0.0
0.0 0.00680272108843537 0.0 0.0 0.0612244897959184 0.0 0.0
0.0204081632653061 0.0 0.00680272108843537 0.0 0.0204081632653061
0.00680272108843537 0.0136054421768707 0.0 0.00680272108843537 0.0
0.00680272108843537 0.0 0.0 0.0476190476190476 0.0 0.0 0.0
0.054421768707483 0.00680272108843537 0.0136054421768707 0.0 0.0 0.0680272108843537
0.0 0.054421768707483 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

```
        0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.00680272108843537
        0.0136054421768707 0.0136054421768707 0.00680272108843537 0.0 0.0 0.0
0       0       1
# pattern 3
0.0206896551724138 0.0  0.0  0.0137931034482759 0.0  0.00689655172413793  0.0  0.0
        0.0  0.0  0.0896551724137931 0.0206896551724138 0.0  0.00689655172413793  0.0
        0.00689655172413793  0.0  0.0137931034482759 0.0  0.00689655172413793  0.0
        0.0  0.0  0.00689655172413793  0.0  0.0137931034482759 0.0551724137931034
        0.00689655172413793  0.0  0.00689655172413793  0.0413793103448276 0.0  0.0
        0.00689655172413793  0.00689655172413793  0.0827586206896552 0.0  0.0551724137931034
        0.0  0.0  0.0137931034482759 0.0  0.00689655172413793  0.0206896551724138 0.0
        0.0  0.0  0.0  0.00689655172413793  0.0  0.0  0.0  0.0137931034482759
        0.0  0.0137931034482759 0.0  0.00689655172413793  0.0  0.0  0.0
        0.0206896551724138 0.0  0.0  0.0137931034482759 0.0  0.00689655172413793  0.0
        0.0  0.0  0.0  0.0896551724137931 0.0206896551724138 0.0  0.00689655172413793
        0.0  0.00689655172413793  0.0  0.0137931034482759 0.0  0.00689655172413793
        0.0  0.0  0.0  0.00689655172413793  0.0  0.0137931034482759 0.0551724137931034
        0.00689655172413793  0.0  0.00689655172413793  0.0413793103448276 0.0  0.0
        0.00689655172413793  0.00689655172413793  0.0827586206896552 0.0  0.0551724137931034
        0.0  0.0  0.0137931034482759 0.0  0.00689655172413793  0.0206896551724138 0.0
        0.0  0.0  0.0  0.00689655172413793  0.0  0.0  0.0  0.0137931034482759
        0.0  0.0137931034482759 0.0  0.00689655172413793  0.0  0.0  0.0
        0.0206896551724138 0.0  0.0  0.0137931034482759 0.0  0.00689655172413793  0.0
        0.0  0.0  0.0  0.0896551724137931 0.0206896551724138 0.0  0.00689655172413793
        0.0  0.00689655172413793  0.0  0.0137931034482759 0.0  0.00689655172413793
        0.0  0.0  0.0  0.00689655172413793  0.0  0.0137931034482759 0.0551724137931034
        0.00689655172413793  0.0  0.00689655172413793  0.0413793103448276 0.0  0.0
        0.00689655172413793  0.00689655172413793  0.0827586206896552 0.0  0.0551724137931034
        0.0  0.0  0.0137931034482759 0.0  0.00689655172413793  0.0206896551724138 0.0
        0.0  0.0  0.0  0.00689655172413793  0.0  0.0  0.0  0.0137931034482759
        0.0  0.0137931034482759 0.0  0.00689655172413793  0.0  0.0  0.0
0       0       1
# pattern 4
0.0142857142857143 0.0  0.0  0.0142857142857143 0.0428571428571429 0.0  0.0
        0.00714285714285714  0.0  0.0  0.1  0.0214285714285714 0.0  0.0142857142857143
        0.0  0.0  0.0142857142857143 0.0142857142857143 0.0  0.0214285714285714 0.0  0.0
        0.0142857142857143 0.00714285714285714  0.0  0.0  0.0357142857142857 0.0  0.0
        0.0  0.0642857142857143 0.00714285714285714  0.00714285714285714
        0.00714285714285714  0.0  0.0785714285714286 0.00714285714285714  0.0428571428571429
        0.0  0.00714285714285714  0.0285714285714286 0.0  0.0  0.0142857142857143 0.0
        0.0  0.0  0.00714285714285714  0.00714285714285714  0.0  0.0  0.0
        0.00714285714285714  0.0  0.0  0.0  0.0  0.0  0.0  0.0
        0.0142857142857143 0.0  0.0  0.0142857142857143 0.0428571428571429 0.0  0.0
        0.00714285714285714  0.0  0.0  0.1  0.0214285714285714 0.0  0.0142857142857143
        0.0  0.0  0.0142857142857143 0.0142857142857143 0.0  0.0214285714285714 0.0  0.0
        0.0142857142857143 0.00714285714285714  0.0  0.0  0.0357142857142857 0.0  0.0
        0.0  0.0642857142857143 0.00714285714285714  0.00714285714285714
        0.00714285714285714  0.0  0.0785714285714286 0.00714285714285714  0.0428571428571429
        0.0  0.00714285714285714  0.0285714285714286 0.0  0.0  0.0142857142857143 0.0
        0.0  0.0  0.00714285714285714  0.00714285714285714  0.0  0.0  0.0
        0.00714285714285714  0.0  0.0  0.0  0.0  0.0  0.0  0.0
        0.0142857142857143 0.0  0.0  0.0142857142857143 0.0428571428571429 0.0  0.0
        0.00714285714285714  0.0  0.0  0.1  0.0214285714285714 0.0  0.0142857142857143
        0.0  0.0  0.0142857142857143 0.0142857142857143 0.0  0.0214285714285714 0.0  0.0
        0.0142857142857143 0.00714285714285714  0.0  0.0  0.0357142857142857 0.0  0.0
        0.0  0.0642857142857143 0.00714285714285714  0.00714285714285714
        0.00714285714285714  0.0  0.0785714285714286 0.00714285714285714  0.0428571428571429
        0.0  0.00714285714285714  0.0285714285714286 0.0  0.0  0.0142857142857143 0.0
        0.0  0.0  0.00714285714285714  0.00714285714285714  0.0  0.0  0.0
        0.00714285714285714  0.0  0.0  0.0  0.0  0.0  0.0  0.0
0       0       1
# pattern 5
0.0275862068965517 0.0  0.0206896551724138 0.0137931034482759 0.0137931034482759 0.0  0.0
        0.0137931034482759 0.0  0.0  0.0758620689655172 0.0137931034482759 0.0  0.0
        0.00689655172413793  0.0  0.00689655172413793  0.0137931034482759
        0.00689655172413793  0.0137931034482759 0.0  0.0  0.00689655172413793  0.0
        0.00689655172413793  0.00689655172413793  0.0206896551724138 0.0
        0.00689655172413793  0.0  0.0551724137931034 0.0  0.0137931034482759 0.0
        0.00689655172413793  0.0827586206896552 0.0  0.0206896551724138 0.00689655172413793
        0.0  0.00689655172413793  0.0137931034482759 0.0  0.00689655172413793  0.0
```

