UNIVERSITY OF LONDON THESIS

Degree PhD Year 2007 Name of Author CHING WAI TAN

COPYRIGHT

This is a thesis accepted for a Higher Degree of the University of London. It is an unpublished typescript and the copyright is held by the author. All persons consulting the thesis must read and abide by the Copyright Declaration below.

COPYRIGHT DECLARATION

I recognise that the copyright of the above-described thesis rests with the author and that no quotation from it or information derived from it may be published without the prior written consent of the author.

LOAN

Theses may not be lent to individuals, but the University Library may lend a copy to approved libraries within the United Kingdom, for consultation solely on the premises of those libraries. Application should be made to: The Theses Section, University of London Library, Senate House, Malet Street, London WC1E 7HU.

REPRODUCTION

University of London theses may not be reproduced without explicit written permission from the University of London Library. Enquiries should be addressed to the Theses Section of the Library. Regulations concerning reproduction vary according to the date of acceptance of the thesis and are listed below as guidelines.

A. Before 1962. Permission granted only upon the prior written consent of the author. (The University Library will provide addresses where possible).

B. 1962 - 1974. In many cases the author has agreed to permit copying upon completion of a Copyright Declaration.

C. 1975 - 1988. Most theses may be copied upon completion of a Copyright Declaration.

D. 1989 onwards. Most theses may be copied.

This thesis comes within category D.

☐ This copy has been deposited in the Library of UCL

☐ This copy has been deposited in the University of London Library, Senate House, Malet Street, London WC1E 7HU.

Using Machine Learning For Decoy Discrimination In Protein Tertiary Structure Prediction

CHING WAI TAN

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of the

University of London.

Department of Computer Science

University College London

May 30, 2007

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Declaration

I, CHING WAI TAN, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

In this thesis, the novelty of using machine learning to identify the low-RMSD structures in decoy discrimination in protein tertiary structure prediction is investigated. More specifically, neural networks are used to learn to recognize low-RMSD structures, using native protein structures as positive training examples, and simulated decoy structures as negative training examples. Simulated decoy structures are derived by reversing the sequences of native structures in the set of positive training examples, and threading the reversed sequences back to the native structures.

Various input features, extracted from these native and simulated decoy structures, are used as inputs to the neural networks. These input features are the identities of residue pairs, the separation between the residues along the sequence, the pairwise distance and the relative solvent accessibilities of the residues. Various neural networks are created depending on the amount of input features used. The neural networks are tested against the in-house pairwise potentials of mean force method, as well as against a K-Nearest Neighbours algorithm.

The second novel idea of this thesis is to use evolutionary information in the decoy discrimination process. Evolutionary information, in the form of PSI-BLAST profiles, is used as inputs to the neural networks.

Results have shown that the best performing neural network is the one that uses input information comprising of PSI-BLAST profiles of residue pairs, pairwise distance and the relative solvent accessibilities of the residues. This neural network is the best among all methods tested, including the pairwise potentials method, in discriminating the native structures.

Therefore this thesis has demonstrated the feasibility of using machine learning, more specifically neural networks, in the problem of decoy discrimination. More significantly, evolutionary information in the form of PSI-BLAST profiles has been successfully used to further improve decoy discrimination, particularly in the discrimination of native structures.

Acknowledgements

First of all, I would like to thank my primary supervisor, Professor David T. Jones, for his supervision and guidance for the past 3 years. Under him, I have learnt a lot about bioinformatics and more specifically, protein structure prediction in general.

Secondly, I would also like to thank my secondary supervisor, Dr. Massimiliano Pontil, and my transfer viva examiner, Dr. Denise Gorse, for their helpful comments towards my Ph.D.

I would also like to express my gratitude to past and present members of UCL Bioinformatics Unit who have helped me in my Ph.D in one way or another, and they are Anna Lobley, Melissa Pentony, Naama Hurwitz, Trevor Graham, Chris Pettitt, Tim Nugent, Chris Hornsby, Tomas Alarcon, Michael Sadowski, Sonia Shah, David Corney, Dan Roden, Dan Frampton, Stefano Lise, Jacky Pallas, Kevin Bryson, Karen Page, Matthew Trotter, Clare Scurfield, Tim Ebbels, Alistair Coleman, Marialuisa Pellegrini-Calace, Jaz Sodhi, Jonathan Ward, and Liam McGuffin.

Other people who have helped me during the course of my Ph.D include Benjamin Dias, Gabriel Olarte Garcia, Andreas Argyriou, Soren-Aksel Sorensen, Patricia Fenoy, the people of the graduate student administration, namely Professor Angela Sasse, Naomi Jones, Vera Cady and Susan Pike, as well as the members of the UCL CS Technical Services Group, namely Neil Daeche, Kizito Mphande-Ritz, Melita Rowley, John Andrews and Barry Stein.

Last but not least, I would like to thank my wife, Yang Wenhui, for her amazing support in my quest for a doctorate degree during the past 3 years.

Contents

List of Figures

List of Tables

Chapter 1

Literature Review

This chapter gives an overview of the field of protein structure prediction. A short introduction of protein structures is first given, followed by a review of the progress of secondary structure prediction. A survey of tertiary structure prediction then follows. Examples of uses of machine learning in protein structure prediction are also discussed, followed by a short introduction to neural networks.

1.1 Introduction to Protein Structures

A protein is made up of a sequence of amino acids, of which there are 20 different types. Amino acids differ only in their side chains. These side chains also give the corresponding amino acids different properties. Amino acids such as phenylalanine, methionine, alanine, valine, leucine, isoleucine, and proline are hydrophobic; Aspartic acid, glutamic acid, arginine, and lysine are charged; serine, cysteine, tyrosine, threonine, asparagine, glutamine, histidine, and tryptophan are polar [1]. Additional classifications of amino acids are aromatic or aliphatic, large or small. Aromatic amino acids have rings in their side chains; aliphatic amino acids do not have rings in their side chains.

There are 3 main categories of proteins, namely globular, membrane and fibrous. Globular proteins exist in the aqueous environment. As the surrounding environment is water, globular proteins have cores consisting of mainly hydrophobic residues and surfaces consisting of hydrophilic residues. The structures of globular proteins are ex-

perimentally easier to determine, and hence they are the most represented in the PDB database. Membrane proteins exist in lipid environments and have different chemical and structural properties from that of globular proteins. Fibrous proteins are elongated and consist of repetitive amino acid sequences. Because of the differing properties and characteristics of the 3 categories of proteins, it is apparent that the protein structure prediction problem is treated separately for each of these categories. Here, this thesis is concerned with the structure prediction of globular proteins.

The backbone of two adjacent amino acids interacts to form a peptide bond between the carboxyl group of the first amino acid and the amino group of the second amino acid, releasing a molecule of water in the process. The resulting amino acids are known as residues. The amino group of the first amino acid and the carboxyl group of the last amino acid in the polypeptide do not form peptide bonds and are known as the N-terminus and C-terminus respectively.

Each protein has its specific function within the cell. There are several types of protein functions. Proteins are involved in signaling, structure, transport, storage, and gene regulation. Almost all enzymes are proteins. In order to perform specialized and complex functions within an organism, proteins sometimes bind with other macromolecules such as carbohydrates, lipids and nucleic acids to form glycoproteins, lipoproteins, and nucleoproteins respectively. Apart from that, the presence of ligands in the form of metal atoms bound to certain portions of folded polypeptide chains give the protein added reactivity in performing its intended function. For example, iron is found in the oxygen-binding protein, hemoglobin. Some folded polypeptides also have embedded water molecules within their internal structures.

The 3D structure of a protein often gives clues about its function. Although it is by no means a simple one-to-one mapping between 3D structure and function, local structure similarities between proteins can suggest a similar function between them. For example, the zinc-finger motif, which consists of two anti-parallel beta strands and an alpha helix, is commonly found in DNA-binding gene-regulating proteins [2]. The helix-turn-helix (HTH) is also a structural motif commonly used in DNA binding

proteins [3].

Information about protein structure is often described in terms of a hierarchy of four levels, namely primary structure, secondary structure, tertiary structure and quaternary structure. The primary structure is the amino acid sequence of the entire polypeptide chain. The secondary structure is the local fold of a segment of the polypeptide chain, which falls into 3 common categories, namely helix, strand and coil. The alpha helix and the beta strand are the two types of basic secondary structural elements that can be found to occur repetitively in protein structures; the coil is a irregular structural element that occur between the alpha helices and beta strands. The tertiary structure is the 3D structure of the polypeptide chain which is made up of ensembles of secondary structural elements, and the quaternary structure of a multi-chain protein is the composite of the tertiary structures of the various polypeptide chains in the protein. The native fold of a protein refers to the structural state of a protein that enables it to perform its function.

Figure 1.1 shows the cartoon drawings of tertiary structures of an all-helical protein and a beta-sheet protein. Beta strands in a protein can form hydrogen bonds with each other, forming beta sheets, with parallel beta sheets formed with strands pointing in the same direction, and anti-parallel beta sheets formed with strands pointing in the opposite direction. In cartoon drawings of proteins, beta strands are represented as arrows (parallel beta sheets would have arrows in the same direction), and loops are represented as strings, as illustrated in Figure 1.1. Figure 1.2 shows the quaternary structure of a hemoglobin molecule, formed by similar tertiary structures shown in 4 different colours.

Supersecondary structures are commonly occurring motifs of secondary structures that occur adjacent to one another. One example of a supersecondary structure is the

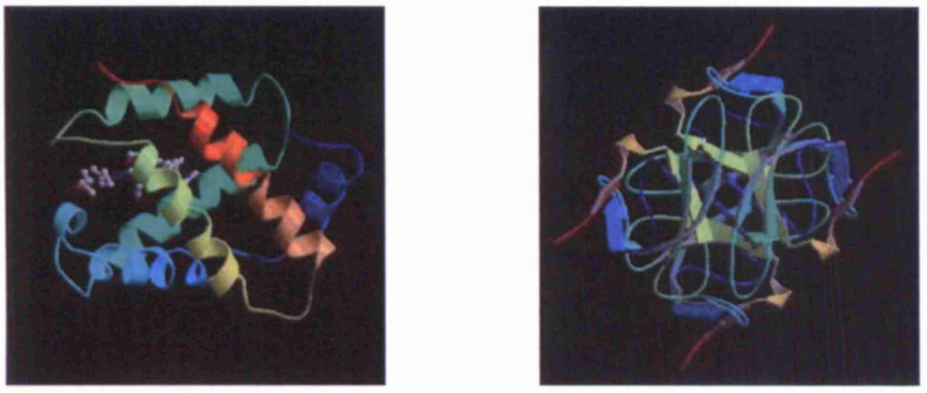


Figure 1.1: Helical protein and beta-sheet protein

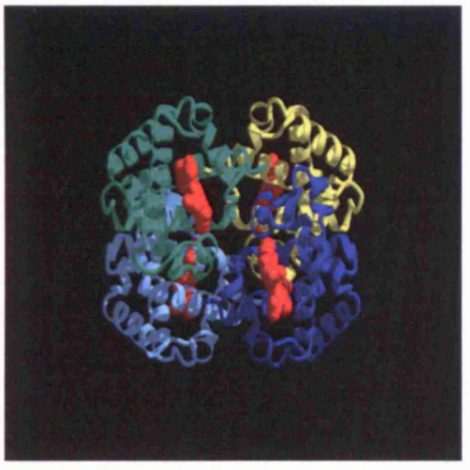


Figure 1.2: Quaternary structure of a hemoglobin molecule

beta hairpin, which consists of two adjacent antiparallel beta strands joined by a loop. Supersecondary structures can be composed of alpha helices, beta strands or a combination of both. In fact, supersecondary structures are so common in proteins that they are used as part of a basis for classifying proteins into protein families in the SCOP database.

A domain is a section of the polypeptide chain that has a stable fold independent of the rest of the chain. The section of polypeptide chain that defines a domain is not necessarily contiguous. Domains can also be units of evolution and function. The presence of domains in proteins adds an extra dimension of consideration in the problem of tertiary structure prediction. There has been recent work in domain boundary prediction [4] as part of protein structure prediction.

In globular proteins, sometimes a particular region within the polypeptide chain adopts many different conformational states instead of just one stable conformation. Such regions only become stable when the protein begins to perform its function. Therefore X-ray crystallography cannot determine the structures of these states when the proteins are crystallized. Such regions are known as disordered regions. The exact 3D conformations of disordered regions in proteins are therefore unknown, and this is an important factor to bear in mind when performing the prediction of structures of globular proteins. Recently, the prediction of disordered regions in proteins has been successful [5].

It is widely assumed, and most certainly rightly so, that the amino acid sequence of a protein alone is sufficient to define the entire tertiary structure of the protein [6]. There have been numerous efforts to predict the secondary and tertiary structures of a given protein. In the application of knowledge-based techniques to the structure prediction problem, other information besides the amino acid sequence has proven useful. Examples of such information include the multiple sequence alignment of a protein with other members of its protein family and sequence profiles, correlated mutations between amino acids, and environmental characteristics such as solvent accessibility.

1.2 Secondary Structure Prediction

Secondary structure prediction has been attempted since the late 1950s [7]. It can serve as a useful intermediate step to predicting the tertiary structure of a sequence because information about the predicted secondary structure states of the residues of a sequence can be used as input to the tertiary structure prediction process. Secondary structure prediction is also a greatly simplified problem because it is essentially the prediction of 3 possible states of each residue in the sequence, as opposed to tertiary structure prediction, which has to predict 3D coordinates.

1.2.1 Definition of secondary structure

The secondary structure of a subsequence forms during the folding process because it is energetically favorable for that particular region of the sequence to adopt such a local conformation. Electronegative and electropositive atoms belonging to C=O and N-H groups respectively in the backbone chain interact with each other to form hydrogen bonds.

The helix and the strand are two types of local conformations that can exist in protein structures, whenever there is regularity in the formation of hydrogen bonds between C=O and N-H groups along the polypeptide chain. A helix forms when the C=O group hydrogen-bonds with an N-H group 3, 4 or 5 positions away along the backbone chain. A strand forms when two sections in the polypeptide chain, which can be far apart along the sequence, form hydrogen bonds between the participating C=O and N-H groups from each section in an extended conformation. Irregular conformations do form between residues, and these are loosely regarded as loop conformations, which also include, in rare cases, residues that do not form any hydrogen bonds.

1.2.2 Secondary structure assignment programs

Intuitively, it is possible to visually inspect and assign, say the helix state to a set of consecutive residues. However an objective assignment of a conformation to each and every residue in the sequence is necessary to avoid ambiguity when it comes to providing 'answers' to secondary structure prediction.

The Define Secondary Structure of Proteins (DSSP) program [8] gives a systematic and unambiguous definition of these secondary structural elements in terms of the presence and location of hydrogen bonds between C=O and N-H groups in the protein sequence. A hydrogen bond is assigned when the net electrostatic force between the C=O and N-H group is below -0.5kcal/mol. DSSP defines 2 elementary conformations, namely the n-turn, n=3,4,5 (T), and the bridge (B), depending on the locations of the 2 interacting C=O and N-H groups within the sequence. Helices are built from 2 or more consecutive n-turns, and these can be the α helix where n=4 (H), 3_{10} helix where n=3 (G) and π helix where n=5 (I). Bridges can be parallel or anti-parallel, depending on the direction of the 2 participating subsequences. Continuous stretches of bridges form β-strands (E) and 1 or more β-strands form β-sheets (also E). Bends (S) are regions with high angles exceeding 70 degrees. Finally, the state '_' refers to a residue of low curvature not in hydrogen-bonded structure.

Other methods for objective assignment of secondary structural state exist, such as STRIDE [9] and DEFINE [10]. The purpose of STRIDE is to model the expert knowledge of the authors of PDB files in terms of secondary structure assignment. It assumes that, barring obvious errors and the usage of DSSP, the authors of PDB files use their expertise to correctly assign secondary structural information to the protein whose structure they have just solved. STRIDE operates on the assumption that hydrogen bonds alone are insufficient criteria for assigning secondary structural states to residues. It defines a formula that incorporates torsion angle and hydrogen bond information, and the parameters of this formula are empirically fitted to match those in the PDB database. According to the authors of STRIDE, one drawback of DSSP is that it tends to split a long helix into 2 given missing hydrogen bonds in the middle in spite of its completely acceptable geometry. STRIDE can overcome this because it

takes torsion angles into account.

DEFINE attempts to identify structural motifs by examining distance matrices obtained using the Cα backbone coordinates and comparing these Cα distances with distances in idealized secondary structure segments. DEFINE also describes super-secondary structural elements. Overall, STRIDE and DSSP are more popular when it comes to secondary structure assignment, with DSSP the more widely used of the two.

In recent years, new methods such as XTLSSTR [11] and KAKSI [12] have been developed for secondary structure assignment. XTLSSTR uses additional information in the form of angles derived from amide-amide interactions to classify secondary structural state, while KAKSI is similar to STRIDE in that it uses the information found in correctly annotated PDB files to derive a set of characteristic values of Cα distances and ϕ, ψ dihedral angles to assign secondary structural state. A niche detection method for specifically detecting π helices exist in the form of SECSTR [13].

Despite these new methods, DSSP is still treated as the de facto standard today, with many X-ray crystallographers using the DSSP program to assign secondary structure to the 3D coordinates of a recently solved protein structure.

While the DSSP definitions of 8 types of secondary structural state are unambiguous and used for exact assignment of state to training datasets of sequences, most prediction techniques are content to deal with only 3 distinct types, namely helices, strands and loops, apart from Baldi's SSPro8 [14] which strives to predict all 8 possible DSSP states. This leads to the issue of the reduction of 8 DSSP states to 3 states, prior to the actual usage of the prediction method and the training, if required, of the prediction method. Of course, if DSSP is not used and the secondary structural states found in the PDB files are taken to represent the correct assignments, there is no need for any reduction method whatsoever. However if DSSP is used as the standard means for assignment, then the reduction issue is relevant. One 8-to-3 state reduction method is to assign G and H to the helix state, B and E to the sheet state, and all others to the loop state. The PSIPRED prediction method [15] uses such a reduction method.

Another reduction method is to assign H to the helix state, E to the sheet state, and all others to the loop state.

Different reduction methods can yield different levels of accuracy (Section 1.2.3 gives a detailed discussion on the accuracy measures used). For example, short helices are generally harder to be correctly predicted by most secondary structure prediction methods [16], and 3_{10} helical residues (DSSP state G) are frequently found in short helices. Therefore, an assignment of G to the loop state makes the prediction method more quantitatively accurate, but has little practical use in providing constraints for the eventual modelling of the 3D structure of the protein sequence. A study performed by Barton and co-workers [17] has demonstrated that the effect of different 8-to-3 reduction methods on the Q_3 accuracy (defined in Equation 1.1 in Section 1.2.3) of some secondary structural prediction methods is about 3%. Therefore, it is important to note that when comparing different secondary structure prediction techniques, it is essential to ensure, whenever DSSP is used, that the same 8-to-3 reduction method has been applied for all methods.

1.2.3 Evaluation criteria for secondary structure prediction

With the issues of the definition of secondary structural states laid out above, the evaluation of the accuracies of secondary structure prediction is now discussed. The most common accuracy measure is the Q_3 score, as shown in Equation 1.1, where the set S consists of 3 elements, namely the helix (H), strand (E), and loop (L).

$$Q_3 = \frac{1}{|S|}\sum_{i \in S} Q_i, \qquad S = \{H, E, L\} \tag{1.1}$$

$$Q_i = \frac{TP_i}{TP_i + FN_i}, \qquad i \in S \tag{1.2}$$

The Q_3 score is the average of each Q_i (i=helix, strand, loop), where Q_i, as shown in Equation 1.2, is defined as the fraction of the number of residues in state i correctly predicted. The number of helices, strands and loops in the database (testing datasets) are frequently not evenly distributed with loops usually comprising of a greater per-

centage than the other two. This can result in a high Q_L score, when a large number of loop residues are correctly predicted, which increases the overall Q_3 accuracy. Therefore, such a Q_3 accuracy can falsely increase the user's confidence in the particular prediction method, because in general, users would be more interested in the correct predictions of helices and strands. This problem can be circumvented by reporting the individual Q_i scores along with the Q_3 score for any secondary structure prediction method.

Leading secondary structure prediction methods have Q_3 scores of about 75% to 80%, depending on the compositions and size of the test datasets. However the Q_3 score does not tell the whole story regarding the accuracy of secondary structure prediction methods. Because the Q_3 score focuses on per-residue accuracy, it neglects to evaluate the overall picture of how predicted secondary structure elements are correctly positioned across the sequence. For example, a spurious prediction of helix-dominated myglobin to be 100% helical would yield a very high Q_3 accuracy, but is not very useful in aiding the understanding of the overall topology of myglobin.

In 1994, Rost and co-workers [18, 19] came up with another accuracy measure for secondary structure prediction, which is known as the Segment Overlap Measure (SOV). SOV_i for each state i (i=helix, strand, loop) measures the extent the predicted segment of state i is identical to the experimentally observed measure. The SOV_3 score is the average of all SOV_i. It is recommended by Rost that both the Q_3 and SOV_3 scores are used for secondary structure evaluation.

Now that the definitions of secondary structure and how predictions can be accurately measured are described, the next section describes the methods available in secondary structure prediction.

1.2.4 Secondary Structure Prediction Methods

Since the 1950s, there have been attempts to predict the secondary structure from sequence alone. Burkhard Rost gave an excellent review of secondary structure prediction methods in [16]. In the paper, he described the various factors that contribute to the increase in performance of secondary structure prediction methods over the years. These factors consist of the usage of evolutionary information from multiple sequence alignments, powerful sequence alignments tools that make these alignments possible, and the increase in size of sequence databases that contribute to the power of the multiple sequence alignment methods.

In the late 1950s, the first secondary structure prediction method [7] attempted to correlate the content of certain amino acids with the contents of α-helices. This soon paved the way for other methods to make use of single sequence information for building classifers to assign secondary structure states for each residue in a sequence. In 1974, Chou and Fasman [20] used a qualitative method, in the form of rules, to try and predict secondary structure. Another popular method then, GOR [21,22], is based on information theory and Bayesian statistics. These secondary structure prediction methods use amino acid propensities along the sequence to predict the secondary structure state of a central residue. Such usage of local information restricts the Q_3 accuracy to around 60%.

The breakthrough of accuracies to 70% and above is achieved through the usage of multiple sequence information. With multiple sequence alignments, it is possible to obtain information regarding the mutability of residues at all positions in a sequence. Such position-specific profiles, as they are called, give vital information about the evolutionary relationships, such as conserved regions, in the protein family to which the sequence belongs.