0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0275862068965517 0.0 0.0206896551724138 0.0137931034482759
0.0137931034482759 0.0 0.0 0.0137931034482759 0.0 0.0 0.0758620689655172
0.0137931034482759 0.0 0.0 0.00689655172413793 0.0 0.00689655172413793
0.0137931034482759 0.00689655172413793 0.0137931034482759 0.0 0.0
0.00689655172413793 0.0 0.00689655172413793 0.00689655172413793
0.0206896551724138 0.0 0.00689655172413793 0.0 0.0551724137931034 0.0
0.0137931034482759 0.0 0.00689655172413793 0.0827586206896552 0.0 0.0206896551724138
0.00689655172413793 0.0 0.00689655172413793 0.0137931034482759 0.0
0.00689655172413793 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0275862068965517 0.0
0.0206896551724138 0.0137931034482759 0.0137931034482759 0.0 0.0 0.0137931034482759 0.0
0.0 0.0758620689655172 0.0137931034482759 0.0 0.0 0.00689655172413793 0.0
0.00689655172413793 0.0137931034482759 0.00689655172413793 0.0137931034482759 0.0
0.0 0.00689655172413793 0.0 0.00689655172413793 0.00689655172413793
0.0206896551724138 0.0 0.00689655172413793 0.0 0.0551724137931034 0.0
0.0137931034482759 0.0 0.00689655172413793 0.0827586206896552 0.0 0.0206896551724138
0.00689655172413793 0.0 0.00689655172413793 0.0137931034482759 0.0
0.00689655172413793 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0 0 1
pattern 6
0.00714285714285714 0.0 0.00714285714285714 0.00714285714285714 0.0214285714285714
0.0 0.0 0.00714285714285714 0.0 0.0 0.0785714285714286 0.0142857142857143
0.0 0.0142857142857143 0.0 0.00714285714285714 0.0142857142857143 0.0142857142857143
0.0 0.0214285714285714 0.0 0.0 0.0214285714285714 0.0 0.0
0.00714285714285714 0.0214285714285714 0.0 0.0 0.0 0.05 0.0
0.00714285714285714 0.00714285714285714 0.0 0.0785714285714286 0.0 0.05
0.0 0.0 0.00714285714285714 0.0 0.0 0.0142857142857143
0.00714285714285714 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.00714285714285714 0.00714285714285714 0.00714285714285714 0.0 0.0
0.0 0.0 0.0 0.00714285714285714 0.0 0.00714285714285714
0.00714285714285714 0.0214285714285714 0.0 0.0 0.00714285714285714 0.0
0.0 0.0785714285714286 0.0142857142857143 0.0 0.0142857142857143 0.0
0.00714285714285714 0.0142857142857143 0.0142857142857143 0.0 0.0214285714285714 0.0
0.0 0.0214285714285714 0.0 0.0 0.00714285714285714 0.0214285714285714 0.0
0.0 0.0 0.05 0.0 0.00714285714285714 0.00714285714285714 0.0
0.0785714285714286 0.0 0.05 0.0 0.0 0.00714285714285714 0.0 0.0
0.0142857142857143 0.00714285714285714 0.0 0.0 0.0 0.0 0.0
0.0 0.00714285714285714 0.00714285714285714 0.00714285714285714 0.0
0.0 0.0 0.0 0.0 0.00714285714285714 0.0 0.00714285714285714
0.00714285714285714 0.0214285714285714 0.0 0.0 0.00714285714285714 0.0
0.0 0.0785714285714286 0.0142857142857143 0.0 0.0142857142857143 0.0
0.00714285714285714 0.0142857142857143 0.0142857142857143 0.0 0.0214285714285714 0.0
0.0 0.0214285714285714 0.0 0.0 0.00714285714285714 0.0214285714285714 0.0
0.0 0.0 0.05 0.0 0.00714285714285714 0.00714285714285714 0.0
0.0785714285714286 0.0 0.05 0.0 0.0 0.00714285714285714 0.0 0.0
0.0142857142857143 0.00714285714285714 0.0 0.0 0.0 0.0 0.0
0.0 0.00714285714285714 0.00714285714285714 0.00714285714285714 0.0
0.0 0.0 0.0 0.0
0 0 1
.
.
.
pattern 1353
0.0140845070422535 0.00469483568075117 0.0 0.00938967136150235 0.00469483568075117
0.00469483568075117 0.00469483568075117 0.0 0.0 0.00938967136150235
0.0657276995305164 0.00938967136150235 0.0 0.0140845070422535 0.0
0.00469483568075117 0.0 0.0234741784037559 0.00469483568075117
0.00938967136150235 0.0 0.0 0.0 0.00469483568075117 0.0
0.00938967136150235 0.0516431924882629 0.0140845070422535 0.0 0.0 0.0657276995305164
0.0 0.00469483568075117 0.0 0.0 0.0845070422535211 0.00469483568075117
0.0563380281690141 0.00469483568075117 0.00469483568075117 0.0140845070422535 0.0
0.0140845070422535 0.0 0.0 0.00469483568075117 0.0 0.00938967136150235
0.0 0.0 0.0 0.0 0.0140845070422535 0.00938967136150235
0.00469483568075117 0.00938967136150235 0.0 0.00469483568075117 0.0
0.0 0.0140845070422535 0.00469483568075117 0.0 0.00938967136150235
0.00469483568075117 0.00469483568075117 0.00469483568075117 0.0 0.0
0.00938967136150235 0.0657276995305164 0.00938967136150235 0.0 0.0140845070422535
0.0 0.00469483568075117 0.0 0.0234741784037559 0.00469483568075117

```
0.00938967136150235 0.0 0.0 0.0 0.00469483568075117 0.0
0.00938967136150235 0.0516431924882629 0.0140845070422535 0.0 0.0 0.0657276995305164
0.0 0.00469483568075117 0.0 0.0 0.0845070422535211 0.00469483568075117
0.0563380281690141 0.00469483568075117 0.00469483568075117 0.0140845070422535 0.0
0.0140845070422535 0.0 0.0 0.00469483568075117 0.0 0.00938967136150235
0.0 0.0 0.0 0.0 0.0140845070422535 0.00938967136150235
0.00469483568075117 0.00938967136150235 0.0 0.00469483568075117 0.0
0.0 0.0140845070422535 0.00469483568075117 0.0 0.00938967136150235
0.00469483568075117 0.00469483568075117 0.00469483568075117 0.0 0.0
0.00938967136150235 0.0657276995305164 0.00938967136150235 0.0 0.0140845070422535
0.0 0.00469483568075117 0.0 0.0234741784037559 0.00469483568075117
0.00938967136150235 0.0 0.0 0.0 0.00469483568075117 0.0
0.00938967136150235 0.0516431924882629 0.0140845070422535 0.0 0.0 0.0657276995305164
0.0 0.00469483568075117 0.0 0.0 0.0845070422535211 0.00469483568075117
0.0563380281690141 0.00469483568075117 0.00469483568075117 0.0140845070422535 0.0
0.0140845070422535 0.0 0.0 0.00469483568075117 0.0 0.00938967136150235
0.0 0.0 0.0 0.0 0.0140845070422535 0.00938967136150235
0.00469483568075117 0.00938967136150235 0.0 0.00469483568075117 0.0
0.0
1 0 0
# pattern 1354
0.037037037037037 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.037037037037037 0.037037037037037 0.0 0.0 0.0 0.037037037037037 0.0 0.0
0.0 0.0 0.037037037037037 0.0 0.0 0.037037037037037 0.0 0.0
0.148148148148148 0.0 0.037037037037037 0.0 0.0740740740740741 0.0 0.0 0.0
0.037037037037037 0.037037037037037 0.0 0.111111111111111 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.037037037037037 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.037037037037037 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.037037037037037
0.037037037037037 0.0 0.0 0.0 0.037037037037037 0.0 0.0 0.0 0.0
0.037037037037037 0.0 0.0 0.037037037037037 0.0 0.0 0.148148148148148 0.0
0.037037037037037 0.0 0.0740740740740741 0.0 0.0 0.0 0.037037037037037
0.037037037037037 0.0 0.111111111111111 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.037037037037037 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.037037037037037 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.037037037037037 0.037037037037037 0.0
0.0 0.0 0.037037037037037 0.0 0.0 0.0 0.0 0.037037037037037 0.0
0.0 0.037037037037037 0.0 0.0 0.148148148148148 0.0 0.037037037037037 0.0
0.0740740740740741 0.0 0.0 0.0 0.037037037037037 0.037037037037037 0.0
0.111111111111111 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.037037037037037 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0
1 0 0
# pattern 1355
0.0247524752475248 0.0 0.00495049504950495 0.00495049504950495 0.0 0.0
0.0099009900990099 0.0 0.00495049504950495 0.0099009900990099 0.0841584158415842 0.0
0.00495049504950495 0.0247524752475248 0.0 0.00495049504950495 0.0099009900990099
0.0148514851485149 0.0 0.0 0.0099009900990099 0.00495049504950495 0.0
0.0099009900990099 0.0 0.0099009900990099 0.0346534653465347 0.00495049504950495
0.00495049504950495 0.0 0.0495049504950495 0.0 0.0 0.00495049504950495
0.0 0.0693069306930693 0.0 0.0643564356435644 0.0 0.00495049504950495 0.0
0.0 0.0 0.00495049504950495 0.0 0.0 0.0 0.00495049504950495
0.0 0.0 0.0 0.0 0.00495049504950495 0.00495049504950495
0.00495049504950495 0.0 0.0 0.0 0.0 0.00495049504950495
0.0247524752475248 0.0 0.00495049504950495 0.00495049504950495 0.0 0.0
0.0099009900990099 0.0 0.00495049504950495 0.0099009900990099 0.0841584158415842 0.0
0.00495049504950495 0.0247524752475248 0.0 0.00495049504950495 0.0099009900990099
0.0148514851485149 0.0 0.0 0.0099009900990099 0.00495049504950495 0.0
0.0099009900990099 0.0 0.0099009900990099 0.0346534653465347 0.00495049504950495
0.00495049504950495 0.0 0.0495049504950495 0.0 0.0 0.00495049504950495
0.0 0.0693069306930693 0.0 0.0643564356435644 0.0 0.00495049504950495 0.0
0.0 0.0 0.00495049504950495 0.0 0.0 0.0 0.00495049504950495
0.0 0.0 0.0 0.0 0.00495049504950495 0.00495049504950495
0.00495049504950495 0.0 0.0 0.0 0.0 0.00495049504950495
0.0247524752475248 0.0 0.00495049504950495 0.00495049504950495 0.0 0.0
0.0099009900990099 0.0 0.00495049504950495 0.0099009900990099 0.0841584158415842 0.0
0.00495049504950495 0.0247524752475248 0.0 0.00495049504950495 0.0099009900990099
0.0148514851485149 0.0 0.0 0.0099009900990099 0.00495049504950495 0.0
0.0099009900990099 0.0 0.0099009900990099 0.0346534653465347 0.00495049504950495
0.00495049504950495 0.0 0.0495049504950495 0.0 0.0 0.00495049504950495
```

```
        0.0     0.0693069306930693 0.0     0.0643564356435644 0.0     0.00495049504950495 0.0
        0.0     0.0     0.00495049504950495 0.0     0.0     0.0     0.00495049504950495
        0.0     0.0     0.0     0.0     0.00495049504950495 0.00495049504950495
        0.00495049504950495 0.0     0.0     0.0     0.0     0.00495049504950495
1       0       0
# pattern 1356
0.0471698113207547 0.0188679245283019 0.0 0.00471698113207547 0.0 0.00471698113207547
        0.00943396226415094 0.0 0.00943396226415094 0.0141509433962264 0.0518867924528302
        0.00943396226415094 0.00943396226415094 0.0188679245283019 0.0
        0.00471698113207547 0.0 0.00943396226415094 0.0 0.0
        0.00471698113207547 0.00471698113207547 0.0 0.00943396226415094 0.0
        0.0 0.0849056603773585 0.00943396226415094 0.00943396226415094 0.0
        0.0566037735849057 0.0 0.0 0.00471698113207547 0.00943396226415094
        0.0566037735849057 0.0 0.0566037735849057 0.0 0.0 0.00471698113207547
        0.00471698113207547 0.00471698113207547 0.00471698113207547
        0.00471698113207547 0.0 0.00471698113207547 0.0 0.0
        0.00471698113207547 0.0 0.0 0.00943396226415094 0.0 0.0 0.0
        0.00471698113207547 0.0 0.0 0.0 0.0471698113207547 0.0188679245283019 0.0
        0.00471698113207547 0.0 0.00471698113207547 0.00943396226415094 0.0
        0.00943396226415094 0.0141509433962264 0.0518867924528302 0.00943396226415094
        0.00943396226415094 0.0188679245283019 0.0 0.00471698113207547 0.0
        0.00943396226415094 0.0 0.0 0.00471698113207547 0.00471698113207547
        0.0 0.00943396226415094 0.0 0.0 0.0849056603773585 0.00943396226415094
        0.00943396226415094 0.0 0.0566037735849057 0.0 0.0 0.00471698113207547
        0.00943396226415094 0.0566037735849057 0.0 0.0566037735849057 0.0 0.0
        0.00471698113207547 0.00471698113207547 0.00471698113207547
        0.00471698113207547 0.00471698113207547 0.0 0.00471698113207547 0.0
        0.0 0.00471698113207547 0.0 0.0 0.00943396226415094 0.0 0.0
        0.0 0.00471698113207547 0.0 0.0 0.0 0.0471698113207547 0.0188679245283019
        0.0 0.00471698113207547 0.0 0.00471698113207547 0.00943396226415094
        0.0 0.00943396226415094 0.0141509433962264 0.0518867924528302 0.00943396226415094
        0.00943396226415094 0.0188679245283019 0.0 0.00471698113207547 0.0
        0.00943396226415094 0.0 0.0 0.00471698113207547 0.00471698113207547
        0.0 0.00943396226415094 0.0 0.0 0.0849056603773585 0.00943396226415094
        0.00943396226415094 0.0 0.0566037735849057 0.0 0.0 0.00471698113207547
        0.00943396226415094 0.0566037735849057 0.0 0.0566037735849057 0.0 0.0
        0.00471698113207547 0.00471698113207547 0.00471698113207547
        0.00471698113207547 0.00471698113207547 0.0 0.00471698113207547 0.0
        0.0 0.00471698113207547 0.0 0.0 0.00943396226415094 0.0 0.0
        0.0 0.00471698113207547 0.0 0.0 0.0
1       0       0
```

Appendix D: Part of Prediction Result of Single Network Approach

SNNS result file V1.4-3D
generated at Tue Dec 16 22:57:06 2003

No. of patterns : 1356
No. of input units : 180
No. of output units : 3
startpattern : 1
endpattern : 1356
teaching output included
#1.1
0 0 1
0.99872 0.08805 0
#2.1
0 0 1
0.85882 0.10733 0.00001
#3.1
0 0 1
0.99965 0.08344 0
#4.1
0 0 1
0.00006 0.16891 0.69951
#5.1
0 0 1
0.00027 0.15903 0.28121
#6.1
0 0 1
0.00028 0.15886 0.27485
#7.1
0 0 1
0.00041 0.15652 0.19682
#8.1
0 0 1
0.00024 0.1597 0.30669
#9.1
0 0 1
0.00005 0.16931 0.71414
.
.
.

#1349.1
1 0 0
0.99938 0.08547 0
#1350.1
1 0 0
0.99873 0.08801 0
#1351.1
1 0 0
0.97487 0.09956 0
#1352.1
1 0 0
0.99805 0.08959 0
#1353.1
1 0 0
1 0.06897 0
#1354.1
1 0 0
1 0.04662 0
#1355.1
1 0 0
0.99995 0.07713 0
#1356.1
1 0 0
1 0.06464 0

Appendix E: Figure of Performance of Single Network Approach

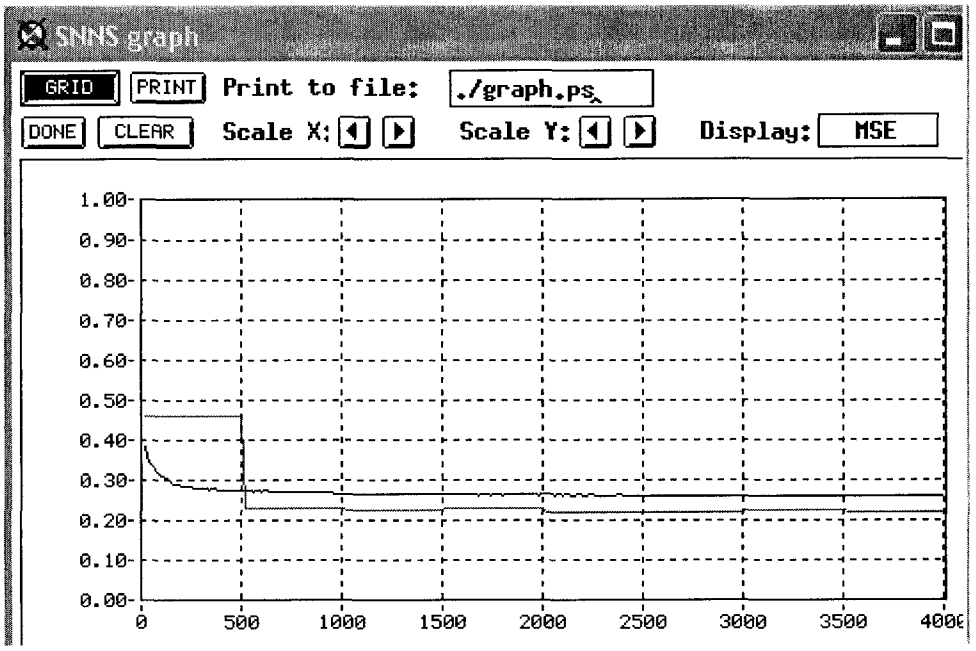


Figure Appendix E: *The performance of Single Network Approach*

Appendix F: Part of Training Set of Pair-wise Binary Classification Approach

SNNS pattern definition file V4.2
generated at Sun Feb 2 19:08:37 2004

No. of patterns: 1407
No. of input units: 60
No. of output units: 2

pattern 1
0.0 0.0248447204968944 0.0 0.0 0.0 0.0 0.0062111801242236 0.0 0.0
0.0124223602484472 0.0248447204968944 0.0 0.0062111801242236 0.0 0.0 0.118012422360248
0.0 0.0 0.0 0.0 0.0 0.0 0.0124223602484472 0.0062111801242236
0.0062111801242236 0.0124223602484472 0.0062111801242236 0.0062111801242236 0.0559006211180124 0.0
0.0 0.0062111801242236 0.0 0.0 0.0 0.0 0.0 0.0 0.0434782608695652
0.0 0.0062111801242236 0.0 0.0 0.0 0.0 0.0186335403726708 0.0062111801242236
0.031055900621118 0.0 0.0 0.0 0.0 0.031055900621118 0.0062111801242236
0.0496894409937888 0.0 0.0062111801242236 0.0062111801242236 0.0 0.0062111801242236 0.0
0 1
pattern 2
0.00662251655629139 0.033112582781457 0.0 0.0 0.00662251655629139 0.0 0.0
0.00662251655629139 0.0 0.0198675496688742 0.0132450331125828 0.00662251655629139
0.0 0.0 0.0 0.0463576158940397 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0198675496688742 0.0 0.0132450331125828 0.00662251655629139 0.0
0.0529801324503311 0.0198675496688742 0.0 0.00662251655629139 0.0 0.0
0.00662251655629139 0.0 0.0 0.0 0.0529801324503311 0.0
0.00662251655629139 0.0 0.0 0.0 0.00662251655629139 0.0132450331125828
0.033112582781457 0.0 0.0 0.00662251655629139 0.0 0.00662251655629139
0.0 0.0397350993377483 0.00662251655629139 0.00662251655629139 0.0264900662251656
0.0 0.0 0.0
0 1
pattern 3
0.0 0.0137931034482759 0.0 0.0 0.0 0.0 0.0206896551724138 0.0 0.0
0.0137931034482759 0.0137931034482759 0.0275862068965517 0.0 0.0 0.00689655172413793
0.0896551724137931 0.0 0.00689655172413793 0.0 0.0 0.0 0.0
0.0137931034482759 0.00689655172413793 0.0 0.0 0.0206896551724138
0.00689655172413793 0.0551724137931034 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0275862068965517 0.0 0.0206896551724138 0.0 0.0 0.0
0.0137931034482759 0.0 0.0413793103448276 0.0 0.0 0.0 0.0
0.00689655172413793 0.00689655172413793 0.0551724137931034 0.0 0.0
0.00689655172413793 0.0 0.0 0.0
0 1
pattern 4
0.0035778175313059 0.0339892665474061 0.00178890876565295 0.00715563506261181
0.00178890876565295 0.00178890876565295 0.00536672629695885 0.0 0.0
0.0125223613595707 0.0125223613595707 0.0035778175313059 0.0 0.0 0.00178890876565295
0.0500894454382827 0.00178890876565295 0.0 0.0 0.0035778175313059 0.0
0.00178890876565295 0.0035778175313059 0.0 0.00178890876565295
0.00894454382826476 0.0035778175313059 0.0035778175313059 0.0411449016100179 0.0 0.0
0.0035778175313059 0.0 0.0035778175313059 0.00178890876565295 0.0
0.00178890876565295 0.0035778175313059 0.0518783542039356 0.0 0.0035778175313059
0.00178890876565295 0.0 0.00178890876565295 0.0161001788908766
0.00178890876565295 0.0447227191413238 0.0 0.0 0.00178890876565295 0.0
0.00715563506261181 0.00178890876565295 0.0572450805008945 0.00715563506261181
0.0 0.0178890876565295 0.0 0.00178890876565295 0.00715563506261181
0 1
.
.
.
pattern 1402
0.00735294117647059 0.0318627450980392 0.00490196078431373 0.00735294117647059
0.00735294117647059 0.00245098039215686 0.00490196078431373
0.00245098039215686 0.00245098039215686 0.022058823529412 0.00490196078431373
0.00735294117647059 0.00245098039215686 0.00245098039215686
0.00245098039215686 0.0686274509803922 0.00245098039215686 0.00245098039215686
0.0 0.0147058823529412 0.00490196078431373 0.00245098039215686

0.00245098039215686 0.0 0.00490196078431373 0.0122549019607843 0.0 0.0
0.0588235294117647 0.00245098039215686 0.0 0.00490196078431373
0.00245098039215686 0.0 0.00735294117647059 0.00245098039215686
0.00490196078431373 0.0 0.0294117647058824 0.00490196078431373 0.0147058823529412
0.0 0.00490196078431373 0.00490196078431373 0.00490196078431373
0.00245098039215686 0.0465686274509804 0.0 0.0 0.00490196078431373 0.0
0.00245098039215686 0.0 0.0416666666666667 0.00735294117647059 0.0
0.00735294117647059 0.00245098039215686 0.00245098039215686 0.0
1 0
pattern 1403
0.00413564929693962 0.0306038047973532 0.00165425971877585 0.00744416873449132
0.00578990901571547 0.00330851943755517 0.00165425971877585 0.000827129859387924
0.000827129859387924 0.0148883374689826 0.00744416873449132 0.00413564929693962
0.00165425971877585 0.00165425971877585 0.00165425971877585 0.0661703887510339
0.00496277915632754 0.000827129859387924 0.00165425971877585 0.00330851943755517
0.000827129859387924 0.00165425971877585 0.00578990901571547
0.00165425971877585 0.00248138957816377 0.0231596360628619 0.00165425971877585
0.00248138957816377 0.0463192721257237 0.00496277915632754 0.000827129859387924
0.00248138957816377 0.00413564929693962 0.000827129859387924
0.000827129859387924 0.0 0.00496277915632754 0.00413564929693962
0.0446650124069479 0.00413564929693962 0.00413564929693962 0.00165425971877585
0.000827129859387924 0.00248138957816377 0.00413564929693962 0.0
0.0512820512820513 0.0 0.00248138957816377 0.00496277915632754 0.0
0.00248138957816377 0.00413564929693962 0.0339123242349049 0.00496277915632754
0.000827129859387924 0.00330851943755517 0.0 0.00248138957816377
0.00248138957816377
1 0
pattern 1404
0.00276243093922652 0.0193370165745856 0.0 0.0 0.0 0.00276243093922652 0.0
0.00276243093922652 0.00552486187845304 0.0110497237569061 0.00552486187845304
0.00552486187845304 0.0 0.00276243093922652 0.0 0.00828729281767956 0.0856353591160221
0.00276243093922652 0.0 0.00276243093922652 0.00828729281767956 0.0
0.00552486187845304 0.0165745856353591 0.0 0.00552486187845304 0.0248618784530387
0.00552486187845304 0.0 0.0607734806629834 0.00552486187845304 0.0 0.0
0.0 0.00552486187845304 0.00276243093922652 0.0 0.00828729281767956
0.0 0.0331491712707182 0.00276243093922652 0.00828729281767956 0.0 0.0
0.00552486187845304 0.0 0.00552486187845304 0.0552486187845304 0.0
0.00828729281767956 0.00276243093922652 0.0 0.0 0.00276243093922652
0.0138121546961326 0.00276243093922652 0.0 0.00552486187845304
0.00276243093922652 0.0 0.0
1 0
pattern 1405
0.00497512437810945 0.027363184079602 0.00746268656716418 0.00746268656716418
0.00497512437810945 0.00497512437810945 0.00248756218905473
0.00248756218905473 0.00497512437810945 0.0298507462686567 0.00248756218905473
0.00497512437810945 0.00497512437810945 0.00248756218905473
0.00248756218905473 0.0721393034825871 0.00497512437810945 0.00497512437810945
0.0 0.00746268656716418 0.00497512437810945 0.00248756218905473 0.0
0.0 0.0 0.0174129353233831 0.00248756218905473 0.0 0.0621890547263682
0.00248756218905473 0.00248756218905473 0.00497512437810945
0.00248756218905473 0.00248756218905473 0.00746268656716418 0.0
0.00746268656716418 0.0 0.0298507462686567 0.00248756218905473
0.00995024875621891 0.0 0.00497512437810945 0.00248756218905473
0.00995024875621891 0.00248756218905473 0.0422885572139304 0.0 0.0
0.00248756218905473 0.0 0.00248756218905473 0.00248756218905473
0.0422885572139304 0.00497512437810945 0.0 0.00746268656716418
0.00248756218905473 0.0 0.00248756218905473
1 0
pattern 1406
0.00355871886120996 0.0320284697508897 0.0142348754448399 0.0106761565836299 0.00355871886120996
0.0 0.0 0.00711743772241993 0.0 0.0355871886120996 0.0106761565836299
0.00355871886120996 0.0177935943060498 0.0 0.00355871886120996 0.0498220640569395
0.00355871886120996 0.0 0.0 0.00355871886120996 0.0 0.0106761565836299
0.0 0.0 0.00355871886120996 0.0213523131672598 0.0106761565836299
0.00711743772241993 0.0640569395017794 0.00355871886120996 0.00711743772241993
0.0 0.00355871886120996 0.0 0.0 0.0 0.0142348754448399 0.0
0.0142348754448399 0.00355871886120996 0.00355871886120996 0.00355871886120996
0.0 0.00355871886120996 0.00355871886120996 0.00355871886120996
0.0498220640569395 0.00355871886120996 0.0 0.00711743772241993 0.0