Zvelebil and co-workers are among the first to have incorporated multiple sequence alignment information into their secondary structure prediction method [23]. However, the landmark PHD secondary structure prediction method [24] is the first that achieved a Q_3 accuracy of above 70%. The PHD method uses evolutionary information as input

to a two-stage neural network. PSIPRED [15] uses PSI-BLAST profiles, intermediate outputs of PSI-BLAST [25], as inputs to a two-stage neural network. In PSIPRED, the importance of filtering low complexity and transmembrane proteins from the sequence database when generating sequence profiles to ensure that the PSI-BLAST profiles obtained are as noise-free as possible was demonstrated. In CASP3 in 1998, PSIPRED was ranked top in the secondary structure prediction category, achieving average Q_3 and SOV_3 scores of 73.4% and 71.9% respectively [26].

Another competitive secondary structure prediction method that uses evolutionary information is SAM-T99 [27], a Hidden Markov Model (HMM) method that constructs protein family profiles. Another HMM method is Christopher Bystroff and David Baker's HMMSTR [28], which is in principle a method for predicting 3D structure, but interestingly has the side effect of generating competitive secondary structure predictions as well. Apart from PHD and PSIPRED, another neural network method exists in the form of SSPro [14]. SSPro uses a recurrent neural network architecture, while PHD and PSIPRED use two-stage feedforward architectures. The second stage of the feedforward architectures of PHD and PSIPRED is to allow the neural network to learn the correlation of the secondary structure propensities of consecutive residues in the sequence, and SSPro achieves such a correlation with the recurrent network architecture instead. In CASP4, SSPro was among the top 10 in terms of SOV when compared with other automated secondary structure prediction servers but the simpler architecture of PSIPRED proved better in performance [29].

HMMs and neural networks are typical useful machine learning methods that can be found in the solutions of several bioinformatics prediction problems, such as gene finding and in this case, secondary structure prediction. Other machine learning methods that have been used in secondary structure prediction are discriminant analysis [30], nearest neighbours [31], linear discriminant functions [32], and support vector machines [33].

The usage of evolutionary information in the form of multiple sequence alignments has undoubtedly increased the accuracy of secondary structure prediction. However,

it is worthwhile noting that the increased sensitivity of sequence search tools such as PSI-BLAST and the increase in size of the sequence databases play their part in ensuring multiple sequence alignments can provide relevant evolutionary information for the development of secondary structure prediction methods [34].

With the large number of purportedly highly accurate secondary structure prediction methods, it can be difficult to select the best method. It is worth pointing out that the reported accuracies of published prediction methods are dependent on the test datasets used. Rost [16] made the important comment that the test dataset for novel secondary structure prediction methods should be as large as possible, in order to be more reflective of the capability of the method. A secondary structure prediction method should also undergo proper cross validation, with care taken to ensure that the training and test datasets share no homologous sequences.

The CASP experiments (up to and including CASP5) perform the role of effective comparison between various secondary structure prediction methods. While CASP results are indicative of the performances of various prediction methods, they come only once every 2 years and it would be desirable for an automated service that exists to compare secondary structure predictions on a regular basis. Fortunately, such a service exists in the form of EVA [35], which is an automated secondary structure assessment server that attempts to evaluate the performances of several secondary structure prediction servers. (EVA actually does more; it evaluates comparative modelling and contact prediction techniques as well). This implies EVA can only evaluate prediction techniques that are automated in the form of servers. EVA sends the sequences of recently solved protein structures to several secondary structure prediction servers, collects the results, and then compares and presents these prediction results online. It also uses the following 8-to-3 reduction technique: HGI to the helix state, EB to the sheet state, and the others to the loop state. Some of the participating prediction servers are PSIPRED [36], PHD [24], JPred [17], and SSPro [14]. Apart from JPred, which uses a consensus based approach, these methods make use of machine learning that learns from input features such as evolutionary information. There is often little difference between the better methods in terms of Q_3 and SOV_3 accuracies, as evaluated by EVA.

The field of secondary structure prediction has reached a level of maturity where consistent Q_3 and SOV_3 accuracies of beyond 80% are arguably difficult to achieve. One possible reason is that the formation of secondary structure of a sequence segment is in part due to long-range interactions within the protein sequence and these are extremely difficult to take into account. It is therefore improbable that 100% in Q_3 and SOV_3 accuracies can be attained. In fact, the CASP community has reached a decision during the CASP5 meeting to drop the evaluation of secondary structure prediction techniques for future CASPs, starting from CASP6.

The present challenge regarding secondary structure is less of improving the accuracy of prediction techniques; rather it is more of how secondary structure prediction techniques can aid in constructing the 3D fold of a target protein sequence. For instance, given a particular secondary structure prediction result, the question of how secondary structural elements can pack together in a compact manner is not still wholly solved in protein structure prediction. Here, it is worth mentioning that specialist methods such as the prediction of β-turns, developed by Adrian Shepherd and co-workers [37], and Raghava and co-workers [38], could be an useful intermediate step when used together with secondary structure prediction, for the prediction of 3D structure. However the use of such specialist prediction methods is currently not very widespread in tertiary structure prediction.

1.3 Tertiary Structure Prediction

The ultimate goal of protein structure prediction is to predict the 3D fold from sequence information alone. With that goal in mind, secondary structure prediction methods can provide useful clues on the local topology of the protein, and such local topology information can help guide the actual prediction of the tertiary structure.

Accurate high resolution computational predictions of structures help provide low cost alternatives to experimental means of obtaining the 3D crystal structure of pro-

teins. Such predictions can also guide experimental efforts in deciding which proteins to crystallize in a bid to cover all possible folds of the protein structure universe. The unravelling of the 3D folds of protein domains is by itself also a means to an end, which is to understand the biological functions and roles of proteins, and how they interact with one another to the benefit or detriment of organisms.

Here, it is important to state that the ensuing review of methods involving tertiary structure prediction pertains only to globular proteins. The prediction of the structures of membrane proteins have additional challenges such as topology prediction, and have much smaller amounts of data to work with. The context of this thesis also pertains only to globular proteins.

1.3.1 Introduction

In the following sections, the term 'target' sequence is used to refer to the sequence whose structure is to be predicted.

The most obvious approach when trying to model the 3D structure of a target sequence is to look for close homologues of the target sequence and then use the 3D structures of these homologues as templates for modelling the target structure. This approach is known as comparative modelling, and works well for target sequences whose close homologues can easily be found from the structure databases [39–44]. The process of using templates to model the 3D structure of the target sequence is however non-trivial [45] and the issues faced in the comparative modelling approach will be discussed in later sections.

The success of comparative modelling relies on the ability of sequence similarity search tools e.g. PSI-BLAST [25] to identify close homologues. Sometimes, for a target sequence, there are similar folds that may exist in the structure databases that cannot be identified by sequence similarity search tools. This is because the sequences of these folds have low percentage sequence identity to the target sequence, which

is beyond the sensitivity of state-of-the-art sequence search tools. Such similar folds may also have emerged due to convergent evolution and have no common evolutionary origin. In such cases, the extent to which sequences are compatible to folds can be evaluated using the fold recognition approach [46–49]. Fold recognition, or threading, encompasses the evaluation of the degree of fit of a sequence to a library of existing folds using energy functions, and the subsequent selection of the fold that yields the lowest energy.

When the structure of the target sequence is indeed a new fold that has never been documented in existing structure databases, template-free approaches are necessary to construct a close approximation to that of the native structure. The earliest methods in template-free approaches used physics-based energy functions (see below for further discussion) for computational protein folding. These are known as the *ab initio* approaches, where the term *ab initio* implies the use of first principles of the laws of physics.

In CASP, the template-free approaches belong to the New Fold category. This category used to be referred to as the *ab initio* approach but subsequently renamed because later template-free approaches encompass the use of statistical knowledge derived from existing structure databases and hence the New Fold category does not consist exclusively of true *ab initio* methods anymore.

New Fold methods consist of lattice based methods [50–52] and fragment assembly methods [53–55]. Both types of methods require a guiding energy function to score the conformation of folds produced by simulations. There are in turn two broad categories of energy functions, namely the physics-based energy functions and the statistical energy functions [56]. Physics-based energy functions use energy functions based on the laws of physics and chemistry. Examples of physics-based energy functions include OPLS force fields [57] and AMBER force fields [58], and solvent models such as the Generalized Born Solvent Model [57]. Statistical energy functions make use of the existing structure database to derive useful discriminatory energy functions. Examples of statistical energy functions include pairwise potentials of mean force [53,59], Bayesian

scoring functions [54] and atomic environmental potentials [60].

Lattice based methods are the earlier methods that use 3D lattices to represent the conformational space of a target protein. Monte Carlo simulations are run, with a scoring energy function. Levitt [52] and Skolnick [50,61] used statistical energy functions to guide the conformational search within the lattice model, while Scheraga used a physics-based energy function to score the conformations [51].

Fragment assembly methods involve the assembly of 3D fragments of short peptide sequences chosen from a library of fragments, guided by an energy function [53–55]. The conformation space of the 3D fold of the target space is huge and because the guiding energy function used for fragment assembly is not perfect, it makes sense to generate large numbers of candidate 3D folds for the approximation of the native structure. These candidate folds are frequently referred to as decoys.

Frequently, the constructed fold with the lowest energy is chosen as the best approximation to the native structure. In this thesis, a novel decoy discrimination method, using machine learning and more specifically neural networks, is developed as a step towards solving the challenge of selecting the best fold.

The 3 different approaches of tertiary structure prediction, namely comparative modelling, fold recognition and New Fold, frequently do not exist in isolation. The process of predicting the structure of a target sequence often involves more than one approach. For example, the construction of loops in comparative modelling targets require the application of New Fold methods. Some New Fold methods [62] involve the perturbation of a starting 3D fold obtained from fold recognition.

In subsequent sections, issues detailing the challenges of each approach are discussed. Major advances in the field of tertiary structure prediction have been achieved with CASP experiments, such as the assessment of prediction quality of candidate models. The next section therefore outlines the role of CASP in the advancement of the field of tertiary structure prediction.

1.3.2 The role of CASP

It is impossible to give a review of tertiary structure prediction without mentioning CASP [63–68]. The Critical Assessment of Techniques for Protein Structure Prediction (CASP) experiment takes place every 2 years, the most recent experiment being CASP7. The first CASP meeting was organized by John Moult [63] in 1994, and subsequent CASP meetings took place every two years. A Protein Structure Prediction Center [69] exists for the purpose of conducting the CASP experiments.

In CASP, structures of newly solved proteins are solicited from structural biological groups, who temporarily withheld their structures from the public so that blind predictions of these proteins could be performed by the structural prediction community. The sequences of these structures are then sent to registered prediction groups over a period of time. Prediction groups can either be automated servers or human prediction groups. In the case of the former, the prediction results are to be sent back to the CASP organizers within 48 hours. After the prediction season is over, the assessors of CASP then analyze the results and a meeting convenes for the participants to discuss these results a few months later.

CASP has played a vital role in advancing the field of protein structure prediction in a number of ways.

- Firstly, CASP provides the opportunity for the blind prediction of protein structures. This ensures that there is no possibility of prediction groups inadvertently using the information of the target structures to derive their predictions, and hence provides a stringent test for all structure prediction methods in the field.

- CASP creates a level playing field for all structure prediction methods by providing a common set of test proteins and this ensures that the performance of each method is critically reviewed in an unbiased manner, where the identities of the prediction groups that submitted the models are withheld from the assessors.

- CASP also helps to advance the field of structure comparison and model assessment because there is a need to effectively rank different prediction models. The Root-Mean-Square deviation (RMSD), defined in Equation 1.3 on page 44 and to be discussed in Section 1.3.3.1, has been found to be inadequate in awarding credit to models which have highly similar predicted local substructures to the corresponding substructures in the native structure (low local RMSD), but have significantly different orientations between the substructures themselves which can result in a high global RMSD. The GDT-TS measure [70] and the Hubbard plot [71] are innovative measures that are borne out of the need to provide critical, accurate and comprehensive tools for the purpose of assessment of prediction models in the CASP experiments.

- CASP also provides a platform for automated servers to compete against the best human prediction groups. While the best automated servers still lag behind the best human prediction groups, the advancement in the performance of automated prediction servers is important, given that the amount of genome sequences continue to grow in the sequence databanks and that the only reasonable way to predict the structures of newly sequenced genomes in a fast and efficient manner is through the use of automated servers. In addition, automated servers can be used by people, such as biologists, who are not necessarily experts in protein structure prediction.

- Over the past few CASPs, there have been new sub problems introduced, such as prediction of disordered regions, domain boundary prediction and contact map prediction. CASP allows for such research problems to be analyzed and tested in a manner that has been reaping benefits in mainstream 3D structure prediction.

- CASP has also inspired a server-only equivalent experiment in the form of CAFASP [72]. The development of meta-servers such as 3D-Jury [73], protein structure prediction servers which perform a consensus prediction by using prediction results from other fully automated servers, provide an extra dimension to the field in terms of providing better performance, although the aspect of credit assignment to high performing meta-servers, especially in a CASP-like scenario, is somewhat contentious and debatable.

- Finally, there is a basis of the comparison of the progress of protein structure prediction in CASP over the past decade [74], which helps to highlight the challenges in the field in a clear manner, which can only be beneficial to the research groups that are working on the protein structure problem.

All in all, the field of protein structure prediction has benefitted immensely from the CASP experiments. The next section describes the various methods of structure assessment in protein tertiary structure prediction.

1.3.3 Structure Comparison

Structure comparison between two different protein structures can be performed in two different contexts. The first type is known as the structural superposition of two disparate structures, where the alignments of the two protein sequences of these structures are known. The second type consists of the structural alignment of two structures in the absence of sequence alignment information. The latter type of structure comparison is obviously harder than the first.

1.3.3.1 Structure Superposition

Structural superposition is performed in a sequence dependent manner, where two structures are aligned with several matching residue pairs, one residue from each structure, acting as anchor points. Frequently, the anchor points extend to the entire sequence which is shared by both structures, when one structure aims to be the prediction of the other in the context of protein tertiary structure prediction. Structure superposition is an applied mathematical problem of aligning both structures, given the anchor points of various residue pairs, so that the lowest quantity of a measure, such as the Root-Mean-Square deviation (RMSD) shown in Equation 1.3, is yielded.

$$RMSD(\boldsymbol{x},\boldsymbol{y}) = \sqrt{\frac{\sum_{i=1}^{n} ||x_i - y_i||^2}{n}} \tag{1.3}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are the $n \times 3$ matrices that describe the 3D coordinates of the two struc-

tures, each of n residues long, that are to be superposed with each other.

The purpose of structural superposition is to assess how close one structure is to another. In cases where the sequences of both structures are not 100% identical but are still highly similar, structural superposition is useful in comparing these structural homologues, for purposes of gaining insightful knowledge that can be inferred from the degree of similarity of these two structures. One example may be the identification of common residues that have clefts of similar shapes that may give clues to the biological functions of one of the structures.

In such cases, the measure of closeness often used is the RMSD in Equation 1.3, and optimal algorithms exist that can align both structures such that they yield the lowest possible RMSD. The lowest RMSD obtained after the optimal superposition of two structures can then reflect the extent of structural similarity between the structures.

In the context of protein tertiary structure prediction where one structure is the target model and the other a predicted model, the RMSD serves as a good gauge for comparing the quality of the predicted model. A model with low RMSD, say ≤ 1.5Å, can be regarded as an excellent prediction, while a model with high RMSD, say 6Å, is obviously of lesser quality [75].

In the running of the CASP experiments, however, the assessors had realized that the simple RMSD measure is not enough to give enough credit to some prediction models. For example, an erroneous orientation in the connecting loop of an otherwise excellent prediction model will give it a high RMSD, without doing justice to the other parts of the model which are correctly predicted. The inevitable competitive nature of the CASP experiments, even though it is meant to be a cooperative experiment, also places a demand on clear-cut ranking performance measures that can be assigned to the prediction models submitted by participating prediction groups.

To deal with the problems above, the GDT-TS measure [70] was devised, and it has served well in subsequent CASPs. The GDT-TS measure is described as follows:

- The aim is to find a superposition between two structures that has the largest set of not necessarily contiguous residues with an RMSD below a certain threshold.
- Several thresholds can be tried, e.g. 2Å, 4Å, 6Å, 8Å etc.
- For each threshold, an iterative procedure is run to obtain superpositions from a starting set of subset of residues until the subset remains unchanged from the previous iteration.
- The number of residues in the subsets obtained for different thresholds is averaged in CASP to provide a mean score for quantifying the quality of each predicted structure.

In CASP, different thresholds are used to capture different degrees of the qualities of different models. A large threshold is suitable for differentiating models in the Template Free category, while a small threshold of 1Å is useful for discerning Comparative Modelling prediction models. However GDT-TS is still imperfect, and most CASP prediction models are examined by eye to determine their quality. GDT-TS is also less useful in discerning prediction models generated using template free modelling because these models tend to have higher RMSDs and therefore the difference in quality between substructures of these models may not be reflected in the GDT-TS scores.

The RMS-coverage graph [71], or Hubbard plot, is also a useful tool for gauging the prediction quality of various models. The Hubbard plot shows the lowest RMSD for a particular subset of not necessarily contiguous residues of each prediction model, where the subset of residues range from 1 to the number of residues in the sequence, and the better the quality of the prediction model, the larger the subset of residues for a given RMSD threshold. This allows the CASP assessors to immediately identify the better quality prediction models of a given target, as well as to identify interesting predictions that perform extremely well for a subset of residues, but not as well for the entire set of residues.

Other researchers have also devised structure assessment measures such as the TM-score [76] and the MaxSub score [77] to determine the quality of prediction models

in protein tertiary structure prediction. The MaxSub program is an automated method which aims to obtain the superposition with the largest subset of not necessarily contiguous Cα atoms under a specified distance cutoff (e.g. 3.5Å) and produces a similar normalized score that sum up the quality of a predicted model. Yang Zhang and coworkers [76] proposed the TM-score which produces a normalized score that describes the quality of a predicted model and which is not dependent on the length of the protein.

1.3.3.2 Structure Alignment

Structure alignment, in the absence of sequence alignment information, is performed in a sequence-independent manner. Structure alignment allows for the classification of unknown folds into fold classes, and also allows for the comparison of the folds of unknown proteins to proteins of known function in the context of functional prediction. In the context of CASP, structural alignment also allows for the search of the best template from the set of known PDB structures during the assessment of Comparative Modelling and Fold Recognition targets.

Common structure alignment tools which assume no sequence dependence are CE [78], SSAP [79], VAST [80] and DALI [81]. Some of these tools are used for the assessment of Fold Recognition prediction models in CASP.

In the next few sections, the issues regarding the common approaches used in tertiary structure prediction are presented and discussed.

1.3.4 Structure Prediction Issues

In more recent CASP experiments including the most recent CASP7, target sequences are not preassigned to the Comparative Modelling, Fold Recognition and Template Free categories when they are made available to prediction groups. Instead the prediction groups would have to adopt whatever they deem to be the best methodology in predicting the structure of the target sequence, be it fragment assembly or template modelling.

Therefore, many of the leading prediction groups in CASP6 adopt comprehensive processes that allow them to first identify possible templates using either fold recognition methods or searches on the structure databases, and in the event of failing to find a template for the target sequence, to adopt template free approaches for modelling the target structure. Successful prediction groups such as Jones-UCL [82] and Skolnick-Zhang [62] have in-house methods that cater to template modelling as well as template-free modelling, while some groups such as VENCLOVAS [83] focus solely on performing well in one category.

Based on the quality of the prediction models submitted for each target sequence, the CASP assessors would then classify each target into an appropriate category, for the purpose of comparison between the quality of the models and the subsequent derivation of conclusions of the state-of-the-art for that category. Comparative Modelling (CM) targets have been further classified into 'easy' and 'hard' subcategories, where templates of 'easy' CM targets can be found using BLAST and templates of 'hard' CM targets can only be found using PSI-BLAST. Fold Recognition (FR) targets have also been subclassified into FR/H and FR/A categories. The FR/H category describes those targets that are evolutionarily related to their templates, while the FR/A category consists of the targets whose templates are not clearly related by evolution (analogous folds) and thus harder to detect. The classifications however are not to be treated as mutually exclusive categories because in truth, there is very little difference between CM and FR/H categories and between the FR/A and New Fold (NF) categories.

The following sections describe some of the issues facing the 3 different main categories in CASP.

1.3.4.1 Comparative Modelling

This section highlights some of the issues that are associated with comparative modelling approaches. As mentioned in Section 1.3.1, comparative modelling methods make use of structural templates as starting points for the derivation of the prediction model.

The steps of comparative modelling are as follows:

- Template recognition and sequence alignment
- Modelling of the structurally conserved regions
- Modelling of the structurally divergent regions
- Modelling of side-chains
- Refinement of model

Template recognition is typically performed by running a PSI-BLAST search with the target sequence on the structure database. For each target, single templates or multiple templates can be selected as starting points to model the target structure. Irregardless of the number of templates used, correct alignment of the target sequence to the template sequences is crucial in producing accurate prediction models. This is because a correct target sequence alignment to the structural template is the basis for the correct transfer of backbone information from the template to the prediction model.

Multiple template information can be utilized in ways that would improve beneficial over single template information. VENCLOVAS, one of the best performing groups in the Comparative Modelling (CM) category in CASP6, successfully used multiple template information for modelling CM targets [83]. According to Venclovas, the relatively easy CM targets benefit more from using multiple templates than for the harder CM targets. The key to their successful approach in CASP6 relies on their focus in getting the correct alignments by using a consensus approach to assess the reliability of sequence structure alignments.

For the modelling of the backbone of the target protein, the MODELLER tool [84] is widely used by prediction groups in CASP. MODELLER allows for the homologous regions of the target protein when given the sequence to template alignment, as well as the modelling of loops that are not covered by the templates. An alternative homology modelling tool is NEST from the Jackal protein structure modelling package [85].

For the modelling of side chains, the SCRWL program [86] has proven to be an effective tool and is also widely used by CASP prediction groups.

Refinement of the model is necessary to bring the model closer to the target structure, and is highlighted as one of the future areas of improvement in comparative modelling. In truth, the need for model refinement is not only restricted to CM targets, but extends to prediction models from the fold recognition and template free approaches too. In the context of comparative modelling, it has been highlighted by the assessors of CASP that models remain closer to to their templates than the target structures in terms of RMSD [43], and this bottleneck remains as one of the main challenges for future CASP prediction groups in the comparative modelling category.

1.3.4.2 Fold Recognition

In CASP, fold recognition targets are defined as protein domains whose templates cannot be found from PSI-BLAST but are present in the structure databases. Because of such a definition, what qualified as fold recognition targets in CASP 10 years ago are comparative modelling targets now. This is due to an increase in the sensitivity of sequence similarity search tools e.g. PSI-BLAST, as well as an increase in size of the structure and sequence databases. Both fold recognition methods and comparative modelling methods seek to identify templates for the modelling of the target sequence, differing only in the search methodology.