0.0106761565836299 0.0 0.0391459074733096 0.0106761565836299 0.0 0.00711743772241993
0.00355871886120996 0.00355871886120996 0.0
1 0
pattern 1407
0.00522193211488251 0.0365535248041775 0.0130548302872063 0.00522193211488251 0.0130548302872063
0.00261096605744125 0.0 0.00522193211488251 0.0 0.0182767624020888
0.00522193211488251 0.00522193211488251 0.00522193211488251 0.0 0.0
0.0809399477806789 0.00522193211488251 0.0 0.0 0.0 0.00261096605744125
0.00783289817232376 0.00783289817232376 0.0 0.00522193211488251
0.0234986945169713 0.0130548302872063 0.00261096605744125 0.0626631853785901
0.00783289817232376 0.0 0.00261096605744125 0.00261096605744125 0.0
0.00261096605744125 0.00783289817232376 0.00522193211488251 0.0
0.0339425587467363 0.0 0.00522193211488251 0.00522193211488251 0.0
0.00261096605744125 0.00261096605744125 0.0 0.0417754569190601 0.0 0.0
0.00522193211488251 0.0 0.0 0.00261096605744125 0.0548302872062663
0.0130548302872063 0.0 0.0 0.0 0.0 0.0
1 0

Appendix G: Part of Testing Set of Pair-wise Binary Classification Approach

SNNS pattern definition file V4.2
generated at Mon Feb 9 12:53:43 2004

No. of patterns: 1356
No. of input units: 60
No. of output units: 2

pattern 1
0.0 0.00680272108843537 0.0 0.0 0.0408163265306122 0.00680272108843537 0.0
0.00680272108843537 0.00680272108843537 0.00680272108843537 0.0
0.00680272108843537 0.00680272108843537 0.0 0.0 0.0 0.0
0.0136054421768707 0.0 0.0 0.0 0.0 0.0 0.00680272108843537 0.0
0.0408163265306122 0.0 0.00680272108843537 0.0136054421768707 0.0 0.0
0.0272108843537415 0.00680272108843537 0.0 0.0 0.0 0.0 0.0 0.0
0.00680272108843537 0.0 0.00680272108843537 0.00680272108843537 0.0
0.00680272108843537 0.0680272108843537 0.0204081632653061 0.0 0.00680272108843537
0.0 0.00680272108843537 0.0 0.0 0.00680272108843537 0.0
0.0136054421768707 0.0476190476190476 0.00680272108843537 0.0 0.0
1 1
pattern 2
0.00714285714285714 0.0 0.0 0.00714285714285714 0.0357142857142857 0.0 0.0
0.0 0.0 0.00714285714285714 0.0 0.00714285714285714
0.00714285714285714 0.0 0.0142857142857143 0.0 0.0 0.00714285714285714
0.00714285714285714 0.00714285714285714 0.0 0.0 0.0 0.0
0.0571428571428571 0.0 0.00714285714285714 0.0142857142857143 0.0 0.0
0.0428571428571429 0.0 0.0 0.0142857142857143 0.0 0.0 0.0 0.0
0.00714285714285714 0.0 0.0 0.0 0.00714285714285714 0.0
0.0285714285714286 0.0142857142857143 0.0 0.0 0.0 0.0 0.00714285714285714
0.0 0.00714285714285714 0.0 0.00714285714285714 0.0214285714285714 0.0
0.0 0.0
1 1
pattern 3
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.166666666666667 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0
1 1
pattern 4
0.0137931034482759 0.0 0.00689655172413793 0.0 0.0689655172413793 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0137931034482759 0.0 0.0137931034482759 0.0 0.0
0.00689655172413793 0.00689655172413793 0.0137931034482759 0.0
0.00689655172413793 0.0 0.0 0.00689655172413793 0.0551724137931034 0.0
0.0 0.00689655172413793 0.0 0.0 0.0482758620689655 0.0
0.00689655172413793 0.0 0.00689655172413793 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0413793103448276 0.00689655172413793 0.0
0.0 0.0 0.00689655172413793 0.0 0.0 0.0137931034482759 0.0
0.0137931034482759 0.0206896551724138 0.0 0.0 0.0
1 1
pattern 5
0.0 0.0 0.0 0.0132450331125828 0.0264900662251656 0.0132450331125828 0.0 0.0
0.00662251655629139 0.00662251655629139 0.00662251655629139 0.0 0.0
0.0 0.0 0.0 0.00662251655629139 0.0132450331125828 0.0
0.00662251655629139 0.0132450331125828 0.0 0.0 0.00662251655629139
0.00662251655629139 0.0463576158940397 0.00662251655629139 0.0
0.00662251655629139 0.0 0.0 0.0529801324503311 0.0 0.0
0.00662251655629139 0.0 0.0 0.00662251655629139 0.0 0.0 0.0
0.0 0.0 0.0 0.00662251655629139 0.0596026490066225 0.00662251655629139
0.00662251655629139 0.0 0.0 0.00662251655629139 0.0
0.00662251655629139 0.0 0.0 0.0 0.0529801324503311 0.0
0.00662251655629139 0.0
1 1
pattern 6

```
0.00684931506849315 0.0 0.0 0.0 0.0410958904109589 0.0 0.0
0.00684931506849315 0.00684931506849315 0.0 0.0 0.00684931506849315
0.0136986301369863 0.0 0.00684931506849315 0.0 0.0 0.0136986301369863
0.00684931506849315 0.00684931506849315 0.0 0.0 0.0
0.00684931506849315 0.0 0.0479452054794521 0.0 0.00684931506849315
0.00684931506849315 0.0 0.0 0.0410958904109589 0.0 0.0 0.0
0.00684931506849315 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.00684931506849315 0.0479452054794521 0.0205479452054795 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0136986301369863 0.0273972602739726 0.0 0.0
0.00684931506849315
1 1
# pattern 7
0.00763358778625954 0.0 0.0 0.0 0.0381679389312977 0.00381679389312977
0.00381679389312977 0.00381679389312977 0.00381679389312977
0.00381679389312977 0.0 0.0 0.0 0.0 0.0 0.00381679389312977
0.00763358778625954 0.00381679389312977 0.00381679389312977 0.0190839694656489
0.00381679389312977 0.0 0.0 0.0114503816793893 0.0 0.0229007633587786
0.00381679389312977 0.0 0.0114503816793893 0.0 0.0 0.0305343511450382 0.0
0.0 0.00381679389312977 0.00381679389312977 0.0 0.00381679389312977
0.0 0.00381679389312977 0.0 0.0 0.00763358778625954
0.00381679389312977 0.0 0.0725190839694656 0.00763358778625954 0.0
0.00381679389312977 0.0 0.00763358778625954 0.0 0.0
0.00381679389312977 0.00763358778625954 0.0152671755725191 0.0343511450381679
0.00381679389312977 0.0 0.0
1 1
.
.
.
# pattern 1351
0.0 0.0 0.0 0.00862068965517241 0.0258620689655172 0.00862068965517241
0.00431034482758621 0.00431034482758621 0.00862068965517241 0.0
0.00431034482758621 0.00862068965517241 0.0 0.0 0.00862068965517241
0.00431034482758621 0.00862068965517241 0.00431034482758621
0.00431034482758621 0.00431034482758621 0.0 0.0172413793103448 0.0
0.00431034482758621 0.0 0.0344827586206897 0.0 0.00431034482758621
0.00862068965517241 0.00862068965517241 0.00862068965517241 0.0172413793103448
0.0 0.00862068965517241 0.00862068965517241 0.0 0.0 0.0 0.0
0.00431034482758621 0.0 0.00431034482758621 0.0 0.0129310344827586 0.0
0.0603448275862069 0.00431034482758621 0.00431034482758621 0.00431034482758621
0.00431034482758621 0.0 0.00431034482758621 0.0 0.00431034482758621
0.0 0.0 0.00431034482758621 0.0 0.0 0.0
1 0
# pattern 1352
0.0 0.00480769230769231 0.0 0.0144230769230769 0.0288461538461538 0.00480769230769231
0.00480769230769231 0.00480769230769231 0.00961538461538462 0.0
0.00480769230769231 0.00961538461538462 0.0 0.0 0.00480769230769231
0.0 0.0 0.0144230769230769 0.00961538461538462 0.0 0.0
0.00961538461538462 0.0 0.00480769230769231 0.00480769230769231
0.0144230769230769 0.0 0.0 0.0240384615384615 0.0 0.0 0.0384615384615385 0.0
0.0 0.0 0.0 0.0 0.00480769230769231 0.00480769230769231
0.00961538461538462 0.0 0.0 0.00961538461538462 0.00480769230769231
0.00480769230769231 0.0288461538461538 0.0144230769230769 0.0 0.0 0.0
0.00480769230769231 0.00480769230769231 0.00480769230769231
0.00480769230769231 0.0 0.00480769230769231 0.0384615384615385 0.0 0.0
0.00480769230769231
1 0
# pattern 1353
0.0 0.0 0.0 0.00436681222707424 0.00436681222707424 0.0174672489082969
0.00436681222707424 0.00873362445414847 0.00873362445414847 0.0
0.0131004366812227 0.00436681222707424 0.00436681222707424 0.0
0.00436681222707424 0.00873362445414847 0.00436681222707424
0.00436681222707424 0.00436681222707424 0.0131004366812227 0.0
0.00873362445414847 0.0 0.00436681222707424 0.0 0.0262008733624454
0.00436681222707424 0.0 0.0 0.00436681222707424 0.00436681222707424
0.0262008733624454 0.0 0.00436681222707424 0.00436681222707424 0.0 0.0
0.0 0.0 0.00873362445414847 0.0 0.0 0.0 0.00873362445414847
0.00873362445414847 0.0698689956331878 0.0131004366812227 0.00436681222707424 0.0
0.0 0.0 0.00436681222707424 0.0 0.00436681222707424 0.0
0.00873362445414847 0.0436681222707424 0.00436681222707424 0.0 0.0
1 0
```

```
# pattern 1354
0.0 0.0 0.0 0.0127118644067797 0.0169491525423729 0.0 0.00423728813559322
0.00423728813559322 0.00423728813559322 0.0 0.00847457627118644
0.00423728813559322 0.0 0.0 0.0127118644067797 0.00423728813559322
0.00847457627118644 0.00423728813559322 0.0169491525423729 0.00847457627118644
0.0 0.00847457627118644 0.0 0.00423728813559322 0.00847457627118644
0.0466101694915254 0.00423728813559322 0.0127118644067797 0.00423728813559322 0.0
0.0 0.0254237288135593 0.00423728813559322 0.0 0.00847457627118644 0.0
0.00423728813559322 0.0 0.0 0.00423728813559322 0.0
0.00423728813559322 0.0 0.00847457627118644 0.00423728813559322
0.0593220338983051 0.0127118644067797 0.00847457627118644 0.0 0.0 0.0
0.00847457627118644 0.0 0.00847457627118644 0.0 0.00423728813559322
0.0508474576271186 0.0 0.0 0.0
1 0
# pattern 1355
0.0 0.0 0.0 0.0140845070422535 0.0234741784037559 0.0 0.00469483568075117
0.00469483568075117 0.00469483568075117 0.0 0.00469483568075117
0.00469483568075117 0.0 0.0 0.00469483568075117 0.0187793427230047
0.00469483568075117 0.00469483568075117 0.0140845070422535 0.00938967136150235
0.0 0.00938967136150235 0.00469483568075117 0.0140845070422535
0.00469483568075117 0.0234741784037559 0.0 0.0140845070422535 0.0187793427230047
0.00469483568075117 0.00469483568075117 0.0234741784037559 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.00469483568075117 0.0 0.00469483568075117
0.0 0.00938967136150235 0.00469483568075117 0.0563380281690141
0.00938967136150235 0.00938967136150235 0.0 0.00469483568075117 0.0
0.0140845070422535 0.0 0.0 0.0 0.0 0.0516431924882629 0.0 0.0
0.00469483568075117
1 0
# pattern 1356
0.00471698113207547 0.0 0.0 0.0235849056603774 0.0283018867924528 0.0 0.0
0.00943396226415094 0.00471698113207547 0.0 0.00943396226415094
0.00943396226415094 0.0 0.0 0.00471698113207547 0.00471698113207547
0.0141509433962264 0.00471698113207547 0.00943396226415094 0.00471698113207547
0.0 0.00943396226415094 0.0 0.00943396226415094 0.0 0.0283018867924528
0.00471698113207547 0.00471698113207547 0.00471698113207547
0.00471698113207547 0.00471698113207547 0.0424528301886792 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.00471698113207547 0.0 0.0 0.0 0.0
0.00471698113207547 0.0660377358490566 0.00471698113207547 0.0141509433962264 0.0
0.0 0.0 0.00471698113207547 0.0 0.0 0.0 0.00471698113207547
0.0849056603773585 0.0 0.0 0.0
1 0
```

Appendix H: Part of Prediction Result of Pair-wise Binary Classification Approach

SNNS result file V1.4-3D

No. of patterns : 1356
No. of input units : 60
No. of output units : 6
startpattern : 1
endpattern : 1356
teaching output included

#1.1
1 0 0 1 0 1
0.88614 0.0222 0.0 0.50626 0.0002 0.84402
#2.1
1 0 0 1 0 1
0.85252 0.10755 0.00001 0.70002 0.03244 0.92945
#3.1
1 0 0 1 0 1
0.41378 0.11702 0.00006 0.80003 0.33323 0.700434
#4.1
1 0 0 1 0 1
0.60059 0.15443 0.14165 0.900334 0.3464546 0.83656
#5.1
1 0 0 1 0 1
0.60059 0.15443 0.14165 0.54355 0.33432 0.93456
#6.1
1 0 0 1 0 1
0.59393 0.11551 0.00004 0.73566 0.354466 0.6534
#7.1
1 0 0 1 0 1
0.63574 0.11859 0.00009 0.743667 0.24345 0.60095
#8.1
1 0 0 1 0 1
0.53134 0.11482 0.00003 0.60455 0.0 0.98434
#9.1
1 0 0 1 0 1
0.99792 0.08982 0.0 0.83555 0.2355 0.99993
#10.1
1 0 0 1 0 1
0.06064 0.12866 0.00085 0.600444 0.34555 0.56005
.
.
.

#1348.1
0 1 0 1 1 0
0.02344 0.99997 0.07572 0 0.59957 0.954545 0.00355
#1349.1
0 1 0 1 1 0
0.000434 0.99938 0.08547 0 0.576766 0.89934 0.03233
#1350.1
0 1 0 1 1 0
0.00034 0.99873 0.08801 0.93434 0.9934 0.02323
#1351.1
0 1 0 1 1 0
0.03443 0.97487 0.00345 0.9956 0.83434 0.9834
#1352.1
0 1 0 1 1 0
0.0344 0.99805 0.08959 0.834355 0.75545 0.03234
#1353.1
0 1 0 1 1 0
0.00344 1 0.06897 0.93434 0.064 0.0
#1354.1
0 1 0 1 1 0

0.003434 0.9343 0.04662 0.7834 0.83434 0.0233
#1355.1
0 1 0 1 1 0
0.0343 0.99995 0.07713 0.56656 0.77565 0.0444
#1356.1
0 1 0 1 1 0
0.00343 0.78834 0.06464 0.73443 0.84343 0.002

Appendix I: Figure of Performance of Pair-wise Binary Classification Approach

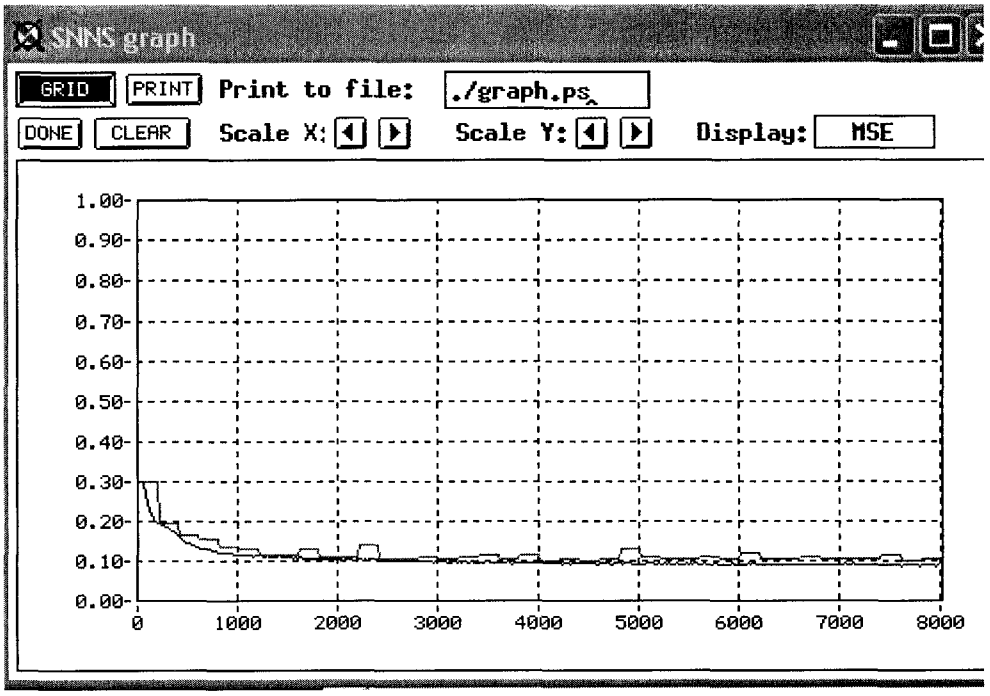


Figure Appendix I: *The performance of Pair-wise Binary Classification Approach*

Appendix J: Part of Training Dataset Using Consensus Method

----------MSSAH---DFKLVLIGDSGVGKSSLLLRFADKAFDPDHESTIGA
AFGVKTLYVDD------KLIKLQIWDTAGQERYRSLAPMYYRGAAGALLVYDITSRDSFT
HAEFWLAELQAYG--NPNIVILLVGNKSDLEDRREVNAEEGEAFAEELGLPFMETSAKTA
TNVEEAFKYIADEIYRKIN--------E----K--PEKMPSLDRSPKGKKSS------GC
C-

----------PQAPAGFLSGKLVIVGDGGVGKTCLLIRFVGGAFPEAYISTVGQNFR
NRTVGII-SKSIKLILWDTAGQER-YHRLRPLYYRNAI-----VILLVFDIDSPASLGNV
SKWVQPLVQHTNPPVVVGLVGNKADMEDKREVLKALYEQLAEEHGWPFFETSAKTGENVS
YIFCSLAERLD----------TRQPVQEPVRIVIEKSEE
TGK--NVFASCCV----------

----------MTEYKLVVVGAGGVGKSALTIQLIQNHFVDEYDPTIEDSYR-
KQVVIDGETCLLDILDTAGQEEYSAMRDQYMRTGEGFLCVFAINNTKSFEDIHNYREQIK
RVKDSEDVPMVLVGNKCDLA-KRTVEFQQAQDLARSYG-IPFIETSAKTRQNVNEIFY-L
VR-I--------AP----------E-S----SQC-----