Unlike the comparative modelling approach where the main steps of sequence searches and template alignment do not differ much between the various comparative modelling methods, there is more room for diversity in the methodologies of fold recognition techniques. The principal aim is to select the best sequence-to-structure fit from a set of candidate folds. Energy functions of different types, profile-profile comparison methods, and neural networks are some of the methodologies that can be used in a fold recognition method. Some of the more established methods, to name a few, are mGenTHREADER [36, 87], 3D-PSSM [88] and FFAS [89]. These fully automated

fold recognition methods also have the capabilities for full scale genome annotation, which aims to bridge the gap between the number of sequences with unknown structures and the number of sequences with known protein structures in newly sequenced genomes.

A competitive fold recognition method has the following features, namely

- It must have a up-to-date fold library with which the target sequence is to be evaluated against.
- It is fully automated, if it is to be used for genome annotation.
- Sequence profile information is used in some way, depending on the method, for the scoring of candidate folds.

In CASP6, credit is given to prediction groups separately for the FR/H and FR/A targets. For the FR/H group, meta-servers did well, and this is made possible by the automated nature of several leading fold recognition methods which meta-servers can easily make use of. David Baker and co-workers did well for the FR/A targets, which fall into the realm of template free modelling, and this will be discussed in Section 1.3.4.3.

Besides CASP, there was LiveBench [90], which was an experiment that tested the performance of automated fold recognition servers, including meta-servers, using newly released PDB entries. The LiveBench experiment complemented CASP in the sense that it provided a platform for fold recognition servers to measure their performance against one another in the period between the CASP experiments. The performance measure used by LiveBench is the MaxSub score [77]. In LiveBench-8, the last LiveBench experiment whose results were published, the meta-servers generally performed better than the non meta-servers. LiveBench served as an useful testbed for research groups who have developed new fold recognition techniques and would want to see how their new techniques perform against other published methods.

In CAFASP4, a new category known as the Model Quality Assessment Programs

(MQAP) is introduced, that aims to assess the quality of models generated by fold recognition methods. MQAP methods are a logical extension to the process of fold recognition, where candidate folds have already been selected, and what remains is a further assessment of which of the candidate folds might be the best. Some of the MQAP methods in existence are MODCHECK [91], ProQ [92], Victor/FRST [93] and Solvex [94].

The MQAP methods, in principle, can be extended to evaluate candidate folds, or decoy structures, produced by template free modelling methods. The difference between evaluating models produced by fold recognition methods and evaluating models produced by template free modelling methods is that the latter models are likely to be less 'protein-like' with steric clashes and therefore effective MQAP methods would have to incorporate appropriate checks during the assessment of such models.

The next section describes the advances and issues in the template free category of protein structure prediction.

1.3.4.3 Template-Free Modelling

In CASP, template free modelling, also known as *de novo* structure prediction, applies to New Fold (NF) targets where no template exists in the structure databases. FR/A targets, whose structures have no common evolutionary origin to their templates, are also considered in the assessment of prediction models in the New Fold category. As mentioned in Section 1.3.1, template free methods can consist of fragment assembly methods and lattice-based methods.

For the past few CASP experiments, David Baker and co-workers have set the standards in the template free category. Their non-automated Rosetta method [95] has consistently distinguished itself from the rest of the methods. In CASP6, the latest version of the fragment assembly method FRAGFOLD [82] from Jones-UCL group has also proved competitive in the NF category. TASSER [62], an automated server that uses a combined approach of lattice models, fold recognition and fragment assembly

methods, is also competitive in the New Fold category.

The later generation of fragment assembly methods have proven to be a more successful approach than the older lattice-based methods. A competitive fragment-based template free method has the following features, namely

- It must have a representative set of fragments that can be used to build candidate structures.
- It has an effective guiding energy function that builds the candidate structures from the fragments.
- It should sample as wide a conformational space as possible during the fragment assembly process.
- It performs clustering to select the most representative structure

The fragment assembly process is repeated to yield large numbers of structures for each target protein, so as to sample as wide a conformational space as possible and therefore increase the chances of building a near native structure. These large number of candidate structures are often known as decoys. Decoy selection is the next step, and this is typically done using MQAP methods as well as clustering [96, 97].

Recent advancements in template-free modelling, apart from the design of effective energy functions, also focussed on the increase of the conformational sampling of structure space, as well as the high resolution refinement of low resolution near-native decoys [95, 98]. It has been suggested by Baker and co-workers [98] that high performance computing is important for carrying out a vast conformational search to identify the most promising near-native low resolution decoy structures, which are then subjected to high resolution refinement protocols to bring these decoy structures closer to the native structure.

In this thesis, a novel means of using machine learning to perform decoy selection, also known as decoy discrimination, is proposed. More specifically, neural networks are used to learn a decoy discrimination function by using positive and negative training

examples in the form of native structures and simulated decoy structures respectively in the training process. Chapter 2 describes the basic methodology and Chapter 3 extends the methodology by using sequence profile information in the decoy discrimination process.

The next section gives an overview of machine learning approaches that have been used in protein structure prediction.

1.4 Machine Learning in Protein Structure Prediction

A variety of machine learning algorithms has been used in various fields in bioinformatics, such as biological sequence analysis, microarray data analysis and protein structure prediction. In this section, the discussion is focused on machine learning techniques used for protein structure prediction.

In protein structure prediction, there are the problems of secondary structure prediction and tertiary structure prediction. Secondary structure prediction is a simplified 1D-representation of the problem of predicting 3D structure from sequence; instead of predicting the coordinates, a state (of either helix, strand or coil) is assigned to each residue in the sequence. Protein secondary structure prediction is a field that has probably seen all kind of techniques being applied to it, not least machine learning. The earliest techniques use information theory [21, 22]. Nearest neighbour methods [31], inductive logic programming [99], neural networks [14, 15, 24] and support vector machines [100] have also been attempted in secondary structure prediction. Of these, methods that performed best use neural networks. Section 1.2 had given a review of secondary structure prediction.

Protein tertiary structure prediction has also seen its share of machine learning tools being applied to it, commonly Hidden Markov Models (HMMs) and neural networks. It is more difficult to rephrase the tertiary structure prediction problem into one in which a machine learning tool can be applied to. A general prediction problem needs

output labels to assign to positive and negative training (and test) examples, and this can be difficult when the goal is to predict the 3D coordinates of a structure. Nevertheless, for fold recognition, neural networks have been applied successfully, as in GenTHREADER [36]. Contact map prediction can be viewed as a 2D variant of 3D structure prediction. Prediction of protein contacts, introduced in CASP2 in 1996, has also been attempted using various machine learning algorithms such as neural networks [101, 102].

Some other prediction problems that are related to protein structure prediction is the prediction of solvent accessibility. Solvent accessibility is a property of residues in the protein sequence that indicates the extent of exposure to the solvent. Neural networks have also been applied to the prediction of solvent accessibility [103]. Another type of prediction problem is the assignment of domain boundaries in protein sequences. Neural networks have been used to predict domain boundaries from sequence information alone [104, 105].

There are other structural related areas such as protein-protein interactions and functional prediction of protein sequence that have also seen the application of machine learning techniques. The field of systems biology involves the modelling of biological entities, large and small, and machine learning tools may play a role in this area along with mathematical modelling.

The next section gives an introduction to neural networks, since it is used in this thesis.

1.5 Introduction to Neural Networks

This section gives a short introduction to neural networks. A description of neural networks is first given, followed by a discussion of the applications of neural networks in areas of biological research.

1.5.1 Background

The motivation of using artificial neural networks in a computational paradigm [106], or simply neural networks as they are called, comes from the biological neural networks that function in the brain. Biological neural networks are built of complex webs of interconnected neurons. For example, the human brain is made up of a complex dense network of about 10^{11} neurons. Each biological neuron is inhibited or excited via connections to other neurons. Together, the complex network of neurons in the human brain can process information, such as facial recognition, in order of milliseconds. Such a powerful biological paradigm of processing information motivates computer scientists to design corresponding parallel computational architectures, in the form of artificial neural networks, for purposes of pattern recognition and distributed processing tasks.

Neural networks have been used successfully in several domains, such as credit card fraud detection, autonomous vehicle steering, and handwriting recognition. Figure 1.3 shows a simple feedforward neural network architecture. In Figure 1.3, there is a set of

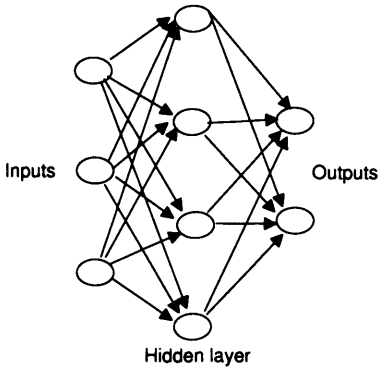
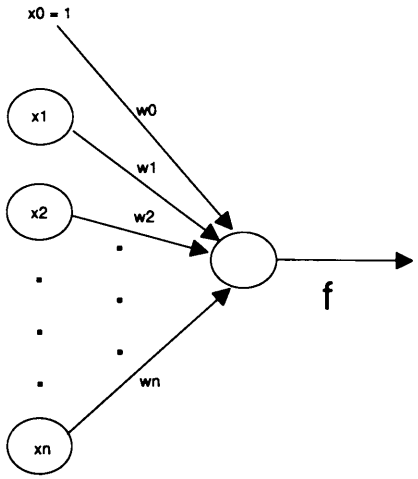


Figure 1.3: A typical neural network architecture

input neurons, followed by a hidden layer of neurons, and output neurons. The output neurons represent the target values which the neural network application are trying to predict, and the input neurons represent feature information associated with the corresponding target values. The hidden layer of neurons is inaccessible to the neural network user, and represents the internal architecture of the network. There are sets of

connections between the neurons in the different layers. Each of these connections has an associated weight value that signifies the inhibition or excitation of that particular connection.

Each of the neurons in the hidden and output layers has a transfer function that is a linear or nonlinear function of the various weights and the values of the preceding neurons associated with the corresponding weights. The transfer functions of the neurons in the same layer are typically identifical, although this is not strictly necessary. There are no transfer functions associated with the input neurons. Figure 1.4 shows the sigmoidal transfer function of a neuron.



$$g = \sum_{i=0}^{n} w_i x_i \tag{1.4}$$

$$f = \frac{1}{1 + e^{-g}} \tag{1.5}$$

Figure 1.4: Sigmoidal Transfer function of a neuron

In Figure 1.4, *x0* is the bias which allows the neural network to find a solution in weight space that does not go through the origin. In the neuron in Figure 1.4, the sum of the product of the various weights and inputs is calculated before being fed into a sigmoidal function which constrains the output value to range between 0 and 1. Such a sigmoidal transfer function is frequently used in various neural network applications.

1.5.2 Neural Network Training

Before a neural network can be of use, it has to undergo a learning or training phase where it learns the association of input patterns and their corresponding target outputs. An error function is usually defined as a function of the output values of the network. Effectively, during training, the neural network is undergoing the process of adjustment of the weight values of all its connections, so as to minimize the error function. A common training algorithm for the weights of the neural networks is the backpropagation algorithm [106].

An important aspect of neural network training is generalization [107]. Generalization refers to the ability of the neural network to perform classification or prediction tasks well on previously unseen data. Both the choice of training patterns and the length of time for training affect the ability of the neural network to generalize. A separate validation dataset can be used to evaluate the error of the network during each training step so as to be able to improve the generalization by restricting the length of time used for training. Typically, training stops when the error evaluated on the validation dataset starts to increase. It is also vital that the patterns between the training dataset, validation dataset and test dataset be dissimilar to one another. Dissimilarity is context dependent, and in the case of protein structures, the patterns in each dataset should be non-homologous to patterns in the other two datasets. A neural network model that is unable to generalize well has effectively learnt the noise patterns in the training data.

One common problem of neural network training is that the process of the search of the optimal solution in the high dimensional weight space can get stuck in local minima. Nevertheless, neural networks are still popular due to their simplicity of use and ability to obtain reasonable solutions to practical problems.

1.5.3 Areas of Biological Research

Neural networks have been used as prediction algorithms in several areas of biological research, such as protein secondary structure prediction. Qian and Sejnowski [108] are the first to attempt secondary structure prediction using neural networks. They used a two-layer neural network for secondary structure prediction and a second network for filtering the outputs of the first neural network. Subsequently, Rost and Sander used evolutionary information derived from multiple sequence alignments in their landmark PHD secondary structure prediction [109]. Later algorithms, such as PSIPRED [15] and SSPro8 [14], also use neural networks.

Apart from secondary structure prediction, neural networks have also been used in other aspects of protein structure prediction, such as the prediction of contact maps [101, 102, 110], solvent accessibility values [103], protein domain boundaries [104, 105], local propensities of secondary structure such as beta-turns [37, 38], and fold recognition [36].

Besides the prediction of protein structures, neural networks have also been used in other areas of biological research, such as the detection of codons in DNA sequences [111], classification problems in microarray data experiments [112], as well as modelling of genetic regulatory networks [113].

1.5.4 Use of Neural Networks in this Thesis

In this thesis, the Neural Network Toolbox of Matlab 7 [114] is used to implement the neural networks for training and testing. Section 1.6 describes the research problem of decoy discrimination that is to be tackled in this thesis.

Table 1.1 shows the list of neural network training algorithms provided by Matlab, and which are used in the implementation of the neural networks. Further details of how these algorithms are used in the implementation would be discussed in Section 2.3.4. The training and test data used in conjunction with neural networks would also be discussed in Section 2.3.1.

No.	Matlab name	Description
1	traingd	normal gradient descent
2	traingdm	gradient descent with momentum
3	traingda	gradient descent with adaptive learning rate
4	traingdx	gradient descent with momentum and adaptive learning rate
5	trainscg	conjugate gradient descent
6	trainbfg	quasi-Newton method
7	trainlm	Levenberg-Marquardt method

Table 1.1: List of network training algorithms

1.6 Decoy Discrimination Using Machine Learning

In this thesis, I propose a machine learning approach to the decoy discrimination problem in the context of template free prediction methods. More specifically, neural networks are used for the decoy discrimination problem. Chapter 2 presents an approach by which neural networks are used for decoy discrimination, and describes the different input features that are used for the neural networks, the network training issues involved, and how decoy training examples are derived. Publicly available decoy datasets are used for testing.

Chapter 3 expands on the ideas in Chapter 2 by using evolutionary information as additional inputs to the neural network methods for decoy discrimination. Here, there are two novel aspects in such an usage of evolutionary information for decoy discrimination. First, the idea of using evolutionary information in decoy discrimination with current energy functions has not been exploited until very recently when Lin and co-workers [115] used binary profiles with pairwise potentials of mean force for decoy discrimination. Secondly, the use of evolutionary information, combined with machine learning, for the purpose of decoy discrimination has never been attempted before, and in Chapter 3, such an approach is shown to be successful in discriminating the native

structures from decoy structures.

1.7 Organization of this Thesis

Chapter 1 gives a brief literature survey of the field of protein secondary and tertiary structure prediction.

Chapter 2 introduces the novel method of using machine learning, more specifically neural networks, for the decoy discrimination problem. The way in which the decoy discrimination problem is represented as a machine learning problem is discussed. Various input features to the neural networks are experimented. This chapter also discusses the results of testing on publicly available decoy datasets, and compares it with the pairwise potentials of mean force method.

Chapter 3 is an extension of the methods developed in Chapter 2. The novel idea of using evolutionary information in decoy discrimination is presented. This chapter details on how sequence profiles can be used in the context of neural networks to improve the decoy discrimination process.

Chapter 4 gives the conclusions of this thesis, and suggests some future extensions to the ideas presented in this thesis.

Chapter 2

Discrimination of Decoys

This chapter describes the first part of the work in this thesis. Here the importance of decoy discrimination in the context of New Fold methods is described. The novel hypothesis of using neural networks to try and discriminate native structures from non-native structures, using simulated decoy distributions and differing amounts of information (as input features) such as pairwise distances and solvent accessibility, is presented. This hypothesis is then tested on various publicly available decoy datasets, and the results are compared to those obtained by using the pairwise potentials of mean force method.

2.1 Overview of Decoy Discrimination

The methodology used for tertiary structure prediction of a target sequence, in cases where 3D folds similar to the tertiary structure cannot be found in existing protein structure databases, is known as the template-free approach, and is referred to as the New Fold category in CASP experiments. In New Fold methods, there are no existing templates that can serve as starting points for approximating the structure of the target sequence. Effective template search methods such as multiple sequence alignment and structural alignment tools are therefore not directly applicable in New Fold methods.

In the absence of guiding templates, fragment assembly methods in the New Fold category typically approximate the 3D structure of a target sequence by joining together preselected short peptide fragments of varying lengths. Some fragment assembly meth-

ods that have seen CASP participation are ROSETTA [54] and FRAGFOLD [116]. The ROSETTA method uses Bayesian scoring energy functions as the guiding energy function; the FRAGFOLD method uses pairwise potentials of mean force to guide fragment assembly.

2.1.1 The Need for Decoy Discrimination

In the above mentioned fragment assembly methods, thousands of candidate models (referred to as decoys) are generated. These candidate models are produced in large numbers so as to increase the chances of producing a model that most closely resembles the actual native structure. By closest resemblance, it is meant that the decoy has a very low global RMSD (Equation 1.3 on page 44) to that of the native structure. A value of zero RMSD to native is highly implausible for any decoy generation methods, and any decoy with $\leq$ 1Å RMSD to native can be treated as a very good prediction of the native structure.

The reason why so many decoys are generated as candidates, as opposed to the possibility of generating just one candidate structure, is because the energy functions used in the assembly process are imperfect. Therefore thousands of models are generated to cover as widely as possible the conformational sample space of the 3D model of a protein structure. The natural consequence is that an extra step is then required to select the most plausible (lowest RMSD to native) model from the thousands of decoys.

Therefore two issues exist in the development of a New Fold fragment assembly method. Firstly, a decoy generation method must exist that churns out reasonable candidate models in large numbers. Reasonable models are taken to mean models that fulfil basic requirements such as the avoidance of steric clashes between neighbouring atoms, and are compact etc. David Baker and colleagues outlined 4 requirements for a good decoy dataset for testing decoy discrimination methods [117]. These are

- A good quality decoy set should contain conformations for a wide variety of different proteins to avoid over-fitting.
- It should contain conformations close to the native structure.

- It should contain conformations that exist near the minima of a decoy discrimination function so that the function can pick them out.
- If the method used to generate the decoy dataset does not make use of information from the native structure, it can be used directly for protein structure prediction.

Secondly, the selection or discrimination process involves selecting the best model(s). It is this second process that is the focus of this chapter.

2.1.2 Selecting the Best Near-Native Decoys

A generated set of candidate decoy structures has varying RMSDs to the native structure, and the challenge is to select the lowest RMSD structure, or the top 5 lowest RMSD structures in the context of CASP participation, to represent as the best prediction of the native structure. Of course, since CASP is a blind experiment where the native structure is not known beforehand, it is impossible to calculate the RMSD during the prediction process, and therefore some form of estimate is required for the selection of a structure with the lowest perceived RMSD. The estimation is performed by the decoy discrimination methods.

A good decoy discrimination method should, in increasing order of importance, be able to

- rank the native structure as the top model, or at least among the top few models, among the decoy structures.
- associate higher scores (or lower energies) with decoy models of better quality.
- select, in the context of CASP participation, a decoy model of substantially good quality from among the decoy models to represent as the blind prediction of the target sequence.

The ranking of the native structure by a decoy discrimination method, while important, is not an essential step of blind protein structure prediction since the native structure is unknown and cannot be evaluated.

For the last 2 considerations, the definition of 'model of good quality' is not restricted to the global RMSD, since models of low local RMSDs can possibly be good quality models too. In this thesis, structural quality measures such as the TM-score [76], GDT-TS [70] and MaxSub [77] are used to judge and quantify the quality of decoy models. All 3 measures, scaled between 0 and 1 inclusive, associate higher values to better quality models.

In this thesis, the proposed machine learning decoy discrimination methods are compared to the in-house tried and tested pairwise potentials of mean force method. The pairwise potentials method has been competitive in the New Fold category for the past few CASP experiments [53, 82, 116, 118] and hence provides a stringent comparison for the proposed machine learning methods.

The measures used in this thesis for benchmarking the performances of the proposed machine learning methods against that of the pairwise potentials method are the

- Z score, which measures how many standard deviations the score of the native structure, produced by any decoy discrimination method, deviates from the average scores of all the decoy models, including the native. This measures the strength of the rank of the native structure, relative to the ranks of all the models considered.

- enrichment factor [117] in Equation 2.6 on page 103, which in the presence of the knowledge of the native structure (and hence knowledge of the RMSDs of the various decoy models), measures the extent to which a decoy discrimination method associates high scoring decoys with low RMSD structures.

- statistical comparison of the ability to select a high quality model as the best prediction, using the one-tailed Wilcoxon sign rank test [119] and the various structural quality measures.

- statistical comparison of the ability to rank high scores (low energies) with high quality models, using the one-tailed Wilcoxon sign-rank test [119], Spearman rank correlation coefficient and the various structural quality measures.

The above tests are performed on publicly available decoy datasets, such as the Tsai decoy dataset [117] and the Decoys 'R' Us suite of decoy datasets [120, 121].

The average Z scores and enrichment factors of the decoy datasets provide quantitative information on how well each method discriminates the native structure from a set of decoys, as well as how well it associates high scores with high quality models. The statistical tests provide a quantitative way to tell, at 5% significance level, if the proposed machine learning methods are better than the pairwise potentials method in terms of top model selection and the relative ranking of the decoy models by a typical decoy discrimination method.

2.1.3 Current Decoy Discrimination Methods

To make the informed guess of choosing the lowest global RMSD decoy in the absence of native structural information, several methods exist. One way is to use energy functions to evaluate the quality of the various decoy models. Here it is useful to differentiate between energy functions that are used for fragment assembly, and for decoy discrimination. If an energy function is used to build the decoy models as well as to discriminate the near-native decoy models from the non near-native ones, the energy function would not be very discerning in discriminating near-native decoys.

In this thesis, the proposed machine learning decoy discrimination method is benchmarked against the pairwise potentials of mean force method. The following subsection thereby gives an overview of the various approaches used in energy functions for the evaluation of decoy models.

2.1.3.1 Energy functions

Discriminatory energy functions can be divided into two categories, statistical and physics-based. Physics-based energy functions include OPLS force fields [57] and AMBER force fields [58], and solvent models such as the Generalized Born Solvent Model [57], while statistical energy functions include pairwise potentials of mean

force [49, 59], Bayesian scoring functions [54], atomic environmental potentials [60]. Physics-based energy functions are derived from the analysis of the fundamental physical forces of interactions between atoms, while statistical energy functions are parameterized from a set of experimental protein structures. In this section, the discussion is restricted to statistical energy functions and its various approaches.