----------RFIKLVTVGDGAVGKTCMLICYTDDKFPTDYIPTVFDNFSAK-VVVD
GKRVNLGIWDTAGQEDY-RLRPLSYRGADVF-LVFSLVSRASYEN--KKWEPELRVK-P-
-VPVVLVGTKLDLREDR----------VSTEQGEELRRQIG-VAYIECSSKTQ-NVKAV
FDDAIKVV-QPP----K----KKKRKS--------G--L-----

---------MTEY-KLVIVGDGGVGKSALTIQLIQNQFVDEYDPTI-EESYRKQVVIDGE
TCLLDILDTAGQEEYSAMRDQYYRTGEGFLIVFSINSRKSFENIPQYREQIVRVKDSDDV
PIVLVGNKCDLAA----------RKVKPREGQDLAKSY-GLPYLETSAKSRQGVEDA
FYTLVREIRQHKQR--NPPDESGPPCKKCKCVLL----------

----------A--G----------N-IQSIKVVIIGDKAVGK----------SCL
IFCDTPGAFEP-YKSTVFDN-YSEIS--DGRTVNVILWDTAGQEEYDR----------
----------LRISAYPGANVDVLVFTIAN
YLSYENVRHKWHPEVCHHCPDVPIELVGEKADLRAQPEVPRRVAQQID----------
MIKITIIIGNANAKPIIIGKKLECSKLQQDGAR----EAVGAVVNLTPIVKVRSNWRL--

--------MSQ-PYDYLGKLVIVGDGGVGKSSLLNQFTDGKFVDTYIPTIGTDYRVRGVV
D----------GDTIQLQLWDTAGQEEYRDLRSKYMR--------DAHGILIVFDLTNP
PS-----FANVV-DWLSELFRVANCEDVAK----------VLVGNKCDLPV--------
NKVVDTETGRELAKEY---GIPFFETSAKSDQG---------VEEAFKTLADEIPKAMAR
RVN--------EREDGTQRIEGLPIVEK----G----------

----------A-R-KLVIVGDGACGKTSLL-VF-KD-F
PE-YHPTVGENYVED--SALDGSQVDLALWDTAGQEEYMALRDVYYRSTDVVLNVFSVTN
GESFENARGKW--E-KR-KP---NVPIVLVGNKKDLE--RHVS-E-AKN-AE-V------
-SAKTIGAV-YMFCSA-TE-G----------VR-VEE-KTR-AL-ARRS--K-
--S-C-IL

----DYD-L-KLLILGDSGVGKTSLLIQFIDNKFSDEYIPTIGDDFR-K-VV--------
-------V-LQLWDTAGQERFRSLT-AFYRDTDG-LLVYDVTS-ESFENLPNW--E---H
--CRNV----IILVGNK-DL----------QRVVKTEQAQALAESYG-IPYFETSA
KTGQNVDAAFECITKLALK----------S-KKKKGCC--

----------SMTDYDY--RLIKFVALGDSGVGKTSLLNRFTDAKFSIVYQKTI
GIDFPEKGVVYEDST----------RVDLQLWDTAGQERFRSLTTAYLRDASGFL
IVFDITNENSFLNVRKWIEQIIEHAYCDDPDIILVGNKADLENRRQVLAEEAEFLATGLG
IPYFETSAKTGTNVEDLFAKLVDLVPE----------RAEITVEKTSLPVHE
ERP---TSEDTAFEERKDGSCNC----------

---------DYEFKLVLVGDSGVGKSAFLIRLIDNEFVDEYDSTIGVEFR-KTVVIDGKT
VKLQIWDTAGQERYRAMRDAYYRGGVGALCVFDITNRKTFDNIHRWLRELKRVKDS--VP
IVLVGNKCDLP-SRAVSTQ--QDLARKYGLPFIETSAKTRQN----F----------
----------N----------

-------MAAIEYKLVVVGDGGVGKSALLIQFTQDHFVDEYDPTIEDSYR-KQVVIDGKT
IELDIWDTAGQEEYSALRPQYYRDGEGFLLVYSITSPDSFENIHEYWTQIRRHKDSD-VP
IVLVGNKSDLAS----------QRQVSPEQGRALAEQI-GFPFLETSAKTADNVEEAF
YTLARAVRG----------IVK--QPDKKGSGCMSLP----------K-V-S

----------KTRAIKCVVVGDGNVGKTCLLNSYT----------IN
TFPDK-----YIITVDTPGLHH--K-L-K-M-S---KALSD----V----L----K----
---N-FDNYSKPVILDGSKINLALWDTAGQKDYDRLRP---LSYPQADVFLSAF-SVNNP
ASLENVISKWLPESRHHFPK----------APIILVGTKLDLRDEIPYESAVRI----
---------EGRIPVSKTQGKKMVIGIGAVKYLECSALAQQGLKQVFEKAVRLGLQPIAQ
KKWRKKEKSLCTLG--------

----------MTSRYDYLFKVLIIGDSGVGKSSLLNRFVEKKFTNQFKSTIGIDFKT
KTIMVDGKRVKLQIWDTAGQERFRSITTAYYRGAMGALLVYDVTAPNTFNNVDSWLTEFR
IHAS----ENIVIILVGNKCDLEDQRVVTFEEGRALADKNNVPFFETSAKENVNVEEAFQ
TIAREILKQITEVELANERPEPGKQDGNDRADQS-------Q-TNAESCSC--

----------A-ATRYKLVIVGDGGVGKSSLLLQFV---------
----------KGHFPEVYDPTVEDNY-VRDISIDGKQCLLQIWDTAGQEHFNALRR
LSYSKTDFILLFYSVTNTDSFDEIA---EKWLDVKDHACPNVPIILVGNKKDLESEREVS
---GEAQAAKDELVFIETSAKSAINIEEIFYSLCSELTKEGVLDVFDG----------
--AT---------GKQGKKKCCLIL

Appendix K: UOWNNS

1. About UOWNNS

"UOWNNS" is a program to implement Artificial Neural Network (NN) using Java program language. It consists two methods: Back Propagation and Self Organizing Map. Some functions will be added for future work. This is a beta version program.

2. User's Guide

a) System requirement

Since it is a Java program, there is no special requirement for the Operating System. To run this program, you need to install the Java Runtime Environment (JRE). This program is developed under JDK 1.3, so it is recommended to install JRE1.3 or JDK 1.3 before you run this program.

b) Installation

- Java runtime environment

Download Java SDK or JRE and install into your system. It is recommend you add Java directory to you PATH variable. More help is available at Sun website: for Windows , for UNIX , and here is some help for Mac.

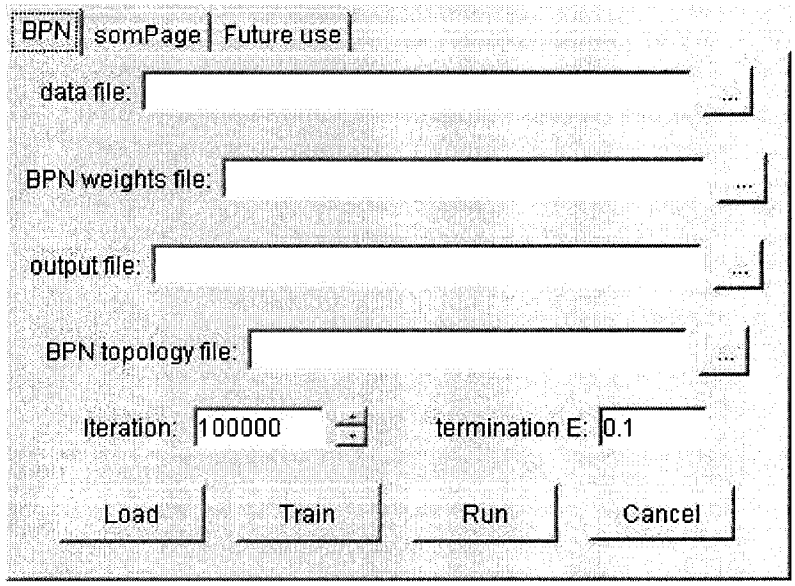
- NN program

Just unzip all the files to a certain folder. In the command line, go to that folder and type in 'java NN' (not 'java nn'). If you did not set the PATH variable, you need to give the full path of java program, for example, you may need to type in something like '\jdk1.3.1_01\bin\java NN'.
There is a class called Test.class in the folder. You can test your Java JRE by typing 'java Test' to run the program.

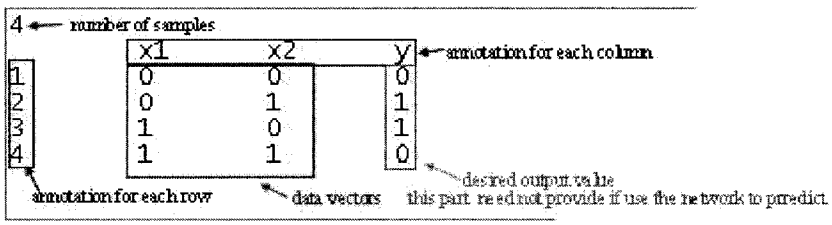
c) Run the prgram

1. BPN

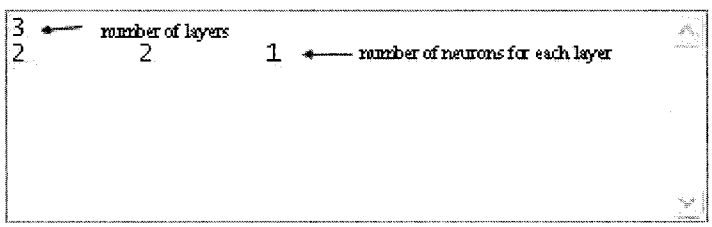
   The Back Propagation page:



- data file: the file contains BPN samples. You can choose the file by click the button besides the text field. Here is the data file format, BPN sample data file:



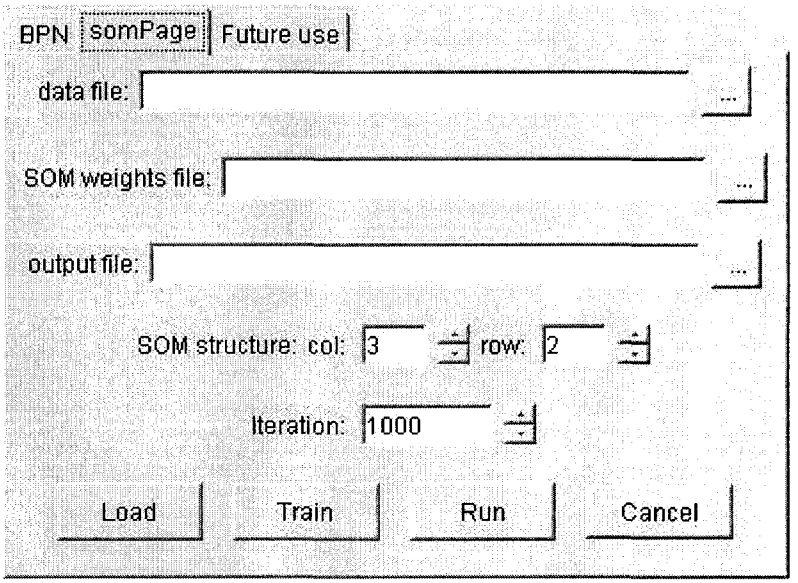
- BPN weights File: the file where the program will store the weights when training or the file where the program will get the weights when prediction.
- output file: the prediction output file. You need not to give this file name if you do not press the Run button.
- BPN topology file: this file tell the program the topology of the network. You need not to provide this file if you can provide weights through Load button. File sample:



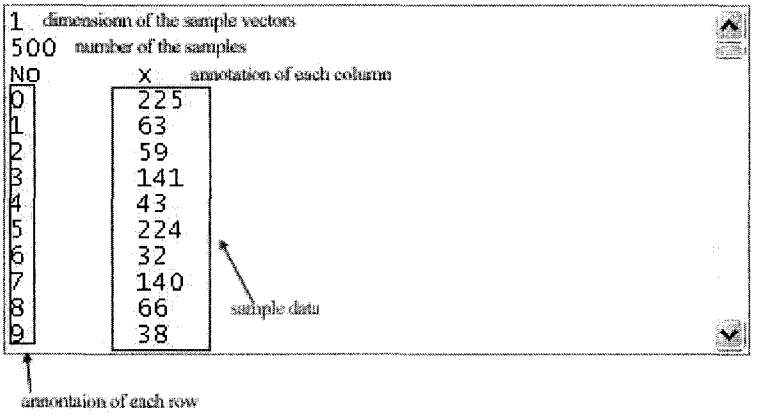
- Iteration: tainning iteration times.
- termination E: threshold to stop the tranning.

- Load: retrieve the network topology and weights from the weights file.
- Train: use the sample data set and the topology to setup the network and process the trainning.
- Run: use the network to predict samples' output.
- Cancel: exit the program.

2. SOM The Self-Organizing Page:



- data file: SOM sample data file. Sample:



- SOM weight file: when trainning, the program will store the gained weights, when pressing Load button, the program will retrieve the weights from this file.
- output file: the program will store the output prediction to this file.
- col:, row: topology of the SOM.
- Iteration: trainning times.

- Load: retrieve the weight from the weight file.
- Train: train the SOM.
- Run: predict with the SOM.
- Cancel: exit the program.

3. Programmer's Guide

The supplied classed contain complete javadoc documentation. Refer to its source code and the API in HTML format.

4. Design & Implementation

- MainFrame.java
  This is the main program and the interface.
- Layer.java
  The Class defines a single layer in Back Propagation Neural Network.
  Using numberOfNeurons and numberOFPrelayer as two parameter to construct an instance of Layer. If this is the input layer, this number shiuld be zero.
  The method, ***calculateOutput***, calculates the output of this layer.
- Cluster.java
  Cluster class clusters samples using SOM.
  1. The method, loadWeight(Point[][] weights), loads the weights of SOM.
- BPN.java
  Class for Back Propagation Neural Network[9].
  1. The method, ***feedForward(double[] input)***, feedforwards the network and creates output from the inputs.
  2. The method, ***backPropagation(double[] desiredOutput)***, adjust the network using back propagation method.
  3. The method, ***train(double[][][] input, double[][][] output, int iteration, double threshold)***, trains the network with sample set.
- NN.java
  This is the class for running neural network.
- Point.java
  Point class is to define sample points for SOM.
  The method,***distance(Point anotherPoint)***, calculates the distance between two points.
- NNException.java
  The class to catch the exception.
- Som.java
  Som class defines the Self Organizing Maps[10].
  1. The method, ***getNearestNeuron(Point sample)***, calculates the nearest neuron in the SOM to the sample point.
  2. The method, ***adjustSOM(Point sample, int[] nearestNeuronPosition, double k, int radius),*** adjusts the SOM according to a certain sample.

5. Testing

This program has been tested under JDK 1.3.1_01.

- BPN Sample file: sampleBPN.txt
  Condition: topology file: topBPN.txt , iteration=1000000, threshold=0.01
  Results: outBPN.txt

Example1:

SampleBPN.txt:

4

	x1	x2	y
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	0

t

topBPN.txt:

3

2 2 1

outBPN.txt:

BPN output file.

	x1	x2	y	Output0
1	0	0	0	0.04804985051309834
2	0	1	1	0.9522602202427373
3	1	0	1	0.9520662792042568
4	1	1	0	0.055521398627920396

Example2:

The Standard Genetic Code.
Triplets encodes the same amino acid. For example, UCC encodes the Alanine, UGU encodes the Cysteine, AAA encodes the Lysine, AAU encodes Asparagine. We use 12 input units, (one possibility is a triplet, 000101000010 means AAA), 2 (or more) intermidiate units, and 4 output units(each unit means a amino acid, x1 means Alanine).

SampleBPN.txt:

4

	y1	y2	y3	y4	y5	y6	y7	y8	y9	y19	y11	y12	x1	x2	x3	x4
1	0	0	0	1	0	1	0	0	0	1	0	0	1	0	0	0
2	0	0	0	1	0	0	1	0	0	0	0	1	0	1	0	0
3	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0
4	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1

topBPN.txt:

3
12 2 4

outBPN.txt:

BPN output file.

y1 y2 y3 y4 y5 y6 y7 y8 y9 y19 y11 y12 x1 x2 x3 x4 Output0 Output1 Output2 Output3

1 0 0 0 1 0 1 0 0 0 1 0 0 1 0 0 0 0.9493297172282192 0.08014596531875971 0.07797009799881426 3.2546860173967603E-6

2 0 0 0 1 0 0 1 0 0 0 0 1 0 1 0 0 0.0649112775377087 0.857729388340988 0.004603350014041741 0.05464063166057247

3 1 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0.0017403244088671675 9.134423992421039E-4 0.8600473516145841 0.041180263614069505

4 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 2.051453383809621E-4 0.10622915044685545 0.1360685421683296 0.9228177838014753

- SOM Sample file: sampleSOM.txt
  Condition: col=3, row=2, iteration=10000
  Result: outSOM.txt

sampleSOM.txt:

1									
500									
No	**x**	**No**	**x**	**No**	**x**	**No**	**x**	**No**	**x**
0	225	100	121	200	61	300	21	400	46
1	63	101	237	201	222	301	108	401	64
2	59	102	106	202	148	302	162	402	28
3	141	103	91	203	49	303	109	403	31
4	43	104	243	204	234	304	119	404	3
5	224	105	128	205	94	305	201	405	168
6	32	106	138	206	198	306	78	406	90
7	140	107	82	207	230	307	48	407	170
8	66	108	130	208	77	308	33	408	251
9	38	109	93	209	128	309	172	409	198
10	171	110	157	210	230	310	144	410	108
11	151	111	60	211	7	311	229	411	238
12	136	112	200	212	219	312	161	412	117
13	151	113	234	213	48	313	133	413	24
14	183	114	0	214	253	314	201	414	82
15	126	115	147	215	0	315	152	415	181
16	149	116	255	216	53	316	177	416	7
17	253	117	226	217	58	317	174	417	33
18	163	118	243	218	190	318	143	418	65
19	59	119	246	219	156	319	135	419	150
20	142	120	36	220	168	320	2	420	14

21	145	121	100	221	125	321	145	421	94
22	231	122	108	222	225	322	109	422	181
23	3	123	126	223	214	323	129	423	132
24	128	124	246	224	36	324	105	424	98
25	102	125	12	225	80	325	40	425	101
26	234	126	151	226	186	326	83	426	186
27	42	127	164	227	179	327	143	427	20
28	185	128	204	228	140	328	219	428	154
29	67	129	191	229	10	329	14	429	127
30	8	130	112	230	254	330	151	430	31
31	144	131	115	231	51	331	189	431	48
32	206	132	33	232	227	332	20	432	217
33	107	133	35	233	59	333	253	433	4
34	200	134	74	234	189	334	174	434	43
35	120	135	73	235	127	335	119	435	142
36	222	136	69	236	127	336	93	436	132
37	132	137	207	237	44	337	243	437	249
38	44	138	154	238	66	338	235	438	229
39	52	139	223	239	166	339	201	439	159
40	182	140	204	240	71	340	178	440	244
41	250	141	89	241	166	341	146	441	85
42	131	142	0	242	182	342	56	442	0
43	236	143	159	243	57	343	132	443	235
44	170	144	77	244	21	344	100	444	23
45	171	145	32	245	208	345	143	445	74
46	58	146	69	246	142	346	42	446	125
47	83	147	165	247	253	347	189	447	48
48	67	148	69	248	107	348	126	448	204
49	159	149	220	249	62	349	35	449	170
50	27	150	41	250	64	350	205	450	63
51	138	151	215	251	183	351	182	451	141
52	133	152	129	252	8	352	78	452	26
53	246	153	84	253	231	353	124	453	73
54	181	154	200	254	23	354	53	454	35
55	5	155	172	255	215	355	131	455	45
56	32	156	207	256	98	356	7	456	129
57	201	157	167	257	199	357	64	457	192
58	235	158	199	258	236	358	78	458	192
59	106	159	20	259	108	359	37	459	201
60	176	160	165	260	176	360	183	460	40
61	22	161	188	261	101	361	200	461	168
62	97	162	218	262	22	362	58	462	116
63	191	163	255	263	60	363	235	463	56
64	84	164	51	264	255	364	209	464	91
65	44	165	78	265	91	365	236	465	138
66	233	166	215	266	59	366	95	466	105
67	66	167	148	267	113	367	13	467	244
68	198	168	90	268	175	368	29	468	97
69	26	169	242	269	59	369	106	469	104
70	52	170	48	270	109	370	120	470	243
71	100	171	19	271	135	371	132	471	70
72	172	172	92	272	221	372	18	472	207
73	11	173	46	273	211	373	252	473	186
74	186	174	191	274	182	374	10	474	13
75	65	175	200	275	110	375	240	475	47
76	19	176	65	276	144	376	184	476	106
77	24	177	129	277	0	377	77	477	18
78	82	178	174	278	77	378	32	478	182
79	219	179	19	279	22	379	69	479	206
80	215	180	154	280	198	380	127	480	249

81	118	181	161	281	162	381	174	481	128
82	211	182	170	282	163	382	153	482	186
83	9	183	224	283	126	383	167	483	59
84	75	184	222	284	47	384	0	484	205
85	83	185	240	285	47	385	126	485	240
86	3	186	198	286	245	386	46	486	158
87	192	187	79	287	129	387	122	487	50
88	132	188	123	288	33	388	61	488	65
89	202	189	65	289	196	389	159	489	89
90	76	190	29	290	56	390	242	490	146
91	250	191	114	291	115	391	74	491	12
92	170	192	174	292	17	392	171	492	50
93	183	193	181	293	40	393	68	493	180
94	213	194	214	294	28	394	76	494	220
95	216	195	45	295	79	395	124	495	167
96	250	196	84	296	24	396	114	496	74
97	250	197	62	297	115	397	144	497	243
98	54	198	32	298	91	398	87	498	122
99	30	199	89	299	21	399	55	499	221

outSom.txt:

Cluster Output File.