Tanaka and Scheraga [122] are the first to suggest the idea of deriving pairwise frequencies of residues as interaction parameters for predicting protein structure. The pairwise frequencies were extracted from a set of native protein structures. Miyazawa and co-workers then included solvent terms in the estimation of interresidue contact energies of native structures [123]. Sippl [59] and others [49] obtained distributions of distances between interresidue contacts, and used a Boltzmann equation to derive net pairwise potentials of mean force. Different definitions of pairwise distance between interresidue contacts have been tried, e.g. Cα-Cα, Cβ-Cβ, N-Cα, N-Cβ, with the ones involving Cβ atoms more successful than those involving Cα atoms. This is because the propensities of pairwise distance involving Cβ-atoms incorporate the additional information of the directionality of the side chains of the contacting residues.

The distance-dependent parameters in the pairwise potentials of mean force assumes that the frequencies of each type of residue pair is independent of other types of pairs, and this was pointed out by Dill and co-workers [124]. Nevertheless, the pairwise potentials of mean force method has been used successfully in the past CASP experiments in the New Fold category [53, 82, 116, 118] and is shown in Equation 2.9 in Section 2.3.8.1.

Other distance-dependent statistical energy functions include the use of conditional probability [125], Bayesian scoring functions [54] and the use of different reference states in the Boltzmann equation [126]. Recent work by Thirumalai and co-workers also include statistical potentials that takes into account of the orientation-dependency in side chains [127]. Besides the extraction of distance propensities from the structure databases, there is also the related approach of characterizing of residue environments from native structures [46, 128]. This led to the development of environmental po-

tentials which use the propensity of different residual environments to evaluate decoy structures [60, 129].

The recent increase in the size of the structure databases has led to an improvement in the accuracies of these knowledge-based statistical energy functions because there is a much larger sample of native structures with which to fit the parameters of these statistical functions.

In this thesis, the proposed machine learning method is compared to the pairwise potentials of mean force in terms of various benchmarking measures presented in Section 2.1.2. As mentioned in Section 1.6, one of the goals in this thesis is to evaluate the feasibility of including evolutionary information in the decoy discrimination process, in the context of the proposed machine learning method. For the pairwise potentials method, the parameterization of interresidue distances along with all possible values of position-specific profiles is deemed to require too large a sample space for the current amount of data in the structure databases. The proposed method in this thesis hopes to circumvent this parameterization problem of using evolutionary information by using the machine learning approach instead.

Other approaches of decoy discrimination have also involved the use of contact maps for representing 3D protein structures (native and decoys), and the subsequent problem of discrimination of decoys from native structures has been reduced to a 2D problem of distinguishing decoy contact maps from native contact maps [130]. This is an interesting approach but suffers from the added complexity of parameter fitting during the conversion of 3D to 2D representation.

One effective way of performing the selection of the best model from the thousands of decoys is through clustering [96, 97]. Clustering involves calculating the RMSD of each decoy against all other decoys, and identifying the decoy with the most number of neighbours within a cutoff RMSD threshold, e.g. 4Å. This decoy, or an averaged structure of its most populated cluster, is then taken as the representative model. Clustering has worked well for New Fold methods in past CASP experiments.

2.1.4 Proposed Method of Decoy Discrimination

In this chapter, a novel method for decoy discrimination using machine learning is proposed. More specifically, supervised learning is performed using neural networks for the decoy discrimination problem. In a typical supervised learning problem, there are positive and negative training examples. In the decoy discrimination problem, positive training examples are in the form of native structures, and negative training examples are in the form of decoy structures, or more specifically simulated decoy structures (Section 2.2.1). The challenge here is to formulate the decoy discrimination problem into one that is suitable for encoding 3D structures as inputs into a neural network. This is described in detail below.

Given a large number of native structures, for given values of sequence separation k between 2 types of residues where k is defined as the number of residues apart along the protein sequence between 2 particular residues, the two types of residues form a particular distribution of distances. This distribution of distances formed by residues in native structures, for a given sequence separation k, is not a new idea and has been used to derive classical potentials of mean force [59] and used in threading [49]. For ease of discussion, this distribution is referred as the Native Residue Pair distribution of Distances (NRPD).

For each particular sequence separation k, there exists a distribution of distances of each type of residue pair. Taking k to range from values of 4 to 22 and treating k > 22 as one distribution, and taking into account 400 possible residue pairs, there are altogether $20 \times 400 = 8000$ distance distributions. The reason for the selection of this particular range of k is for the purpose of straightforward comparison with pairwise potentials of mean force during benchmarking. In some methods using the pairwise potentials of mean force [36, 116], a short range sequence separation is defined as $4 \leq k \leq 10$, a medium range sequence separation is defined as $11 \leq k \leq 22$, and a long range sequence separation is defined as $k > 22$.

In this proposed decoy discrimination method, for practical reasons of not being able to consider all possible values of k, distributions for separations $k > 22$ have been lumped together as one distribution. Separations of $1 \leq k \leq 3$ are also ignored.

Figure 2.1 shows one such distribution of an Alanine pair at sequence separation k=4. Figure A.1 in Appendix A shows the distributions of different types of residue pairs at the sequence separation k=6. The proteins used to derive these plots is shown in Table D.1. In Figure A.1, the different types include hydrophobic residue pairs (ALA-ALA, PRO-PRO), similarly charged residue pairs (ASP-GLU, ARG-LYS), opposite charged residue pairs (ASP-LYS, ARG-ASP) and polar residue pairs (SER-SER, THR-THR). In most of these figures, a peak at about 11Å at separation k=6 can be seen. This peak is due to the formation of the regular α helix formation which fixes the distances between the Cβ atoms of the helical residues to about 11Å at k=6.

One way of looking at the NRPD distributions is to consider the 2D distance map

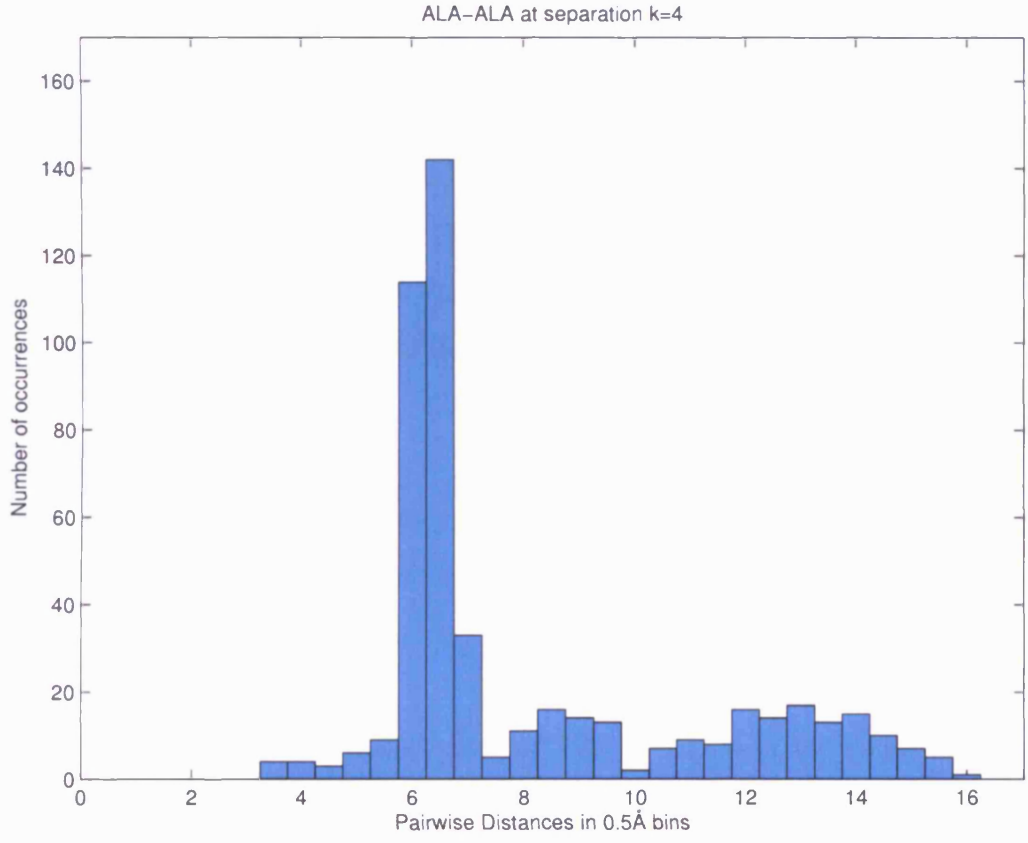


Figure 2.1: Histograms of native pairwise distances of ALA-ALA at k=4

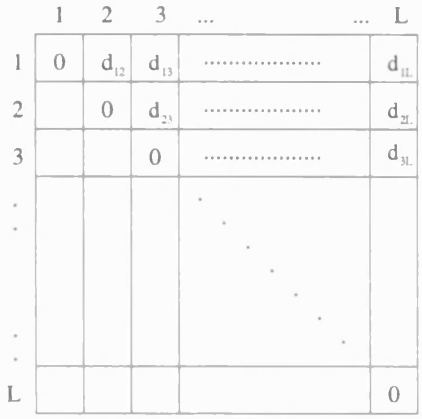


Figure 2.2: 2D Distance Map Representation of a Structure

representation of a native structure, as shown in Figure 2.2. A 2D distance map is a symmetric L x L matrix of distance values, where L is the length of the protein sequence. Each (i, j) entry contains the distance between residue i and residue j, where $1 \leq i, j \leq L$. The actual definition of pairwise distance is taken to be the distance between the corresponding Cβ atoms of the residues involved (See Section 2.3.3 for further details on the definition of distance). The sequence separation between the residue positions is thus $k = |i - j|$. Each entry in the distance map can be viewed as a sample point from a particular NRPD distribution of residue pair i and j at a particular separation k. Of course, due to the physical constraints in a real protein structure, the distance entries are not independent of one another. But in this hypothesis, it is convenient to assume independence of distance map entries, as in the case of Sippl's pairwise potentials of mean force [59] and its applications in fold recognition [49].

Next it is assumed that decoy structures have an equivalent residue pair distance distribution (DRPD) that is different from NRPD for each of the 8000 native distance distributions. Effectively, the DRPD represent distributions of non-native decoy structures.

Figures B.1 and B.2 are enhanced plots of Figure A.1, with the additional decoy distributions (DRPDs) alongside the NRPDs. These decoy distributions are obtained from the proposed sequence reversal method of simulating decoy structures from native structures (See Section 2.2.1 for more details on the sequence reversal method).

It can be seen that the helical tendencies of the pairwise residues (to form a distance of 11Å) in decoy histograms in Figure B.1 is falsely lower than the native histograms for the ALA-ALA plot. However, in the PRO-PRO plot, it can be seen that the decoy histogram of the proline pair has a falsely high peak at 11Å. These various decoy histograms provide the negative training examples for the distribution of residue pairs at separation k=6. Other similar examples can be observed in the rest of the plots in Figures B.1 and B.2.

Assuming that all 8000 NRPDs and corresponding DRPDs have been derived using a set of decoys, the 'goodness' of each decoy can then be judged in the following manner. The distance map of each decoy is first calculated. Each of the entries in the distance map is treated as sample points from a distance distribution of pairwise residues at particular values of k. It is then of interest which of the distance distributions, NRPD or DRPD, is more likely to have generated each sample point. For example, if a particular sample point (ALA, THR, k=6) has a value of 10.375Å, it is interesting to see whether the ALA-THR k=6 NRPD is more likely than the ALA-THR k=6 DRPD to have generated this sample point.

One possible manner of comparision is to use a lookup table with likelihood values calculated from histograms generated from both NRPD and DRPD. However this method is oversimplistic because distance values are continuous and histograms are essentially summaries of binned distances. Furthermore it is essential to assign a single score to each entry of the distance map, for the purpose of reflecting the likelihood of the distance sample point to have come from either NRPD or DRPD. A high score would indicate that it is more likely to have been sampled from NRPD, and a low score would indicate otherwise. Because the assignment of a single score to each distance map entry is required for a predictive essence, a single 'distribution' that takes into account both NRPD and DRPD distributions for all 8000 types of residue pair distance distributions needs to be derived.

The following section gives a detailed description of the method implementing this hypothesis, and answers the question of how scoring is performed, and how a DRPD is

derived.

2.2 Description of Method

It is assumed that there are differences in the distributions of pairwise interactions of residues in the decoy structures (DRPD), to that of the distribution of pairwise interactions of residues in the native structures (NRPD). For each type of residue pair and sequence separation k, both the NRPD and DRPD distributions have to be combined in a manner whereby a single likelihood score can be assigned to each entry of the distance map of the decoy structure concerned. This is achieved using neural networks and is further discussed in Section 2.2.2. For now, a means of estimation of the decoy residue pair distance distributions (DRPD) is required.

In order to have a DRPD, a decoy dataset for the derivation of the distributions of distances is needed. However, good decoy datasets are hard to come by, and they are also typically generated for a very small dataset of protein domains. This poses a problem because a large diverse dataset of native structures is needed to form a NRPD, and correspondingly a DRPD. Using currently available decoy datasets (produced by other research labs) for derivation of decoy distance distributions also render the decoy datasets unavailable for testing.

2.2.1 Decoy Simulation of Native Sequences

One way to get a large and unbiased decoy dataset is to create a generic decoy representation for each native protein structure. In this way, the experiments carried out are not in any way constrained by the lack of available decoy structures for native proteins. A generic decoy representation of a native sequence also does not have any dependence on any decoy generation method. However, the disadvantage remains as to whether a particular generic form of decoy representation is representative of actual decoys generated by New Fold prediction methods.

Two methods of simulating generic decoys for each native sequence are used. The first method is to take the sequence of a native structure, reverse the sequence and then thread it onto the structure. This is known as the 'sequence reversal' method. The second method is to take the distance map of each native structure, and add a small random deviate to each of the entries in the map. This is known as the 'perturbed distance map' method. These methods are further elaborated in Section 2.3.1.2.

Figure 2.3 shows a binned distribution of distances of Alanine-Alanine (ALA-ALA) residue pairs of both the domains of native protein structures and simulated generic decoys, using both the sequence reversal and the perturbed distance map method. It can be seen that both decoy distributions of ALA-ALA has a lower peak at 6.5Å, compared to that of the native distribution.

Figures B.1 and B.2, previously mentioned in Section 2.1.4, show more residue pair distance plots of reversed sequence decoy distributions (DRPDs) at separation k=6.

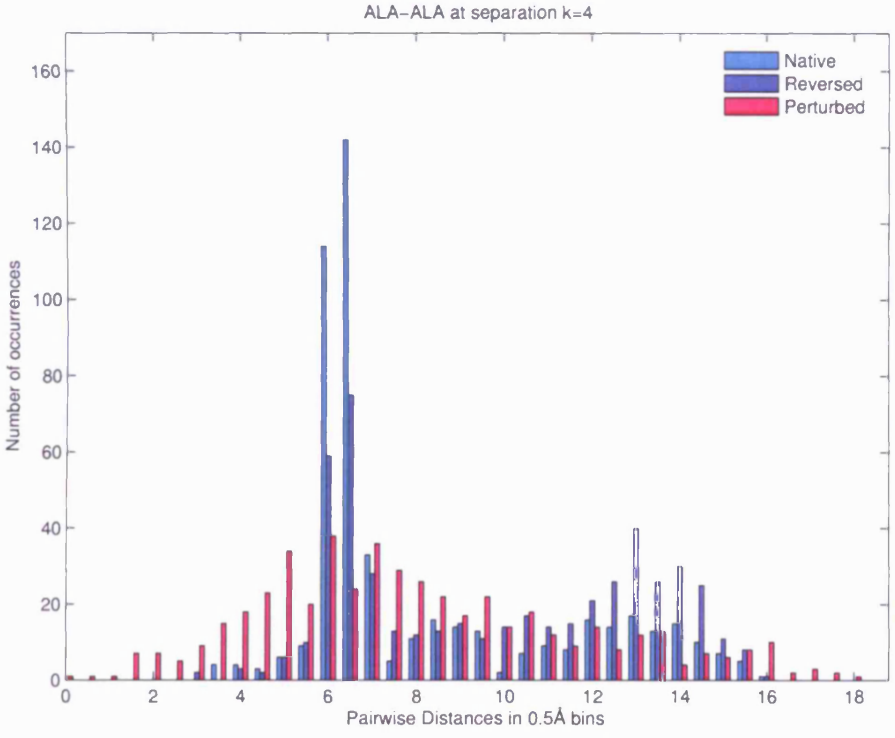


Figure 2.3: Histograms of native and simulated decoy pairwise distances of ALA-ALA at k=4

2.2.2 Machine Learning Framework

Now that a simulated decoy dataset has been selected, the focus is on how best to use both the NRPD and DRPD in some sort of machine learning method that it can be applied to each candidate decoy structure, for the goal of selecting/rejecting it on the grounds of similarity, or the lack of it, to the native structure.

Here the approach described in Section 2.1.4 is reiterated. For each decoy that is to be judged, its distance map is examined. If each entry of the map is treated as sample points from a particular distribution, it can either come from NRPD or DRPD. A single score between 0 and 1 is to be assigned to each entry in the distance map, and that score would describe the likelihood of the entry coming from NRPD. A score closer to 1 would indicate that there is a higher likelihood of the distance entry being drawn from NRPD than from DRPD. Here, it should be taken note that this is not referring to the actual probability of it being drawn from NRPD. Rather, it is the likelihood that the distance sample point is being drawn from NRPD.

To achieve this paradigm of a single score describing such a likelihood, a single functional approximation that represents an average of the NRPD and DRPD distributions is needed. This function does not describe the probability of an occurrence of a distance point. It is a function that describes the likelihood of a distance point being sampled from NRPD rather than DRPD. To implement this paradigm, all entries of distance maps of native structures are regarded as positive training examples with labels *'1'* while all entries of distance maps of simulated decoys are considered negative training examples with labels *'0'*. In this thesis, the term 'negative' is used to refer to non native-like structures.

Each decoy from a set of thousands of decoys has to be scored individually, to judge if it is a near-native structure or not. To implement this new paradigm of decoy discrimination, a likelihood measure needs to be developed which takes into account both the NRPD and DRPD distributions of a given pairwise residue contact at a particular sequence separation k, say for ALA-ALA at k=4.

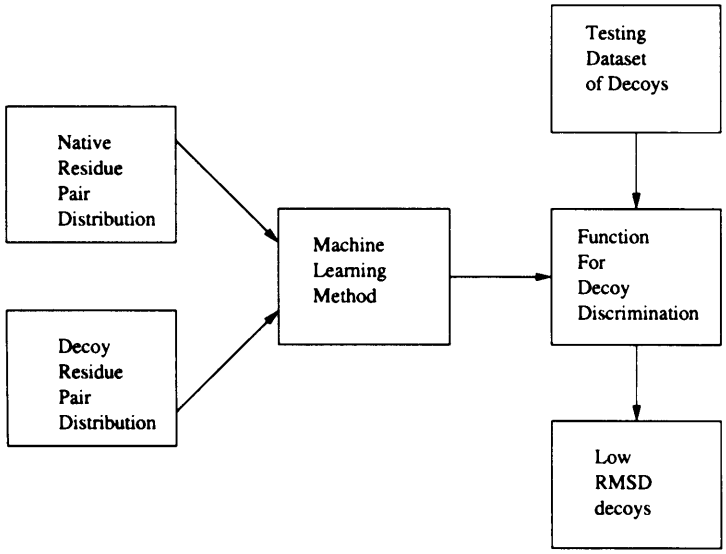


Figure 2.4: Machine Learning Framework

A machine learning framework is used, as shown in Figure 2.4, where the training data comes from the set of native and simulated decoy structures, where NRPD and DRPD is modelled from respectively. In order to use these 2 distributions (NRPD, DRPD) in a predictive manner, all input instances from native structures are assigned the output label of *'1'*, and all input instances from simulated decoy structures are assigned the output label of *'0'*. All input instances are in the form of $(R1, R2, k, d)$ where each pair of residues has a sequence separation $k \geq 4$, and has a pairwise distance d calculated from the coordinates of the structure. In a distance map representation, d would be the entry of the distance map with features $(R1, R2, k)$. The positive and negative training vectors would therefore come from each entry in each distance map where $k \geq 4$.

The 'Testing Dataset of Decoys' in Figure 2.4 are 'real' decoys generated by some fragment assembly method, e.g. FRAGFOLD. Each decoy structure is decomposed into a number of test vectors in the form of $(R1, R2, k, d)$ where $k \geq 4$. The vectors are then used as inputs to the function produced by the 'Machine Learning Method' and an output score will indicate the likelihood of each vector. The output scores of all test vectors of the decoy structure are then averaged and this mean score will give a measure for the entire decoy structure of how native-like it is. Hopefully low RMSD decoys are among the high scoring decoys evaluated by the function, as shown in Figure 2.4.

A neural network is used to represent the 'Machine Learning Method', one for each separation value k and one for $k > 22$, to derive these likelihood scores. In this work, the strategy of using neural networks for decoy discrimination, using native structures as positive training examples and simulated decoy structures as negative training examples, is to present to the networks sufficient examples of features of correct and erroneous protein structures in the hope that they can learn to pick out native or near-native structures, based on these features, from a set of decoys.

The 'Machine Learning Method' is divided into 20 neural networks, one for each k between 4 and 22 inclusive and one for $k > 22$, for performance reasons. Each input value to each network then consists of a vector ($R1$, $R2$, d). The output of the network is either *'1'* or *'0'*, depending on whether the vector comes from a native protein structure (output label *'1'*) or decoy protein structure (output label *'0'*). Because there are two ways of representing the decoy distributions (namely the sequence reversal method and the perturbed distance map method), there would be a separate set of neural networks for each decoy simulation method. After training, the neural network would yield a curve that is averaged over the training instances, as shown in Figure 2.5, which summarizes the likelihood that an input vector ($R1$, $R2$, d) comes from a native structure. The sequence reversal method is used in Figure 2.5 for the generation of decoy structures.

Appendix C shows more of these plots for each of the native and decoy histogram plots in Appendix B. These plots would be discussed further in the next section (Section 2.2.3).

The method of the neural network training is such that a number of native training instances, in the form of ($R1$, $R2$, d), with outputs *'1'* and a number of decoy training instances, in similar form, with outputs *'0'* are presented to the network. The values of d presented are floating point numbers, and are not binned. (Figures 2.1, 2.3 and 2.5 have the distance values rounded in 0.5Å bins solely for display purposes). It is hoped

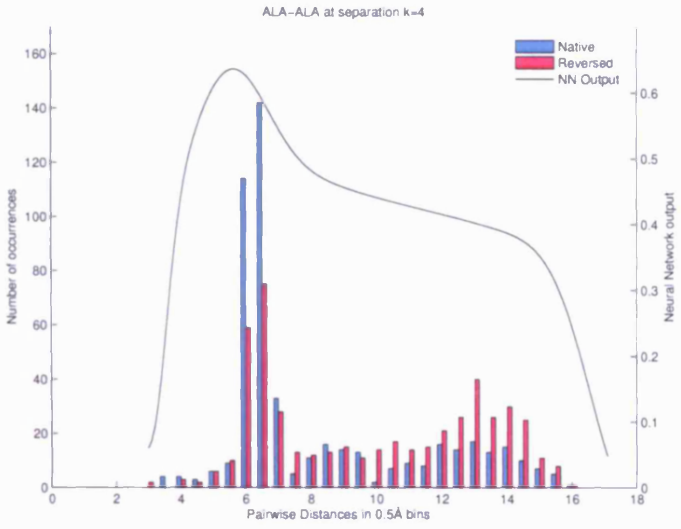


Figure 2.5: Network outputs averaged over native and decoy distance distributions for ALA-ALA at k=4

that at the end of training, the output of the network, which has a range of between 0 and 1 inclusive, will indicate the likelihood to which a new input instance belongs to a near-native decoy structure. Theoretically, the higher the value of the likelihood, the more likely the instance is to come from a near-native decoy structure. Table 2.1 shows some examples of the training instances fed into a k=4 neural network. Section 2.3.4 discusses the neural network training issues involved.