Cluster Output File.																			
No	**x**	**row**	**col**	**No**	**x**	**row**	**col**	**No**	**x**	**row**	**col**	**No**	**x**	**row**	**col**	**No**	**x**	**row**	**col**
10	171	0	0	1	63	1	0	78	82	0	1	212	219	1	1	420	14	0	2
14	183	0	0	2	59	1	0	85	83	0	1	214	253	1	1	427	20	0	2
18	163	0	0	4	43	1	0	102	106	0	1	222	225	1	1	430	31	0	2
28	185	0	0	8	66	1	0	103	91	0	1	223	214	1	1	433	4	0	2
32	206	0	0	19	59	1	0	107	82	0	1	230	254	1	1	442	0	0	2
34	200	0	0	27	42	1	0	109	93	0	1	232	227	1	1	444	23	0	2
40	182	0	0	29	67	1	0	121	100	0	1	247	253	1	1	452	26	0	2
44	170	0	0	38	44	1	0	122	108	0	1	253	231	1	1	454	35	0	2
45	171	0	0	39	52	1	0	130	112	0	1	255	215	1	1	474	13	0	2
54	181	0	0	46	58	1	0	131	115	0	1	258	236	1	1	477	18	0	2
57	201	0	0	48	67	1	0	141	89	0	1	264	255	1	1	491	12	0	2
60	176	0	0	65	44	1	0	153	84	0	1	272	221	1	1	3	141	1	2
63	191	0	0	67	66	1	0	168	90	0	1	273	211	1	1	7	140	1	2
68	198	0	0	70	52	1	0	172	92	0	1	286	245	1	1	11	151	1	2
72	172	0	0	75	65	1	0	187	79	0	1	311	229	1	1	12	136	1	2
74	186	0	0	84	75	1	0	191	114	0	1	328	219	1	1	13	151	1	2
87	192	0	0	90	76	1	0	196	84	0	1	333	253	1	1	15	126	1	2
89	202	0	0	98	54	1	0	199	89	0	1	337	243	1	1	16	149	1	2
92	170	0	0	111	60	1	0	205	94	0	1	338	235	1	1	20	142	1	2
93	183	0	0	134	74	1	0	225	80	0	1	363	235	1	1	21	145	1	2
112	200	0	0	135	73	1	0	248	107	0	1	365	236	1	1	24	128	1	2
127	164	0	0	136	69	1	0	256	98	0	1	373	252	1	1	31	144	1	2
128	204	0	0	144	77	1	0	259	108	0	1	375	240	1	1	35	120	1	2
129	191	0	0	146	69	1	0	261	101	0	1	390	242	1	1	37	132	1	2
137	207	0	0	148	69	1	0	265	91	0	1	408	251	1	1	42	131	1	2
140	204	0	0	150	41	1	0	267	113	0	1	411	238	1	1	49	159	1	2
147	165	0	0	164	51	1	0	270	109	0	1	432	217	1	1	51	138	1	2
154	200	0	0	165	78	1	0	275	110	0	1	437	249	1	1	52	133	1	2
155	172	0	0	170	48	1	0	291	115	0	1	438	229	1	1	81	118	1	2
156	207	0	0	173	46	1	0	295	79	0	1	440	244	1	1	88	132	1	2
157	167	0	0	176	65	1	0	297	115	0	1	443	235	1	1	100	121	1	2
158	199	0	0	189	65	1	0	298	91	0	1	467	244	1	1	105	128	1	2
160	165	0	0	195	45	1	0	301	108	0	1	470	243	1	1	106	138	1	2

161	188	0	0	197	62	1	0	303	109	0	1	480	249	1	1	108	130	1	2
174	191	0	0	200	61	1	0	322	109	0	1	485	240	1	1	110	157	1	2
175	200	0	0	203	49	1	0	324	105	0	1	494	220	1	1	115	147	1	2
178	174	0	0	208	77	1	0	326	83	0	1	497	243	1	1	123	126	1	2
182	170	0	0	213	48	1	0	336	93	0	1	499	221	1	1	126	151	1	2
186	198	0	0	216	53	1	0	344	100	0	1	6	32	0	2	138	154	1	2
192	174	0	0	217	58	1	0	366	95	0	1	9	38	0	2	143	159	1	2
193	181	0	0	231	51	1	0	369	106	0	1	23	3	0	2	152	129	1	2
206	198	0	0	233	59	1	0	396	114	0	1	30	8	0	2	167	148	1	2
218	190	0	0	237	44	1	0	398	87	0	1	50	27	0	2	177	129	1	2
220	168	0	0	238	66	1	0	406	90	0	1	55	5	0	2	180	154	1	2
226	186	0	0	240	71	1	0	410	108	0	1	56	32	0	2	181	161	1	2
227	179	0	0	243	57	1	0	412	117	0	1	61	22	0	2	188	123	1	2
234	189	0	0	249	62	1	0	414	82	0	1	69	26	0	2	202	148	1	2
239	166	0	0	250	64	1	0	421	94	0	1	73	11	0	2	209	128	1	2
241	166	0	0	263	60	1	0	424	98	0	1	76	19	0	2	219	156	1	2
242	182	0	0	266	59	1	0	425	101	0	1	77	24	0	2	221	125	1	2
245	208	0	0	269	59	1	0	441	85	0	1	83	9	0	2	228	140	1	2
251	183	0	0	278	77	1	0	462	116	0	1	86	3	0	2	235	127	1	2
257	199	0	0	284	47	1	0	464	91	0	1	99	30	0	2	236	127	1	2
260	176	0	0	285	47	1	0	466	105	0	1	114	0	0	2	246	142	1	2
268	175	0	0	290	56	1	0	468	97	0	1	120	36	0	2	271	135	1	2
274	182	0	0	293	40	1	0	469	104	0	1	125	12	0	2	276	144	1	2
280	198	0	0	306	78	1	0	476	106	0	1	132	33	0	2	283	126	1	2
281	162	0	0	307	48	1	0	489	89	0	1	133	35	0	2	287	129	1	2
282	163	0	0	325	40	1	0	0	225	1	1	142	0	0	2	304	119	1	2
289	196	0	0	342	56	1	0	5	224	1	1	145	32	0	2	310	144	1	2
302	162	0	0	346	42	1	0	17	253	1	1	159	20	0	2	312	161	1	2
305	201	0	0	352	78	1	0	22	231	1	1	171	19	0	2	313	133	1	2
309	172	0	0	354	53	1	0	26	234	1	1	179	19	0	2	315	152	1	2
314	201	0	0	357	64	1	0	36	222	1	1	190	29	0	2	318	143	1	2
316	177	0	0	358	78	1	0	41	250	1	1	198	32	0	2	319	135	1	2
317	174	0	0	362	58	1	0	43	236	1	1	211	7	0	2	321	145	1	2
331	189	0	0	377	77	1	0	53	246	1	1	215	0	0	2	323	129	1	2
334	174	0	0	379	69	1	0	58	235	1	1	224	36	0	2	327	143	1	2
339	201	0	0	386	46	1	0	66	233	1	1	229	10	0	2	330	151	1	2
340	178	0	0	388	61	1	0	79	219	1	1	244	21	0	2	335	119	1	2
347	189	0	0	391	74	1	0	80	215	1	1	252	8	0	2	341	146	1	2
350	205	0	0	393	68	1	0	82	211	1	1	254	23	0	2	343	132	1	2
351	182	0	0	394	76	1	0	91	250	1	1	262	22	0	2	345	143	1	2
360	183	0	0	399	55	1	0	94	213	1	1	277	0	0	2	348	126	1	2
361	200	0	0	400	46	1	0	95	216	1	1	279	22	0	2	353	124	1	2
364	209	0	0	401	64	1	0	96	250	1	1	288	33	0	2	355	131	1	2
376	184	0	0	418	65	1	0	97	250	1	1	292	17	0	2	370	120	1	2
381	174	0	0	431	48	1	0	101	237	1	1	294	28	0	2	371	132	1	2
383	167	0	0	434	43	1	0	104	243	1	1	296	24	0	2	380	127	1	2
392	171	0	0	445	74	1	0	113	234	1	1	299	21	0	2	382	153	1	2
405	168	0	0	447	48	1	0	116	255	1	1	300	21	0	2	385	126	1	2
407	170	0	0	450	63	1	0	117	226	1	1	308	33	0	2	387	122	1	2
409	198	0	0	453	73	1	0	118	243	1	1	320	2	0	2	389	159	1	2
415	181	0	0	455	45	1	0	119	246	1	1	329	14	0	2	395	124	1	2
422	181	0	0	460	40	1	0	124	246	1	1	332	20	0	2	397	144	1	2
426	186	0	0	463	56	1	0	139	223	1	1	349	35	0	2	419	150	1	2
448	204	0	0	471	70	1	0	149	220	1	1	356	7	0	2	423	132	1	2
449	170	0	0	475	47	1	0	151	215	1	1	359	37	0	2	428	154	1	2
457	192	0	0	483	59	1	0	162	218	1	1	367	13	0	2	429	127	1	2
458	192	0	0	487	50	1	0	163	255	1	1	368	29	0	2	435	142	1	2
459	201	0	0	488	65	1	0	166	215	1	1	372	18	0	2	436	132	1	2
461	168	0	0	492	50	1	0	169	242	1	1	374	10	0	2	439	159	1	2
472	207	0	0	496	74	1	0	183	224	1	1	378	32	0	2	446	125	1	2

473	186	0	0	25	102	0	1	184	222	1	1	384	0	0	2	451	141	1	2
478	182	0	0	33	107	0	1	185	240	1	1	402	28	0	2	456	129	1	2
479	206	0	0	47	83	0	1	194	214	1	1	403	31	0	2	465	138	1	2
482	186	0	0	59	106	0	1	201	222	1	1	404	3	0	2	481	128	1	2
484	205	0	0	62	97	0	1	204	234	1	1	413	24	0	2	486	158	1	2
493	180	0	0	64	84	0	1	207	230	1	1	416	7	0	2	490	146	1	2
495	167	0	0	71	100	0	1	210	230	1	1	417	33	0	2	498	122	1	2

Vita Auctoris

The author was born in Shanghai, China in 1976. He completed his B.Sc. Degree in Biology at China Eastern Normal University. He had been working as a research associate at Shanghai Normal University, Shanghai, China, before he immigrated to Canada in 2002.

He is currently a candidate for the M.Sc. degree in computer science, supervised by Dr. Alioune Ngom, at the University of Windsor, Ontario, Canada. His primary research interest is neural network technique in the protein family classification.