2.2.3 Interpretation of Network Output

A total of 20 neural networks, one for each sequence separation $4 \leq k \leq 22$ and one for $k > 22$, are each trained with thousands of positive and negative training examples. For each network of a particular separation k, these positive and negative training examples are taken from a subset of SCOP domains, as described in detail in Section 2.3.1. Effectively these training examples are the distance map entries of a particular sequence separation k in the form of ($R1$, $R2$, d). The output labels would be *'1'*s and *'0'*s for maps belonging to native protein structures and simulated decoy structures respectively.

Such a method of neural network training is unlike conventional neural networks

Protein	Type	Residue1	Residue2	Separation	Distance	Output Label
1a32	Native	ALA	SER	4	4.765	1
1a32	Native	TRP	GLY	4	6.367	1
1a32	Native	THR	TYR	4	8.894	1
...	...	...	...	...	...	..
1a32	Decoy	PHE	TYR	4	7.894	0
1a32	Decoy	LEU	ILE	4	9.664	0
1a32	Decoy	MET	LEU	4	10.032	0
...	...	...	...	...	...	..

Table 2.1: Example of k=4 training input instances and their output labels

used in pattern recognition problems. In a typical handwriting recognition problem, the neural network is presented with input instances of positive and negative training examples of the letter *'A'* in vector form, complete with labels *'1'*s and *'0'*s. A proper training dataset in such pattern recognition problems should not be inconsistent, which means that the training dataset should not have identical input instances with different labels.

In the approach discussed here, similar input instances can have different labels. In truth, it is probably difficult to have two input instances with identical values of floating-point distances because the input vector ($R1$, $R2$, d) has continuous values for d. However the idea is that in neural network training, near-similar input instances with opposite labels would be presented to the gradient descent algorithm. It is the functional depiction of the larger quantity of input instances of a particular class (say, native with output label *'1'*) 'winning' against the smaller quantity of input instances of the other class (decoy) that the neural network is expected to achieve, instead of the usual nonlinear functional approximation over a high dimensional input space in the case of the handwriting recognition application. In a general sense, for the decoy discrimination method described here, it is still functional approximation using neural networks, albeit an adaptation for the specific problem of decoy discrimination.

As an illustration, in Figure 2.5, distances between 6.25Å to 6.75Å are grouped in

the 6.5Å bin. Strictly speaking, the distances presented to the neural network are not binned; they are binned in this case for the purpose of convenience of visual presentation. For the k=4 neural network using the sequence reversal method, there are 142 and 75 positive and negative training instances respectively. The averaged network output for ALA-ALA at k=4 has a value of about 0.59 at d=6.5Å. This means that a distance entry with the value of 6.5Å would have a likelihood score of 0.59 ($\geq$ 0.5), which suggests that it is more likely to be from a near-native decoy structure. In the same figure, for d=14Å, the magenta bar (decoy histogram) is higher than the cyan bar (native histogram). This means that the number of negative training examples exceed the number of positive training examples for d=14Å. The network output is about 0.385 for this particular distance of 14Å, which suggests that they are more likely to have come from a non near-native decoy.

In Appendix C, plots from the k=6 neural network show the extent to which the network is combining the frequencies of the native and decoy histograms of residue pairs into a single score. The topology of the network is shown in Figure 2.14. Figures C.1 and C.2 show 8 out of 400 possible residue pairs whose native and decoy histograms have been averaged by the k=6 neural network. Positive and negative training examples used by the network in the form of (*R1*, *R2*, *d*, k=6) are extracted from the training dataset (See Section 2.3.1 for details on the training dataset).

The network output at a particular distance d tries to reflect the ratio of the native to decoy histograms. For example, for the ALA-ALA plot in Figure C.1, there is a peak of 11Å where the number of positive training examples is about twice that of the negative training examples. The lower peak of about 4Å is due to the fact that there are no negative examples at 4Å. Recall that the network uses continuous values in the training examples as illustrated in the *'Distance'* column in Table 2.1, and not binned frequencies as shown in the plots. Binned frequencies are used in the plots for display purposes only.

In each of these 8 plots, it can be seen that the line plot has higher peaks in distance bins where there are more positive examples than negative examples and vice

versa. In general, the network plots are smoother than the ratio of the native to decoy histograms because they represent the function of the training examples of ALL 400 residue pairs and hence the line plot of each residue pair does not closely reflect the different ratios of the number of positive to negative examples in each distance bin.

Here it is important to take note that the network output is NOT a probability of an entry in the distance map coming from the distributions of NRPD or DRPD. Rather it is the likelihood (taking values between 0 and 1 inclusive) that the distance is being derived from the distributions of NRPD. Likelihood values of 0.5 or greater indicate that the NRPD distribution is more likely, and suggest that that distance is more likely to have come from a near-native decoy. However that only pertains to one entry. All entries need to be considered when the whole decoy structure is being judged to be near-native or not.

At this point, it is worthwhile to reiterate that one single network of a particular sequence separation k is responsible for achieving the likes of averaged network outputs shown in Figures C.1 and C.2 for all 400 types of residue pairs. Each network, for a particular sequence separation k, effectively encodes the behaviour of how a particular distance is likely to score for each possible residue pair. With this approach, the novelty is that evolutionary information in the form of multiple sequence alignment profiles can be included into the machine learning method of decoy discrimination proposed in this thesis (Chapter 3).

Figure 2.6 shows the general architecture of the proposed neural network method of decoy discrimination. Each structure, native or decoy, is decomposed into subsets of input data to each of the 20 neural networks representing each separation value k (or $k > 22$). The subsequent result matrices can then be combined into a single score, and it is then desired that the native structure has the highest score among the decoy structures.

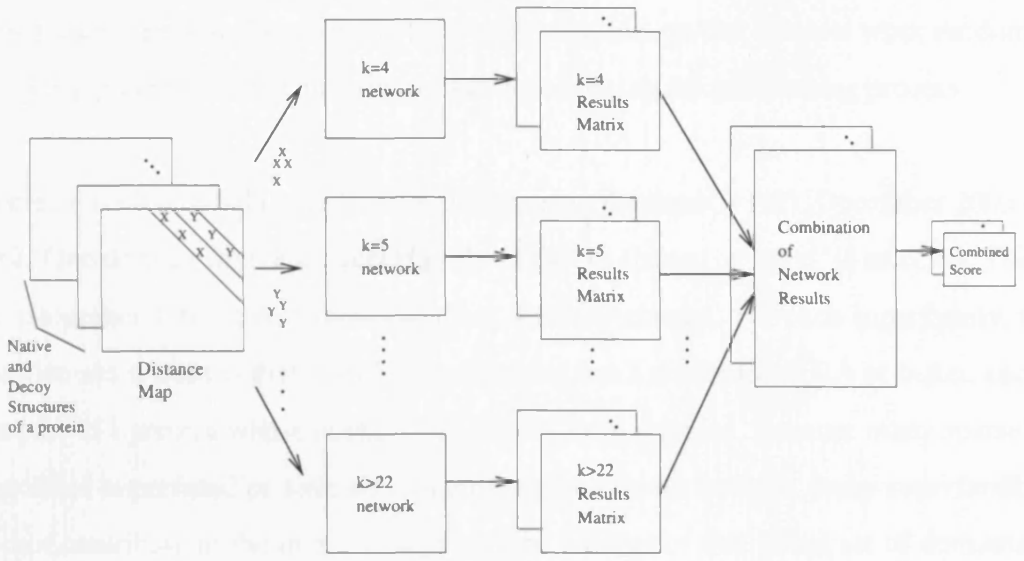


Figure 2.6: Proposed neural network method of decoy discrimination

2.3 Materials and Methods

This section illustrates how the training and test datasets are obtained, discusses the neural network training issues involved, and describes the testing framework used for the proposed decoy discrimination method.

2.3.1 Training, Validation and Test Datasets

A large and diverse dataset of protein domains is required for training, validation and testing. The subset of proteins reserved for training is used to derive the representation of the native distance distributions and the simulation of the decoy distributions. The validation subset is necessary in the context of neural network training, and the remaining subset is used for preliminary testing. In this work, this dataset is obtained from the SCOP database [131] and subsequently partitioned into 3 parts. The initial unpartitioned dataset is referred to as the 'initial dataset' in the remainder of this section.

The proteins in the initial dataset are chosen to be structurally non-homologous to one another. This means that no two pairs of protein domains in the initial dataset is structurally similar, in the context of SCOP's classification method. This is done to facilitate the partitioning of the validation and test datasets from the initial dataset. If all pairs of protein domains are non-homologous, there is no cause for worry of homolo-

gous proteins existing between the training and validation/test datasets when randomly assigning proteins to the validation/test datasets during the partitioning process.

To create such a initial dataset, the SCOP domain database (v1.65, December 2003) is used. One domain from each superfamily of SCOP classes *'a'* to *'d'* is selected. There are altogether 1095 superfamilies in these 4 SCOP classes. For each superfamily, the first domain is chosen that is an X-ray structure, has a resolution of 2Å or better, and is not part of a protein whose domain has already been selected. Because many sparsely-populated superfamilies have no domains whose criteria are met, many superfamilies do not contribute to the initial dataset and the number of this initial set of domains is 740.

However, 28 proteins in this initial set of domains share the same superfamily with at least one of the proteins in the decoy datasets, namely the Baker dataset and the Decoys 'R' Us suite of decoys (See Section 2.3.2 for more details on the decoy datasets used for testing). Therefore these 28 proteins are excluded from the initial dataset. Furthermore, 265 proteins from this initial set (after the 28 proteins have been excluded) have less than 10 alignments in the multiple sequence alignments after PSI-BLAST [25] is run, and these are excluded from the training dataset. Strictly speaking, there is no need to exclude these 265 proteins from the training dataset when training with single sequences (without evolutionary information). However to facilitate the comparision of results obtained from single sequence information to results obtained from the inclusion of evolutionary information during benchmarking, it would be more precise if the set of training data is being kept constant. The final set of protein domains, after both exclusion steps, has 475 proteins.

This final dataset of 475 protein domains is then divided into 3 parts, namely 60% for the training dataset, 20% for validation and 20% for preliminary testing. Tables D.1, D.2 and D.3 show the training set of 285 protein domains, the validation set of 95 protein domains and preliminary test data of 95 protein domains respectively. All 3 datasets have mixtures of secondary structural classes, as shown in Table 2.2. From Table 2.2, it can be seen that the $\alpha\beta$ proteins are about twice the number of α-only

Dataset	Number of proteins			
	All	**α-only**	**β-only**	$\alpha\beta$
Training	285	58 (20.4%)	59 (20.7%)	168 (58.9%)
Validation	95	18 (19.0%)	25 (26.3%)	52 (54.7%)
Preliminary Test	95	22 (23.2%)	17 (17.9%)	56 (58.9 %)

Table 2.2: Structural compositions of the training, validation and preliminary test datasets

and β-only proteins because the $\alpha\beta$ class in Table 2.2 consists of proteins from SCOP classes *'c'* and *'d'*, while the α-only and β-only proteins come from SCOP classes *'a'* and *'b'* respectively.

Here two types of testing are performed. The first level consists of preliminary testing, where the decoy proteins are simulated by randomizing the 95 sequences in the preliminary test dataset. The acid test is the second type of testing, where 'real' decoys are the ones generated by some fragment assembly methods. Any future references to the term 'test dataset' refers to this second type of testing.

The training dataset is used to train 20 neural networks, one for each sequence separation k from 4 to 22 inclusive, and one for $k > 22$. Each generic decoy representation (sequence reversal method and perturbed distance map method) has its own training done independently of the other representation.

2.3.1.1 Preliminary Test Dataset

The preliminary test dataset in Table D.3 consists of 95 proteins of different structural compositions, as shown in Table 2.2. The purpose of the preliminary test dataset is to simulate random structures, that is, structures with their residues randomly shuffled along the sequence. The aim is to provide a first level test to see if the neural networks of both types of simulated negative training examples, namely the sequence reversal and perturbed distance methods, can successfully distinguish the native structure from

random structures.

For each of the 95 proteins, 50 random structures are created by shuffling the sequence and then superposing the randomly shuffled sequence to the 3D structure. The neural networks are then used to test, as described later in Section 2.3.7, to see if the native structure has the highest likelihood among all the random structures. The Z score (Equation 2.5 on page 102) is also used to measure the extent to which the native structure is recognized. Section 2.4.1 shows the results of the discrimination of the 95 native structures in this preliminary test dataset.

2.3.1.2 Simulated Decoy Datasets

The training dataset for NRPD are the native protein domains taken from SCOP. The DRPD needs to be approximated by creating generic decoy structures. There are two ways to do this.

In the sequence reversal method, decoys are modelled by using native structures with their sequences reversed. This renders most of the side chain atoms meaningless, apart from Cβ atoms. In fact, the distance between residues is defined as the distance between corresponding Cβ atoms (Section 2.3.3). For non-glycine residues occupying glycine positions after the reversal of sequence, virtual Cβ atom positions are calculated according to Equation 2.3.

The sequence reversal method is a reasonable first approximation to near-native decoys compared to structures with purely random sequences because some information regarding sequence order and neighbourhood compositions of residues are retained in the reversed sequence. Here it should be pointed out that even though certain protein domains in the training dataset have domain boundaries (as shown in Table D.1) and do not span the entire polypeptide chain, the reversal of sequence is done on the entire polypeptide chain, and the final 'reversed' decoy sequence is then extracted using the original domain boundaries. This is to ensure that the original chunk of 3D structure is retained, with only the identity of the sequence modified.

In the perturbed distance map method, the distance map of each native structure is perturbed by adding a random distance component to each distance map entry. A Gaussian distribution of mean μ=0Å and standard deviation σ=2Å is applied to each entry in the distance map. The final distance map may or may not belong to a realistic protein structural model whose 3D coordinates can be derived from the 2D coordinates in the map. But such a representation can be viewed as simulating decoy models with steric clashes.

In Figure 2.5, it can be seen that there are no training examples, positive or negative, for distances below 3.5Å. Here, pseudo negative training examples with distance values from 0Å to 3.5Å in steps of 0.5Å are included for each residue pair. The purpose is to allow the neural network to classify the output values of these 'impossible' (due to steric clashes) distance values to belong to the negative class. The same is repeated for the upper 'impossible' distance values from the largest native distance value $d_{largest}$ to $d_{largest}$+50Å, in steps of 0.5Å, for each residue pair.

2.3.2 Decoy Datasets for Testing

Decoy discrimination methods require decoy datasets for testing. In this thesis, the well-known Tsai decoy dataset [117] from David Baker's laboratory and the Decoys 'R' Us suite of decoys [120, 121] are used for testing the effectiveness of this decoy discrimination method. Table 2.3 shows the decoy datasets in the Decoys 'R' Us suite. All these decoy datasets are freely available for download from the web. Each decoy dataset consists of a native protein structure and its corresponding set of decoy structures, which are generated according to the author's unique decoy generation method. The number of proteins listed in Table 2.3 may not correspond to that in the website [121] because some proteins in the datasets are already obsolete.

No.	Name of decoy set	Number of proteins	Average number of decoys per set	Reference
1	4state_reduced	6	665	[132]
2	lattice_ssfit	8	2000	[133]
3	fisa	4	1432	[54]
4	fisa_casp3	4	1432	[54]
5	lmds	10	439	[133]
6	lmds_v2	10	439	[133]
7	semfold	6	12900	[134]

Table 2.3: Decoys 'R' Us suite of decoys

Decoy Dataset	Number of proteins				
	All	α-only	β-only	$\alpha\beta$	Others
4state_reduced	6	3 (50%)	0 (0%)	1 (16.7%)	2 (33.3%)
lattice_ssfit	8	3 (37.5%)	0 (0%)	4 (50%)	1 (12.5%)
fisa	4	4 (100%)	0 (0%)	0 (0%)	0 (0%)
fisa_casp3	4	4 (100%)	0 (0%)	0 (0%)	0 (0%)
lmds	10	3 (30%)	1 (10%)	2 (20%)	4 (40%)
lmds_v2	10	2 (20%)	1 (10%)	3 (30%)	4 (40%)
semfold	6	4 (66.7%)	0 (0%)	2 (33.3%)	0 (0%)

Table 2.4: Structural compositions of Decoys 'R' Us suite of decoys

Table 2.4 show the structural compositions of the decoy datasets. Proteins of SCOP classes *'a'* and *'b'* are classified in the α-only and β-only columns respectively, while proteins belonging to SCOP classes *'c'* and *'d'* are considered as $\alpha\beta$. The *'Others'* column refers to small proteins (SCOP class *'g'*) and peptides (SCOP class *'j'*).

Protein	Class	Number of Decoys	Protein	Class	Number of Decoys
1a32	α	1400	1mzm	α	1442
1ail	α	1399	1orc	$\alpha\beta$	1399
1bq9	β	1400	1pgx	$\alpha\beta$	1399
1cc5	α	1399	1ptq	$\alpha\beta$	1399
1cei	α	1400	1r69	α	1399
1csp	β	1399	1tif	$\alpha\beta$	1399
1ctf	$\alpha\beta$	1453	1tuc	β	1400
1dol	$\alpha\beta$	1400	1utg	α	1399
1hyp	α	1400	1vcc	$\alpha\beta$	1400
1lfb	α	1399	1vif	β	1399
1msi	β	1399	5pti	β	1399

Table 2.5: Baker decoy dataset of 22 proteins

The Baker decoy dataset consists of 22 X-ray protein structures, and each protein has a set of about 1400 decoys. It is used by Jerry Tsai and colleagues for testing various physical energy functions. Only the X-ray structures of the original dataset are included. Table 2.5 shows the list of protein domains and the number of decoys for each domain. One advantage of using the Baker decoy dataset is that it has several proteins of different secondary structural compositions, namely α-only, β-only and $\alpha\beta$ structures. The quality of the decoys in the Baker dataset is also higher than most of the decoy datasets in the Decoy 'R' Us suite, in the sense that they are more native-like and hence harder to discriminate from native structures.

2.3.2.1 Description of the Decoy Datasets

The 4state_reduced decoy dataset was created by Britt Park and Michael Levitt in 1996 [132]. For 8 small proteins of between 54 and 76 residues long, several thousand decoys are generated from near-native models of the native structures by the exhaustive enumeration of 10 residues with the four different states of (ϕ, ψ) in a dihedral angle model. These 10 residues are made up of 5 consecutive residue-pairs, and are

positioned between secondary structure elements of the near-native model. The ensemble size is narrowed down by discarding models with radii of gyration higher than a specified threshold and removing conformations which have inter-residue contacts of ≤ 3.5Å greater than a specified number. These decoys are native-like because they are generated from near-native models, and the enumeration of conformations takes place mainly in the loop residues.

The Tsai decoy set, fisa and fisa_casp3 decoy datasets are generated by David Baker and co-workers [54]. The method consists of a simulated annealing procedure to assemble native-like conformations using a variety of fragments. The fragment set is obtained from unrelated structures with similar local sequences, and the conformations are assessed using Bayesian scoring functions. The Tsai decoy dataset is of higher quality than the rest, due to the fact that the decoys undergo an extensive minimization procedure where each decoy structure is perturbed slightly and assessed to see if the perturbation yields lower energies.

The lattice_ssfit decoy dataset was created by Yu Xia, Ram Samudrala and co-workers in 2000 [135]. The method consists of complete enumerations of conformations using a simple tetrahedral lattice model, where a subset of conformations is selected for all-atom model generation using predicted secondary structure information. A subset of these all-atom generated models is then evaluated using a knowledge-based atomic level energy function.

The semfold decoy dataset was generated by Ram Samudrala and Michael Levitt in 2002 [134], using 6 *ab initio* targets that are predicted to have helical content. For each protein, decoys are generated starting from an all-atom conformation with idealized torsion angles for helices and extended default values for non-helical residues. New conformations are generated by iteratively perturbing the existing conformation of an arbitrary single residue. Trajectories are generated using a Monte Carlo algorithm with simulated annealing, and a genetic algorithm for search. The conformations are then evaluated using a variety of energy functions.

The lmds and lmds_v2 decoy datasets were created by Keasar Chen and Michael Levitt [136]. The authors proposed that since a global optimization of an energy function that approximates the actual free energy landscape is difficult to obtain, one possible method to generate decoy datasets that include a near-native structure is to obtain good representations of the space of all local minima in the energy landscape of protein folding. These local minima are assumed to contain the native structure as well. To reduce the search space for all local minima, an existing energy function [137] is modified to represent broad regions of local minima. An iterative procedure of decoy generation and parameter fitting of the modified energy function is also used.

2.3.2.2 Quality of the Decoy Datasets

This section shows the quality of the various decoy datasets, in terms of the distribution of the RMSDs of the decoys of individual proteins. This is important because the performance of decoy discrimination methods depends on the quality of the decoy dataset. For instance, a decoy dataset with a high percentage of decoys with high RMSDs would be easier for most decoy discrimination methods to discriminate the corresponding native structure, while a better decoy discrimination method can identify native proteins than other methods when tested on a decoy dataset with many low RMSD decoys.

Table 2.6 shows the RMSDs of all the decoys of the various proteins for each dataset at 5%, 25%, 50%, 75% and 95% percentiles. It can be seen that the decoys in each dataset have widely varying RMSDs. From Table 2.6, it can also be seen that the Baker decoy dataset has the highest quality in terms of the number of low RMSD decoys. The Baker decoy dataset, being the second largest dataset, has a RMSD of 3.818Å at 5% percentile. This means that it has about 1500 decoys with RMSDs of 3.818Å or lower. It can be seen that the Baker decoy dataset has a higher proportion of low RMSD decoys, compared to other smaller datasets such as lattice_ssfit.

Decoy Dataset	Total Number	Percentile				
		5%	25%	50% (Mean)	75%	95%
Bakerdecoy	30860	3.818	6.290	8.595	10.503	12.413
4state_reduced	3996	2.033	4.031	5.398	6.441	7.530
lattice_ssfit	16000	7.177	8.787	9.186	10.930	12.818
fisa	2000	3.728	4.847	7.359	10.098	12.139
fisa_casp3	5991	6.886	9.7253	11.606	13.454	17.177
lmds	4336	3.563	5.280	7.795	9.631	11.423
lmds_v2	1200	4.086	6.210	8.679	11.022	13.256
semfold	78214	6.852	9.400	10.707	11.680	13.032

Table 2.6: RMSD distributions of all decoy datasets

Figures 2.7 and 2.8 show the RMSD distributions of set of decoys of each of the 22 proteins in the Baker dataset, grouped into classes of secondary structure compositions. The group of α-only proteins is separated into 2 plots for purposes of clarity.

Figure 2.9 shows the RMSD distributions of the 4state_reduced and lattice_ssfit decoy datasets, while Figure 2.10 shows the RMSD distributions of the fisa and fisa_casp3 decoy datasets.

Figures 2.11 and 2.12 show the RMSD distributions of the lmds and lmds_v2 decoy datasets respectively. There are 10 proteins in each of the lmds and lmds_v2 datasets. For the sake of clarity, two plots are shown for each of these figures. Figure 2.13 shows the RMSD distributions of the semfold decoy dataset.

2.3.3 Definition of Pairwise Distance

The pairwise distance between two residues is taken as the distance between the corresponding Cβ atoms. In the case of glycine, the Cβ atom is approximated from the associated Cα, N and C coordinates (r_i values) using the following equation as used in [132]. Two unit vectors x and y are defined below.

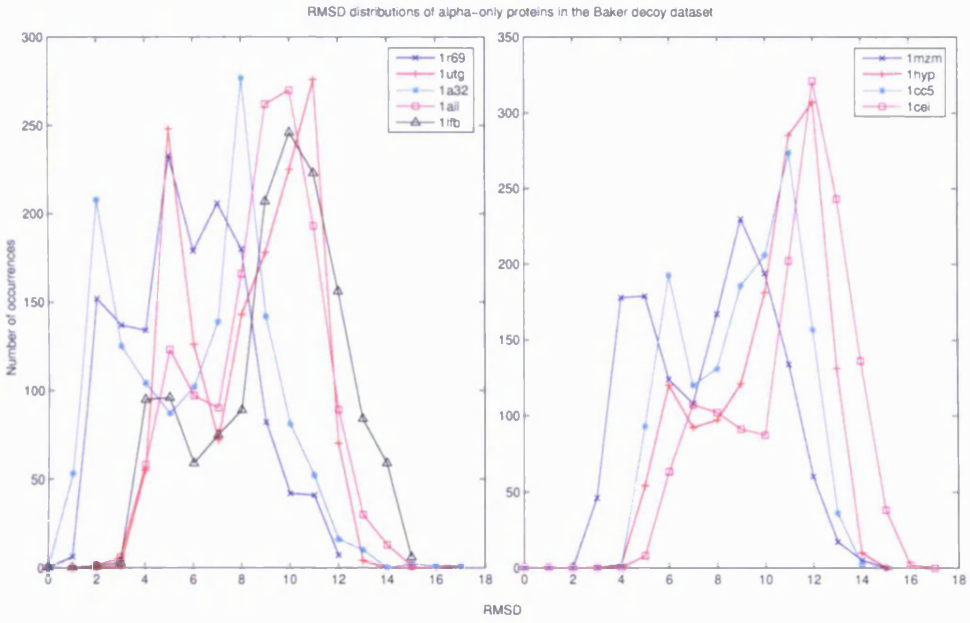


Figure 2.7: RMSD distributions of α-only proteins in the Baker decoy dataset

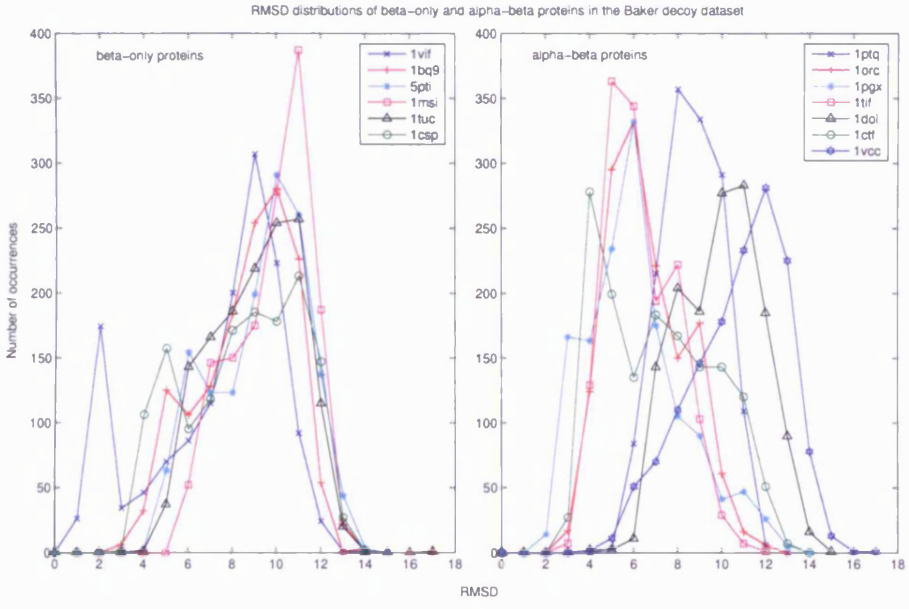


Figure 2.8: RMSD distributions of β-only and $\alpha\beta$ proteins in the Baker decoy dataset

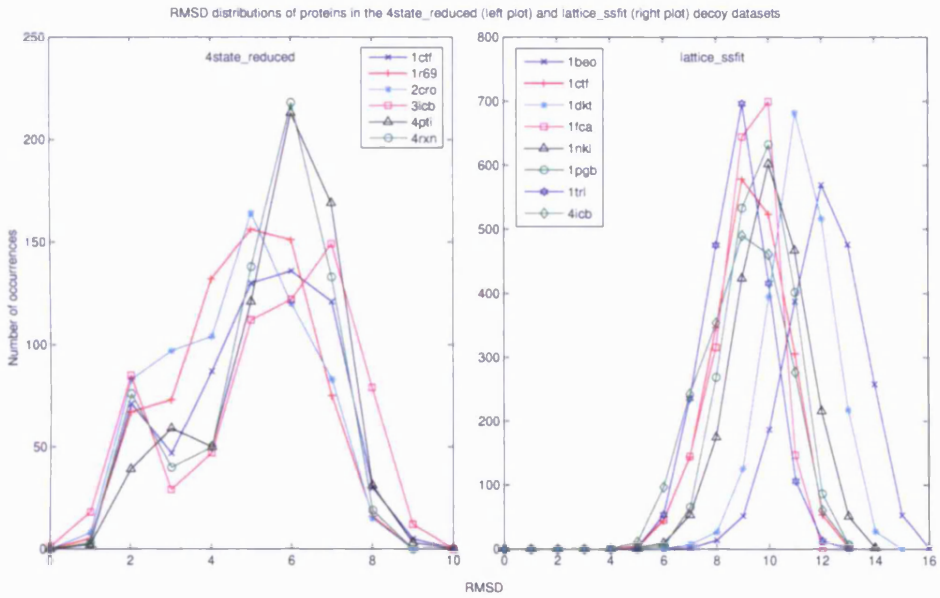


Figure 2.9: RMSD distributions of the 4state_reduced and lattice_ssfit decoy datasets

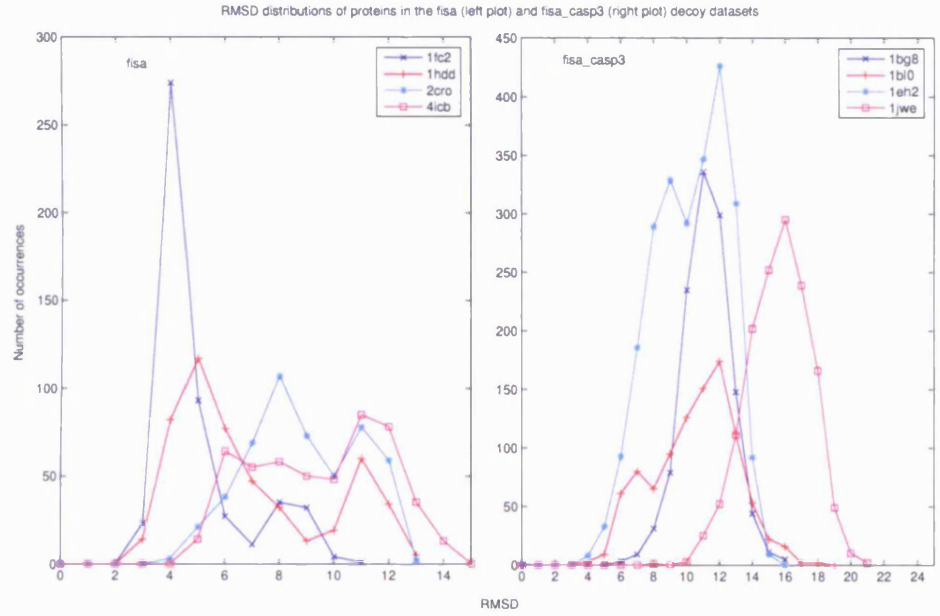


Figure 2.10: RMSD distributions of the fisa and fisa_casp3 decoy datasets

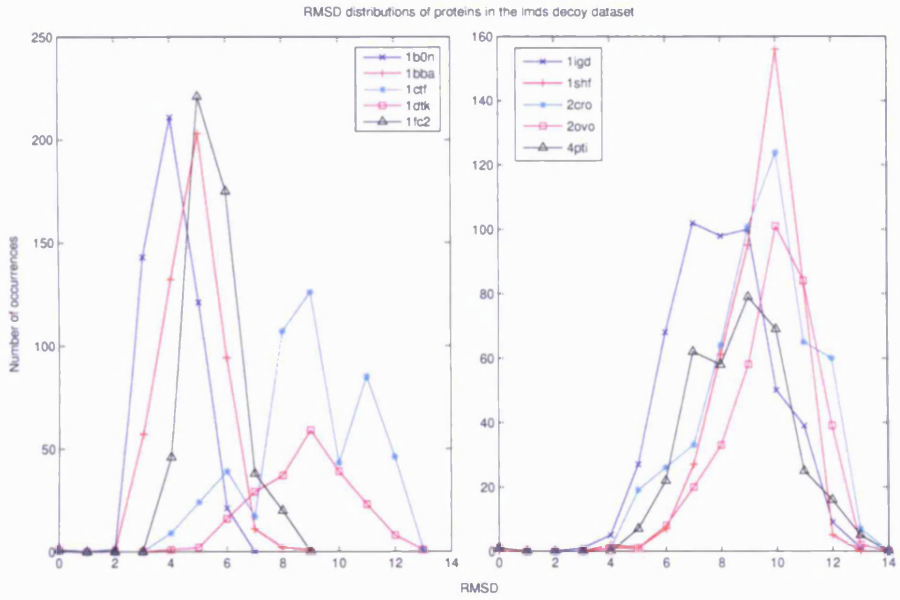


Figure 2.11: RMSD distributions of the lmds decoy dataset

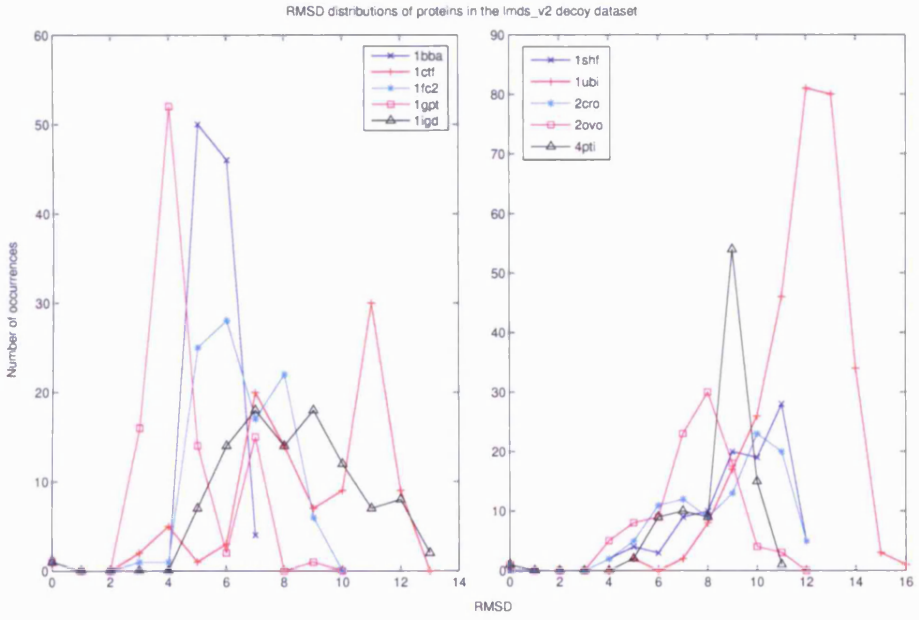


Figure 2.12: RMSD distributions of the lmds_v2 decoy dataset

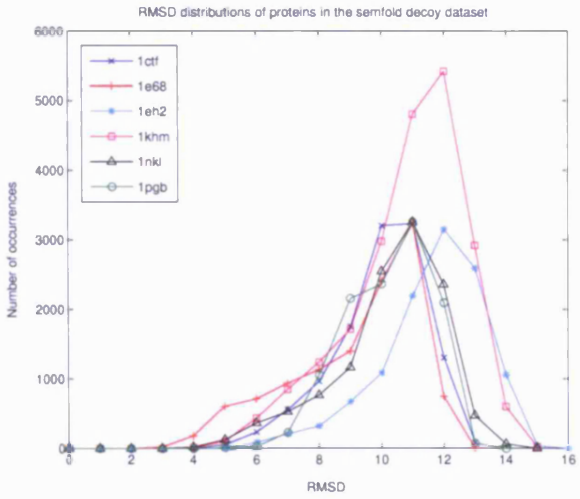


Figure 2.13: RMSD distributions of the semfold decoy dataset

$$\mathbf{x} = \frac{(r_i - r_{i-1}) + (r_i - r_{i+1})}{|(r_i - r_{i-1}) + (r_i - r_{i+1})|} \tag{2.1}$$

$$\mathbf{y} = \frac{(r_i - r_{i-1}) \times (r_i - r_{i+1})}{|(r_i - r_{i-1}) \times (r_i - r_{i+1})|} \tag{2.2}$$

The position of the Cβ atom, r_β is then calculated using the following equation, where l is the distance of the Cβ atom from the Cα atom and set to 1.53Å while θ is set to 37.6 degrees.

$$r_\beta = lcos\theta\mathbf{x} + lsin\theta\mathbf{y} \tag{2.3}$$

This approximation of Cβ atoms in glycine residues is performed on the training, validation, preliminary test, as well as the decoy test datasets.

2.3.4 Neural Network Training Issues

A machine learning method is proposed that outputs a score, when presented with a pair of residues (*R1*, *R2*), sequence separation *k* and distance *d*, indicating the extent to which residues *R1* and *R2*, at sequence separation *k* and *d* Å apart, belongs to a native structure. Averaged over all possible residue pairs in the sequence over all possible

values of k, it is hoped that the mean score would be a likelihood measure, indicating how near-native the decoy structure is.

This machine learning method is implemented in the form of a neural network. The inputs of the neural network would be a particular pair of residues (*R1*, *R2*), sequence separation k and distance d. For performance reasons, each integer value k ranging from 4 to 22 inclusive, and one for $k > 22$, is divided into 20 different neural networks representing each value of k.

The training dataset is taken from Table D.1. Each of the 20 networks has its own set of training data, in the form of positive and negative training examples. Positive training examples are taken from the pairwise distances of native protein structures. In the sequence reversal method, negative training examples are taken from native structures with their sequences reversed. In the perturbed distance method, the native pairwise distances have a random deviate added (Section 2.3.1.2 describes the details for both methods). For the $k > 22$ network, only 1 out of every 100 (positive and negative) training examples are used during neural network training due to the lack of sufficient memory resources.

Figure 2.14 shows the neural network topology. A two-layer feedforward neural network is used. The first layer comprises of 41 neurons, with 20 neurons for each residue. These 40 neurons take the binary value *'0'* or *'1'*, to indicate the presence or absence of a residue type. Only 1 of the 20 neurons for each residue can take the value of *'1'* for each training example. For example, the vector [1 0 0 ... 0] represent an Alanine residue. The 41^{st} neuron represents the distance between the two residues, and accepts a floating point number.

Each of the 41 neurons in the input layer is connected to all the neurons in the hidden layer. The ***W1*** weight matrix is of size $41 \times N_H$ where N_H is the number of neurons in the hidden layer. Several transfer functions are experimented for the first layer (see Table 2.7), while the transfer function in the second layer is the typical linear function. In Figure 2.14, *f(s,**W1**)* and *g(f,**W2**)* indicate the transfer functions in the input and

hidden layers respectively, while ***W1*** and ***W2*** are the 41 $\times$ N_H input weight matrix and N_H $\times$ 1 hidden weight matrix respectively, and s refers to the input examples. The 1 $\times$ N_H input bias vector ***b1*** and the 1 $\times$ 1 hidden bias vector ***b2*** are also shown in the diagram.

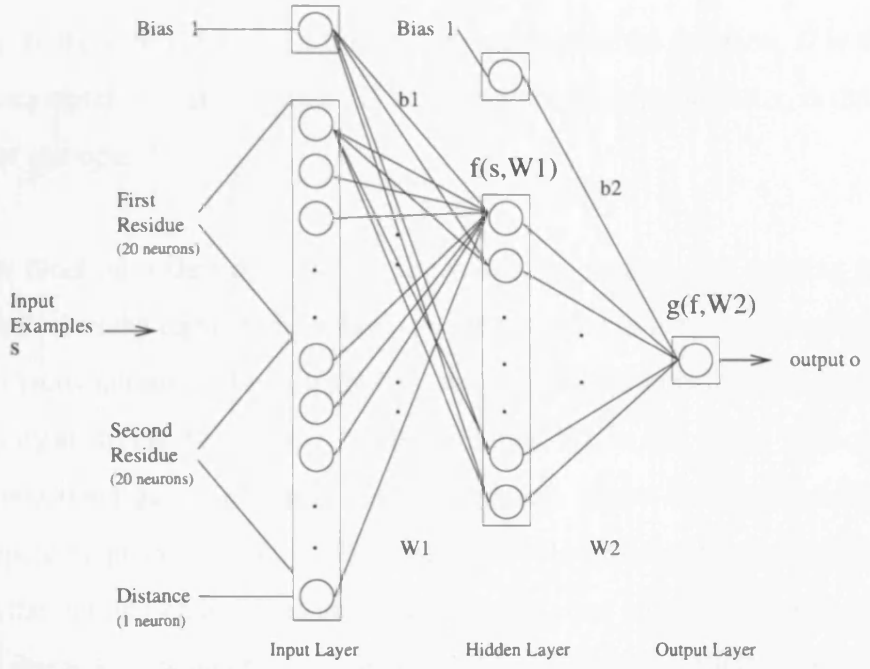


Figure 2.14: Neural Network Topology

2.3.4.1 Training Procedure

As mentioned in Section 1.5.4, the Neural Network Toolbox of Matlab 7 [114] is used for the training and testing of the various neural networks. The toolbox is useful because it has standard functions for creating and designing neural networks, performing error minimization and batch training. There is also provision for the use of validation data to prevent overfitting (Section 2.3.4.2).

For each network of a particular separation k ($4 \leq k \leq 22$, $k > 22$), the entire set of positive and negative training examples of 400 residue pairs is shown to the network, which then calculates the total error of the training examples, using the transfer functions of both layers. In neural network terminology, this is the batch mode of training. The error function used is the Mean Square Error (MSE), which is shown in

Equation 2.4,

$$e_D = \frac{1}{2}\sum_{d \in D}(t_d - o_d)^2 \tag{2.4}$$

where e_D is the total error of the training examples after the iteration, D is the set of training examples, t_d is the output label of the training example d, and o_d is the network output for example d.

The error function is then evaluated on the validation dataset. The training process is terminated when the error on the validation dataset starts to increase, compared to that of the previous iteration. If not, a gradient descent algorithm is then used to calculate the adjustments to the weight matrices ***W2*** and then ***W1***, in that order. This constitutes one training iteration. Subsequent training iterations repeat the error calculation and weight updating process until the MSE decreases below a predefined threshold of 0.01 or when the training ends due to an increase in the error on the validation dataset. In practice, due to the formulation of this decoy discrimination method where conflicting labels of the training examples prevent the MSE from becoming too small, the training always terminates due to the increase in error on the validation dataset.

The following sections describe the validation dataset, and investigate the different transfer functions used in training a typical neural network.

2.3.4.2 Validation Dataset

A validation dataset (Table D.2) is used during the training of each neural network. It consists of 95 proteins of different structural classes, as shown in Table 2.2. The validation dataset is used to prevent the network from overfitting the training data. In theory, overfitting should not occur because the number of training examples for each neural network (in the order of 100 000) far exceeds that of the number of hidden units (in the order of 10). But the validation dataset is still used nonetheless.

No.	Matlab name	Description
1	tansig	sigmoid transfer function (Figure 2.15A)
2	radbas	radial basis transfer function (Figure 2.15B)
3	tribas	triangular radial basis transfer function (Figure 2.15C)

Table 2.7: List of transfer functions

Like the training dataset, the validation dataset is in the form of (*R1*, *R2*, *d*) for each sequence separation k, where $4 \leq k \leq 22$, or $k > 22$.

2.3.4.3 Transfer Functions of Neural Network

Several transfer functions are experimented for the first layer of the feedforward neural network. Table 2.7 shows a list of transfer functions. The k=4 and k=5 networks are used to empirically select which transfer function can yield the lowest MSE. The selected transfer function is then used in all the 20 neural networks. Here, the sequence reversal method is selected to provide the negative training examples for this simple benchmarking test. The Levenberg-Marquardt algorithm is chosen as the training algorithm for all 3 transfer functions (Section 2.3.4.5).

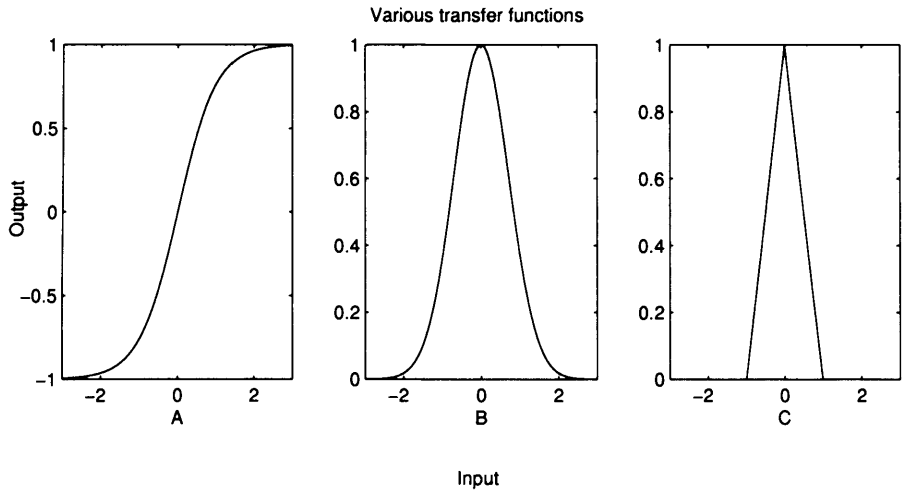


Figure 2.15: Different transfer functions used for benchmarking of the k=4 network

2.3.4.4 Transfer Function Benchmarking Results

Figure 2.16 shows the results of the MSEs achieved by the 3 transfer functions listed in Table 2.7. The errors are obtained either after 100 training iterations or after early stopping of the training process due to an increase in error on the validation dataset, as described in Section 2.3.4.2.

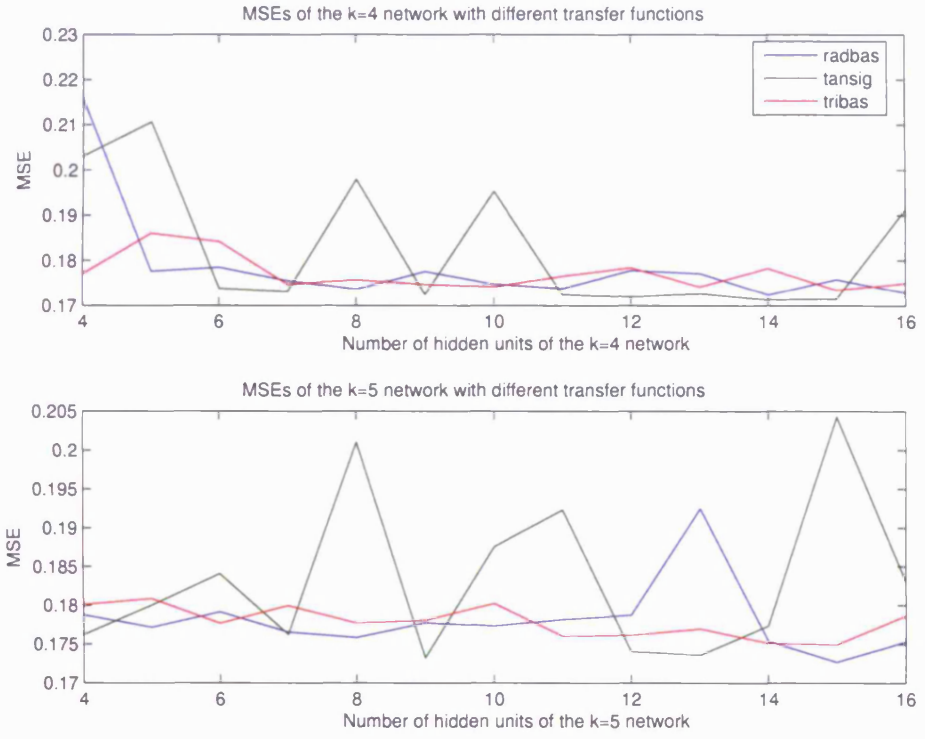


Figure 2.16: MSEs of the k=4 and k=5 networks of the various transfer functions in Table 2.7

The error performance yielded by the sigmoid transfer (tansig) function over various number of hidden units is unstable, even though it occasionally yields the lowest error among the 3 different transfer functions. The reason for its instability could be due to the fact that it is ill-fitted to model the natural shape of the output function, which is that of a bell shape, as shown in Figure 2.5. The triangular radial basis functions (tribas) and the radial basis function (radbas) are much more stable in terms of the error performance over various hidden units. Here, the radial basis function (radbas) is chosen because of its smooth interpolation nature.

2.3.4.5 Neural Network Training Algorithms

Several neural network training algorithms are experimented for searching in the high dimensional weight space for the weight vector that yields the lowest error. Table 1.1 shows the various network training algorithms tried with the k=4 neural network. The learning rate of all the algorithms is set to 0.1.

Figure 2.17 shows the results of the MSEs achieved by the various network training algorithms listed in Table 1.1 for the k=4 neural network. The errors are obtained either after 300 training iterations or after early stopping of the training process due to an increase in error on the validation dataset. The number of training iterations is set to 300. In practice, all the network training algorithms in Table 1.1, apart from traingdm, experience early stopping for this simple benchmark.

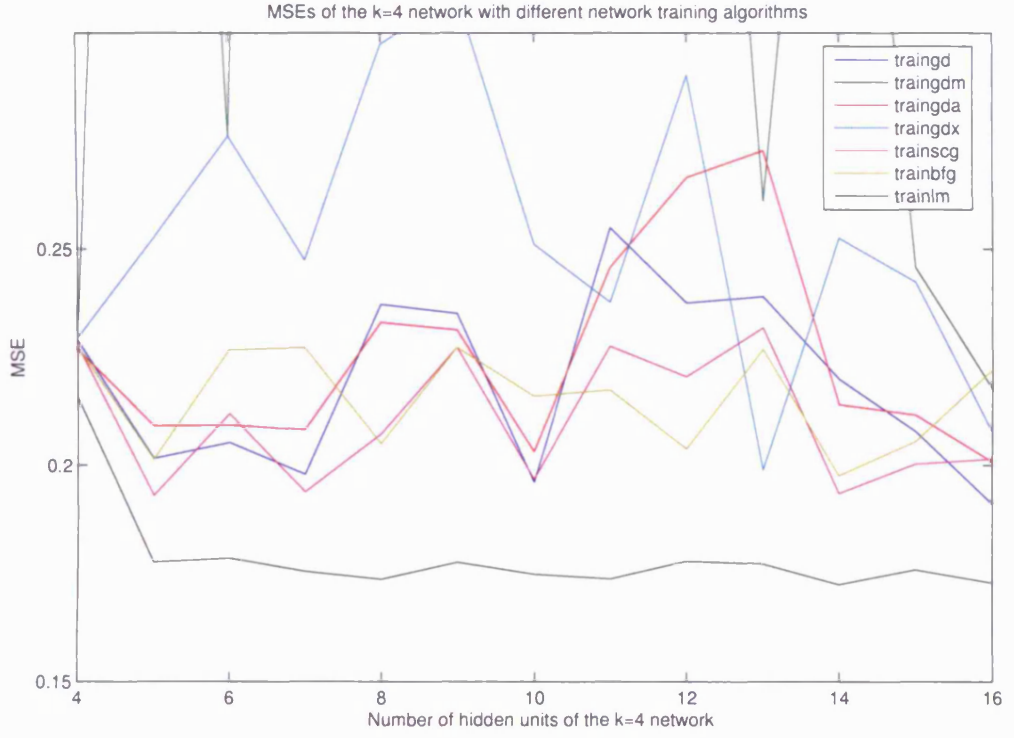


Figure 2.17: Different network training algorithms used for benchmarking of the k=4 network

It can be seen from Figure 2.17 that the Levenberg-Marquardt algorithm (trainlm) yields the lowest MSE across all numbers of hidden units and is also the most stable

algorithm. Therefore this algorithm is selected for the rest of the neural networks.

2.3.4.6 Number of Hidden Units

For each of the 20 neural networks, the number of hidden units is varied from 4 to 16 inclusive. The maximum number of hidden units attempted is 16 due to the limitation of memory resources. For each k, the network with a particular number of hidden units that yield the lowest MSE (Equation 2.4 on page 98) is chosen.

2.3.5 Test Measures

Unlike usual pattern recognition problems where the test dataset consists of clear labels (whether a character is an *'A'* or not), one way to evaluate how well the proposed decoy discrimination method performs on 'real' decoy datasets is to see how well the native structures fare among the decoys in the scoring/discriminating function. The Z score is frequently used in this aspect.

$$Zscore = \frac{S_{native} - S_{mean}}{S_{\sigma}} \tag{2.5}$$

where S_{native} is the score of the native structure produced by the proposed decoy discrimination method, S_{mean} is the mean score across all decoys, including the native structure, and S_{σ} is the standard deviation.

For each test protein, there exists a set of decoy structures with varying RMSDs. In a CASP scenario, the native structure is unknown. In a way, the Z score can only be calculated in the aftermath of CASP when the native structure, once it is known to all, is ranked to see how it fares in a particular decoy discrimination function.

Therefore another way to benchmark a decoy discrimination method is the enrichment measure. Here the focus is on assessing if the decoy discrimination method succeeds in identifying the lowest RMSD near-native structures. The enrichment factor, introduced by David Baker [117], is the proportion of low RMSD decoys in a low

energy subset of the decoy population, over the total number of low RMSD decoys in the entire decoy population. In the current context, the term 'low energy' would be replaced by 'high likelihood'.

To quantify this, David Baker uses 15% as the thresholds for the cutoff for both the low RMSD decoy subset and the low energy subset. In Equation 2.6, the enrichment is defined as the intersections of both the subsets divided by what might be expected for an uniform distribution of low-energy decoys as well as for low RMSD decoys. Values greater than one suggest that the decoy discrimination method has an enrichment over a uniform distribution [117].

$$enrichment = \frac{M_{15\%} \cap R_{15\%}}{15\% \times 15\% \times N} \tag{2.6}$$

where $M_{15\%}$ is the list of decoys with the top 15% highest scores as identified by the decoy discrimination method, $R_{15\%}$ is the list of decoys with the top 15% of lowest RMSDs, and N is the total number of decoy models.

Both the enrichment factor and the native Z score are used for testing both the proposed machine learning decoy discrimination method and the pairwise potentials method, for the sake of comparison between the two, on the various decoy datasets.

2.3.6 Statistical Tests

The Z score and the enrichment measure provide information on how well the proposed machine learning method, as well as the pairwise potentials method, can discriminate the native structure, and associate low RMSD decoys with high scoring decoys. Here, several statistical tests are proposed to evaluate if the proposed machine learning method is better than the pairwise potentials method in three different ways.

There are altogether 70 sets of decoys from the 8 different decoy datasets, and hence this constitute a sample size of 70 for the statistical tests. Each test is repeated for the various different structural similarity measures. The three different tests are the

- top model selection, where the one-tailed Wilcoxon sign-rank test is used to reject or not reject the null hypothesis that the median of the distribution of the differences in the structural similarity scores (TM-score, GDT-TS and MaxSub) of the top structure selected from the 70 sets of decoys by the two different methods is zero. The alternative hypothesis, being a one-tailed test, is that the median of the differences in the similarity scores produced by the machine learning method and the pairwise potentials method is higher than zero.

- ranking of all the decoy models to that of the structural similarity scores (TM-score, GDT-TS and MaxSub) for each of the 70 samples, where the Spearman correlation coefficient is used to calculate the rank correlation. The one-tailed Wilcoxon sign-rank test is used to reject or not reject the null hypothesis that the median of the distribution of the differences in the Spearman correlation coefficients produced by both methods is zero.

- ROC analysis, where in a machine learning essence, the various decoy models in all the 70 sets of decoys are dichotomized into 'true' and 'false' data, and the ROC curves of the various decoy discrimination methods are plotted against one another.

The following sections describe the various tests in detail.

2.3.6.1 Wilcoxon sign-rank test on top model selection

The Wilcoxon sign-rank test is used here for gauging the ability of two different decoy discrimination methods for selecting a high quality structure as its highest ranked model from a set of decoys. The 'structure of high quality' is quantified by each of the 3 different structural similarity measures, namely the TM-score, GDT-TS and MaxSub. Hence the Wilcoxon sign-rank test is to be performed for each of the 3 structural similarity measures.

The one-tailed Wilcoxon sign-rank test is used to reject/not reject the null hypothesis that the median of the distribution of the differences between the structural similarity scores of the top ranked model produced by the machine learning method and the

pairwise potentials method is zero.

The null and alternative hypotheses are as follows:

$$H_0 : M_{A-B} = 0$$

$$H_A : M_{A-B} > 0 \tag{2.7}$$

where M_X is the median of the distribution of random variable X, where A and B are the random variables describing the structural similarity scores of the 70 samples for the proposed machine learning decoy discrimination method and the pairwise potentials method respectively. A significance level of 5% is used.

In this statistical test, the one-tailed test is performed because it is of interest to see if there is added value for using the proposed decoy discrimination method in place of the pairwise potentials method for top model selection, and not the other way around.

In later sections and the following chapter, variants of the neural networks are proposed, with additional features, and for each such variant, the Wilcoxon sign-rank test is carried out for each of the variants, as well as for each structural similarity measure.

2.3.6.2 Wilcoxon sign-rank test on Spearman correlation coefficients

For each of the 70 sets of decoys from all the decoy datasets, the Spearman rank correlation coefficient is calculated between the output scores of a particular decoy discrimination method and the structural similarity scores of the decoys in that set. This is performed for each decoy discrimination method, and each structural similarity measure (TM-score, GDT-TS, MaxSub).

Here, the one-tailed Wilcoxon sign-rank test is again used to reject/not reject the null hypothesis that the median of the distribution of the differences between the Spearman rank correlation coefficients produced by the machine learning method and the pairwise potentials method is zero.

The null and alternative hypotheses are as follows:

$$H_0 : M_{A-B} = 0$$

$$H_A : M_{A-B} > 0 \tag{2.8}$$

where M_X is the median of the distribution of random variable X, where A and B are the random variables describing the Spearman correlation coefficients of the 70 samples for the proposed machine learning decoy discrimination method and the pairwise potentials method respectively. A significance level of 5% is used.

2.3.6.3 ROC analysis

Strictly speaking, this is not a statistical test, but a classifier test. The various decoy models in all the 70 sets of decoys are dichotomized into 'true' and 'false' data. There are altogether 142625 decoy models in the 70 sets of decoys from the 8 decoy datasets.

The purpose of the ROC analysis is to investigate how well the proposed decoy discrimination methods assign lower output scores to poorer quality models, and higher output scores to higher quality models. Here, the definition of 'quality' includes the RMSD, which measures the extent of global similarity to the native structure, and TM-score, GDT-TS, and MaxSub, structural similarity measures which take into account the local similarity of a decoy model to the native structure.

In a way, the ROC analysis complements the benchmarking measures of Z score and enrichment score. While the Z score and enrichment focus on the native structure and low RMSD structures respectively, the ROC analysis investigates how well a decoy discrimination method performs across the entire range of quality of models.

One set of thresholds for the dichotomy is 6Å, 0.4, 0.25 and 0.3 for RMSD, TM-score, GDT-TS and MaxSub respectively. Another set of more stringent thresholds is chosen to investigate how the ROC curves vary for differing thresholds. The second set of thresholds for the dichotomy is 4Å, 0.5, 0.35 and 0.4 for RMSD, TM-score, GDT-TS and MaxSub respectively. Such thresholds may be somewhat arbitrary, but the purpose

of this test is to get an idea of how well the different decoy discrimination methods can assign the decoy models into the two classes. Hence there is a need to select particular thresholds for the similarity measures.

In summary, the ROC analysis allows for the assessment of how well various decoy discrimination methods assign models of high quality to the 'true positive' class for each structural similarity measure, while keeping the fraction of 'false positives' to a minimum.

2.3.7 Testing a Decoy Structure

The distance map (as shown in Figure 2.2) of a structure, be it native or decoy, is first generated. Ignoring the lower half of the symmetric distance map, each of these entries in the top half of the map is represented in vector form (*R1*, *R2*, *k*, *d*), where *R1* and *R2* are the residues, *k* is the sequence separation between the two residues and *d* is the distance apart. For each distance entry, the neural network of the particular *k* is then selected (distance entries with *k* less than 4 are left out). Each distance entry is in the vector form (*R1*, *R2*, *d*). The vector (*R1*, *R2*, *d*) is then put through the neural network and the output score is obtained for that distance entry. This output score then goes into a results matrix, as shown in Figure 2.18, which shows a typical results matrix for any given structure. The diagonals marked *'X'* and *'Y'* indicate the scores obtained from neural networks of separations k=4 and k=5 respectively.

2.3.7.1 Different ways of combining Neural Network Results

After the network output scores are assigned to each distance map entry, the scores are combined. Each structure has a corresponding combined score. It is hoped that the native structure would have the highest combined score. Scores of different separation ranges can be combined to see how well the neural networks of different sequence separation ranges differ in discriminative power.

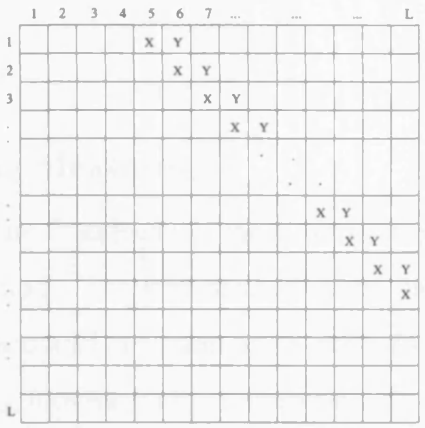


Figure 2.18: Results Matrix of a Structure

No.	Name of combination	Range of k
1	Short Range (S)	$4 \leq k \leq 10$
2	Short+Medium Range (SM)	$4 \leq k \leq 22$
3	Short+Medium+Long Range (SML)	$4 \leq k \leq 22, k > 22$

Table 2.8: Different ways of combining scores from the various neural networks

There are altogether 2^{20}-1=1048575 ways to combine 20 different types of scores. These 20 types refer to each k where k ranges from 4 to 22 inclusive, and $k > 22$. Here, 3 levels of combinations are chosen. These are shown in Table 2.8.

A short range sequence separation (S) is defined as $4 \leq k \leq 10$, a short and medium range sequence separation (SM) is defined as $4 \leq k \leq 22$, and a short, medium and long sequence separation range (SML) is defined as $k \geq 4$. For each results matrix (Figure 2.18), the scores of all entries in the distance map for each particular range of sequence separation (S, SM, SML) are averaged to produce a single score that represents the likelihood score that describes whether the structure of the results matrix is near-native or not. This is performed for all decoys, as well as the native structure.

The results of these 3 ways of sequence separation combination can then be evaluated using the Z score (Equation 2.5 on page 102) and the enrichment measure (Equation 2.6 on page 103).

2.3.8 Benchmarking Measures

In this section, 2 methods are described, for the purpose of benchmarking the proposed decoy discrimination method. The first method is the pairwise potentials of mean force [59]. The pairwise potentials of mean force method has already proven to be an effective energy function in fold recognition methods such as mGenTHREADER [36] and ab initio fragment assembly methods such as FRAGFOLD [116]. It can also serve as an energy function for the evaluation of candidate decoy structures. Section 2.3.8.1 gives details on the calculations of the pairwise potentials.

The second method is the K-Nearest Neighbours (K-NN) algorithm. In most machine learning problems, it is useful to compare the results of any proposed machine learning algorithm with that obtained from a simple K-NN classifier. In this thesis, the K-NN method is formulated in the context of the decoy discrimination problem.

2.3.8.1 Using Pairwise Potentials Of Mean Force

The pairwise potentials of mean force, used in mGenTHREADER and FRAGFOLD, can also be used for decoy discimination, as in the case of the MQAP method MODCHECK [91]. Here, it is used as a means of providing a benchmark for the proposed neural network method.

Equation 2.9 shows how the net potential of a residue pair ab, with sequence separation k and distance interval s, is calculated. The distance is taken between Cβ-Cβ atoms of the residue pair. In the case of glycine, an approximate Cβ position is calculated.

$$\Delta E_k^{ab} = RTln[1 + m_{ab}\sigma] - RTln[1 + m_{ab}\sigma\frac{f_k^{ab}(s)}{f_k(s)}] \tag{2.9}$$

The term m_{ab} is the number of pairs ab observed with sequence separation k, σ is the weight given to each observation and is set to 0.02, $f_k(s)$ is the frequency of occurrence

of all residue pairs at topological level k and separation distance s, $f_k{}^{ab}(s)$ is the equivalent frequency of occurrence of residue pair ab and RT is taken to be 0.582 kcal/mol.

Equation 2.9, shown on page 109, is derived for all 400 types of residue pairs for each separation k, where $4 \leq k \leq 22$, as well as for long range potentials, where $k > 22$.

For each decoy (and native) structure of a protein, the energy of the structure is calculated according to Equation 2.10.

$$E_{(structure)} = \sum_{k} \sum_{ab \in R} \Delta E_k^{ab} \tag{2.10}$$

where R is the set of pairwise residues in the structure which are separated by k residues in the sequence. Here, instead of calculating the pairwise potentials based on the training dataset, the in-house pairwise potentials method is used. This is done so that a stringent comparison of the proposed machine learning method can be done against the pairwise potentials method since the latter has proved competitive in the last few CASP experiments.

After the energy of each structure is calculated, the Z score of the native structure can be derived. The native structure is expected to have the lowest energy, and hence the lower the Z score the better. For purposes of effective comparison to that of the proposed neural network method, the signs of the Z scores obtained by the pairwise potentials method are inverted.

2.3.8.2 K-Nearest Neighbours Algorithm

The K-Nearest Neighbours (K-NN) algorithm is a common machine learning classifier that takes a particular test example and assigns to it the class where the majority of the K nearest training data points belong to. The K-NN method can be used in the context of the proposed decoy discrimination method.

The training data used for classifying test data is the same as that used for training of the neural networks. For each separation k, there exists 400 sets of training data with both types of labels *'1'*s and *'0'*s (for each type of residue pair) which the K-NN classifier can be applied to, depending on the particular test data point. For each decoy (and native) structure, there exists a set of test data points in the form of ($R1$, $R2$, d) for each separation k. Each of these test data points (with a particular distance d) is then used to select the K nearest neighbours in the training set of ($R1$, $R2$) where the training data points are of the class label *'1'* or *'0'*. The distance measure used to classify 'nearest' is that of the standard Euclidean distance. In the benchmarking of methods in this thesis, the K-NN method is restricted to use the distance measure only, even as more input features, such as solvent accessibilities, are added in later sections. The number of neighbours used in the benchmarking is 10 and 100. Instead of assigning an absolute *'1'* or *'0'* to the test data point, the ratio of the number of training data points with labels *'1'*s to the number of training data points with labels *'0'*s is taken.

As an example, Figure 2.19 shows a test data point marked *'X'* of say, an Alanine-Leucine pair. The number of nearest neighbours is 10 in this example. The 10 nearest neighbours in the training set of all Alanine-Leucine data obtained from native structures and simulated decoy structures are selected, and the ratio of the number of labels *'1'* (indicated by the white circles) to the number of labels *'0'* (indicated by the black circles) is calculated. In this case, the test data point is assigned the value of 0.6, since there are 6 *'1'*s and 4 *'0'*s. In this case, the Euclidean distance of the points is 1-dimensional and hence the graphical representation in Figure 2.19 is that of a straight line.

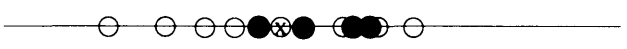


Figure 2.19: An example of the K-Nearest Neighbours Algorithm

Figure 2.20 shows a results matrix of a candidate structure, which is similar to that in Figure 2.18, obtained by applying the K-NN algorithm. The figure shows the values of each individual test data point along the k=4 and k=5 diagonal assigned by the K-NN

algorithm.

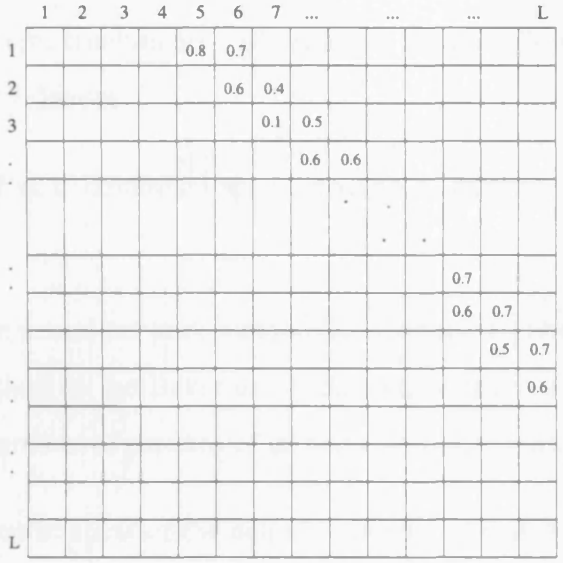


Figure 2.20: A typical K-NN results matrix

Similarly to the neural network method, the values in the K-NN results matrix can be combined in 3 ways, namely the short range (S) combinations of separation ($4 \leq k \leq 10$), the short and medium range (SM) combination ($4 \leq k \leq 22$), and the short, medium and long range (SML) combination ($k \geq 4$). This is done for the sake of comparison to that of the neural network method.

2.4 Results

In this section, the results of testing are presented and discussed. The ensemble of trained neural networks are tested on a simulated test decoy dataset (Table D.3), as well as on the Baker decoy dataset and the Decoys 'R' Us suite of decoy datasets.

This section shows the

- results of testing on the preliminary test dataset, which demonstrates the viability of the approach of using neural networks for decoy discrimination.
- results from a single k=4 neural network of the *1r69* protein of the Baker decoy dataset, which illustrates the plausibility of the neural network approach to decoy

discrimination.

- results of different combinations of sequence separations of the *1r69* protein of the Baker decoy dataset.
- results of different combinations of sequence separations on all the decoy datasets.
- Z scores of the neural network method, K-Nearest Neighbours and the pairwise potentials method on the Baker decoy dataset, which is useful since the Baker dataset has a number of proteins of different secondary structural classes.
- Z scores and enrichment of the neural network method, K-Nearest Neighbours and the pairwise potentials method on all decoy datasets.

The statistical tests are deferred to Section 2.5.5 where the results of variants of the neural network methods with additional input features, along with this neural network method with inputs of pairwise distance only, are presented and discussed.

2.4.1 Testing of Preliminary Test Dataset

Firstly, the preliminary test dataset is used to find out how well the neural networks work in discriminating native structures from 'random' decoys. Random decoys are generated by randomizing the residues along the sequence and then threading it to the structure, as described in Section 2.3.1.1. Each native structure has 50 'random' decoys, and the likelihood of each of the structures is assessed using the neural networks. The native structures are expected to come out tops.

The comparison of native structures versus random decoys are benchmarked using the sequence reversal method and the perturbed distance method (as described in Section 2.3.1.2), and the results are shown in Tables 2.9.

In Table 2.9, the results are presented in terms of the number of native structures that have been correctly identified as the one with the highest rank by the decoy discrimination method. The average Z score (Equation 2.5 on page 102) for the 95 test

Simulated Decoy Method	S Combination	SM Combination
Sequence reversal	91 (3.851)	90 (3.969)
Perturbed distance	82 (3.393)	86 (3.505)

Table 2.9: Number of native structures with the highest rank (and Z scores) among the random decoys

proteins is also given in parentheses. The sequence reversal method performs slightly better than the perturbed distance method, in terms of the number of native structures identified as the ones with the highest score among the 'random' decoys, and in terms of the average Z scores of the native structures. Both the short range (S) and short and medium range (SM) combinations of sequence separations perform better for the sequence reversal method than that of the perturbed distance method, in terms of both the Z score and the number of native structures that are ranked the highest.

Therefore, the sequence reversal method is selected for the acid test, namely the benchmarking of the decoy datasets, as well as for subsequent enhancements to the neural network method.

2.4.2 Testing of Baker Dataset

The preliminary tests may yield good results for 'random' decoys, but it remains to be seen how well this proposed decoy discrimination method performs on 'real' decoy test datasets. For each protein in the Baker decoy dataset as listed in Table 2.5, the native structure and its corresponding decoy structures are tested according to the procedure listed in Section 2.3.7.

For the Baker decoy dataset, and for other subsequent decoy datasets in Decoys 'R' Us, the neural network method is evaluated against the K-Nearest Neighbours method (K-NN) and the pairwise potentials of mean force method, using the Z score (Equation 2.5 on page 102) and the enrichment measure (Equation 2.6 on page 103). Because the Baker decoy dataset has proteins with different secondary structural classes (Table 2.5),

it is possible to see how different classes of proteins perform with the neural network, K-Nearest Neighbours and pairwise potentials methods in the Baker decoy dataset.

First, the following section investigates the k=4 network output of a particular protein, *1r69* and its set of decoys.

2.4.2.1 k=4 Neural Network Result of the *1r69* protein

Figure 2.21 shows the distribution of the mean of the scores obtained for separation k=4 (mean of the scores along the '*X*' diagonal in Figure 2.18) for all structures (native and decoys) of protein *1r69*. The native structure is marked with an arrow. Figure 2.22 shows the scatter plot of the RMSD of the decoys versus the mean of the k=4 scores for all the structures. The native structure, with an RMSD of zero, is marked with an arrow. Here it is important to point that the distribution of the mean of the k=4 scores pertains only to a subset of residue pairs in each native or decoy structure, while the RMSD distribution is over the entire structure, that is, the calculation of the RMSD is performed globally over the entire decoy structure, and not just a fragment of it.

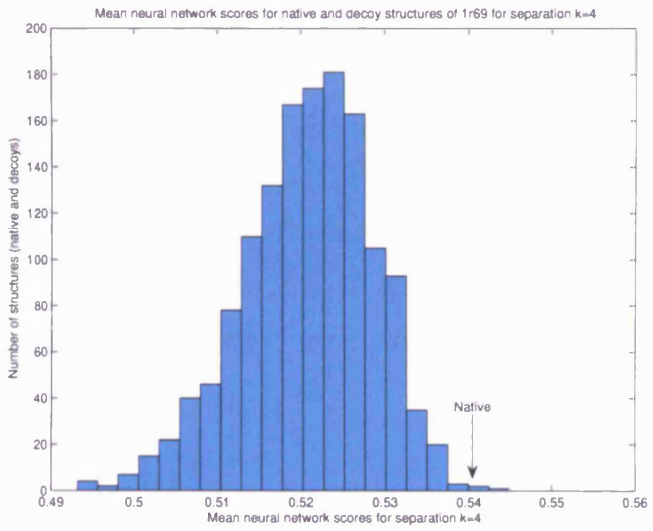


Figure 2.21: Mean neural network scores for separation k=4 for structures of protein *1r69*

Additional plots of the distributions of the mean scores of k=4 for the rest of the

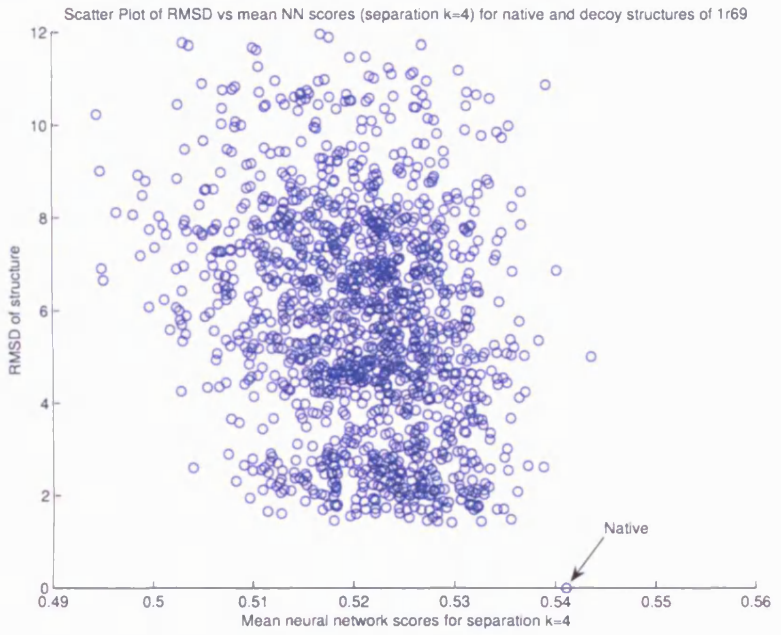


Figure 2.22: Scatter plot of RMSD vs mean NN scores for k=4 for structures of protein *1r69*

proteins in the Baker decoy dataset (Table 2.5) is found in Figures E.1 to E.4 in Appendix E. While it can be seen from Figure 2.21 that the native protein structure has a high mean score for k=4, not all of these proteins rank their native structures as well as the *1r69* protein.

Appendix F shows the distributions of the various separations from $4 \leq k \leq 22$, and $k > 22$, for protein *1r69*. For the *1r69* protein, it can be seen from Figures F.1 to F.5 that the smaller individual sequence separations ($k \leq 10$) give higher mean scores to the native protein structure.

While the mean of the k=4 scores demonstrates some discimination of the native structures, it is assumed that the mean of the scores from the different neural networks of various separations k can be combined to give better performance in terms of the discrimination of native structures. This is discussed in the next section.

2.4.2.2 Results of different combinations of various separations k

Figures 2.23, 2.24 and 2.25 show the histograms of distributions of neural network scores with the combination S ($4 \leq k \leq 10$), SM ($4 \leq k \leq 22$), and SML ($k \geq 4$) respectively.

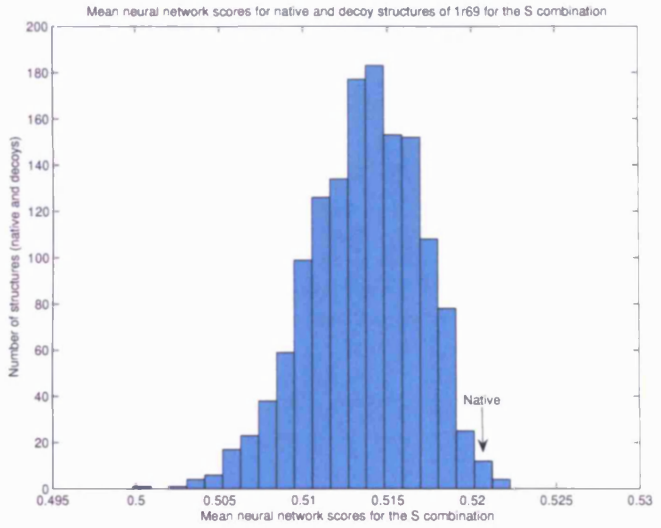


Figure 2.23: Mean neural network scores for separations $4 \leq k \leq 10$ (S combination) for structures of protein *1r69*

It can be seen from Figures 2.23 to 2.25 that the short range sequence separation, the S combination, ranks the native structure of the *1r69* protein highest among the decoys, and would most likely give a higher Z score to the native structure compared to the SM and SML combination. The SML combination appears to be the worst of the 3 combinations.

Figure 2.26 shows the average Z scores of proteins in different structural classes in the Baker decoy dataset for the 3 combinations (S, SM, SML), as well as for the single k=4 results. The first group of histograms in Figure 2.26 is obtained by averaging the Z scores for all the 22 proteins in the Baker dataset in Table 2.5. The α-only, β-only and $\alpha\beta$ classes have 9, 6 and 7 proteins respectively.

It can be seen in Figure 2.26 that the SM and SML combinations give poor Z scores across all types of structural classes, with the SML combination the worse of the two.

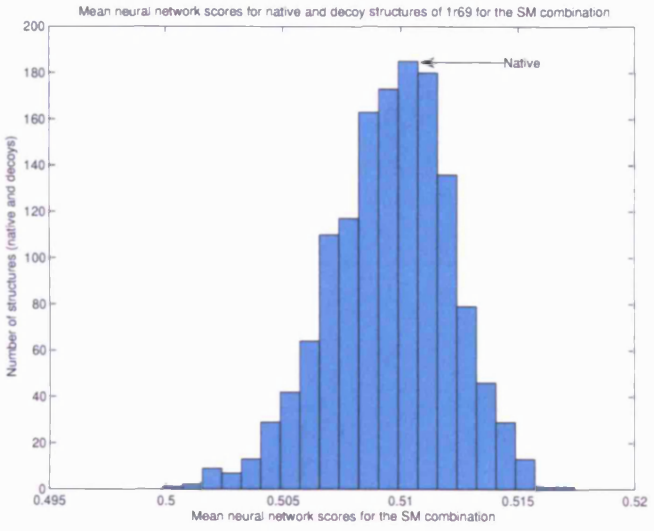


Figure 2.24: Mean neural network scores for separations $4 \leq k \leq 22$ (SM combination) for structures of protein *1r69*

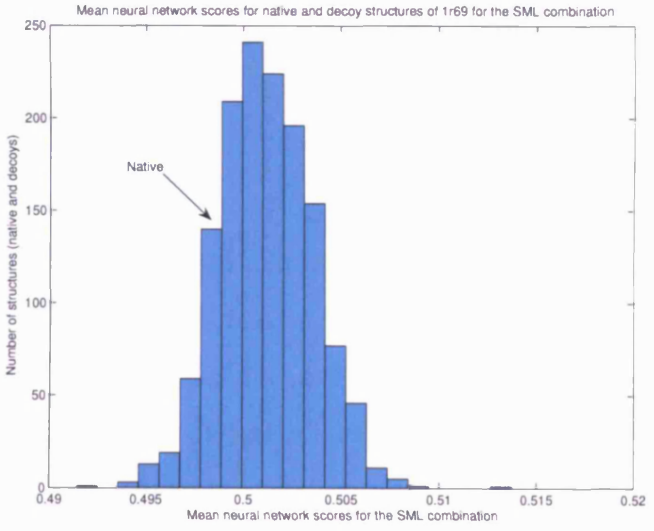


Figure 2.25: Mean neural network scores for separations $k \geq 4$ (SML combination) for structures of protein *1r69*

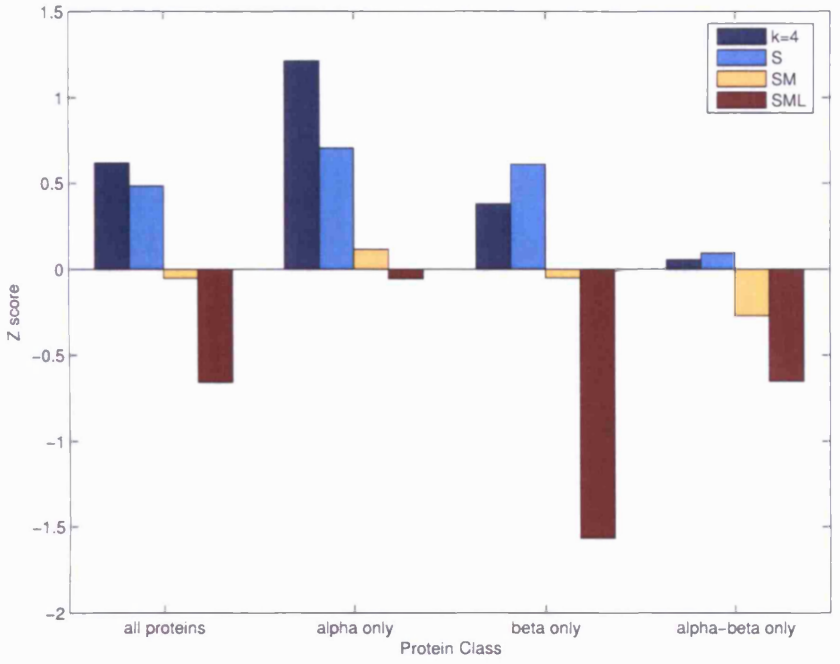


Figure 2.26: Z scores produced by the k=4, S, SM and SML combinations of sequence separations of the proposed neural network method on the different secondary structural classes of the Baker decoy dataset

The S combination performs comparatively to that of the single k=4 mean network result; it is slightly better for β-only proteins, while worse off for α-only proteins. As shown in the first group of histograms in Figure 2.26, the average Z score of the S combination of network results ($4 \leq k \leq 10$) is slightly lower than that of the single k=4 network results.

Figure 2.27 is essentially an extension of Figure 2.26. It shows the Z scores for the Baker dataset and the decoy datasets in Decoys 'R' Us suite, as well as all the combined decoy datasets, averaged across all the proteins in each dataset. The number of native proteins in each decoy dataset is indicated by the bracketed number.

In Figure 2.27, for the combined datasets, the S combination yields the highest Z score, followed by the k=4, SM and SML combination. Considering each decoy dataset, the S combination gives higher Z scores than SM and SML combinations for all but one decoy datasets, the odd one out being the semfold dataset where the SM combination is higher than that of the S combination. Apart from the fisa, fisa_casp3

and Baker datasets, the S combination performs better, in terms of Z score, than that of the single k=4 network.

Therefore, it seems that the S combination appears to yield the best results for the various decoy datasets in terms of Z score. In the next section, the results of the short range (S) combination of the proposed neural network decoy discrimination method are benchmarked against the pairwise potentials method, as well as the K-Nearest Neighbours (K-NN) method.

2.4.3 Comparison of NN scores with other benchmarked methods

In this section, the short range (S) combination ($4 \leq k \leq 10$) of neural network scores for the Baker decoy dataset is compared to the short range combinations of the pairwise potentials method and the K nearest neighbours method (K-NN). The name given to this neural network method is **NN-dist**, where 'dist' stands for distance-only information. Two values of K for the K-NN method are used, namely 10 and 100, as mentioned in Section 2.3.8.2. The Z scores of pairwise potentials have the magnitude signs inverted for effective comparison as mentioned in Section 2.3.8.1; hence the lowest energy structure produced by the pairwise potentials method would have the highest Z score. Similarly to Figure 2.26, Figure 2.28 shows the comparison across proteins of different structural classes in the Baker dataset for these different methods.

From Figure 2.28, the NN-dist method has a Z score, averaged over all proteins, lower than that of the pairwise potentials and K-NN (K=10) method. The pairwise potentials method does extremely well for all classes of proteins, and is easily the best method. The K-NN method, with K=10, outperforms the NN method slightly, while its K=100 counterpart has a similar performance to the NN-dist method for all proteins.

Figure 2.29 shows the Z scores for all the decoy datasets, including a combination of all datasets. It can be seen that the NN-dist method is comparable to the K-NN methods (K=10, 100) for the combined datasets of 70 decoy sets, while the pairwise potentials method does best. Looking at the average Z scores of the individual decoy

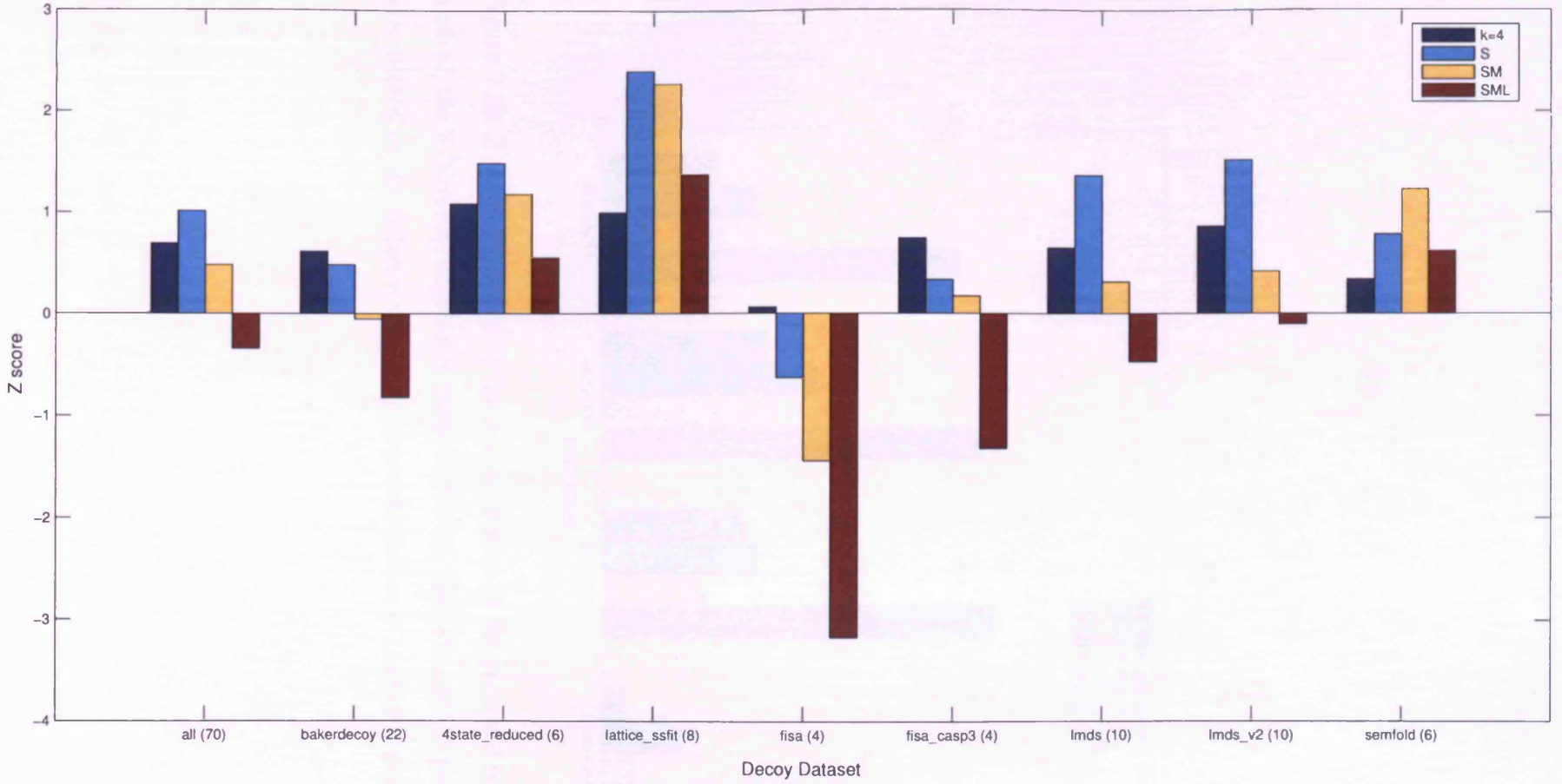


Figure 2.27: Z scores produced by the k=4, S, SM and SML combinations of sequence separations of the proposed neural network method on the different individual decoy datasets, including the combination of all the individual datasets

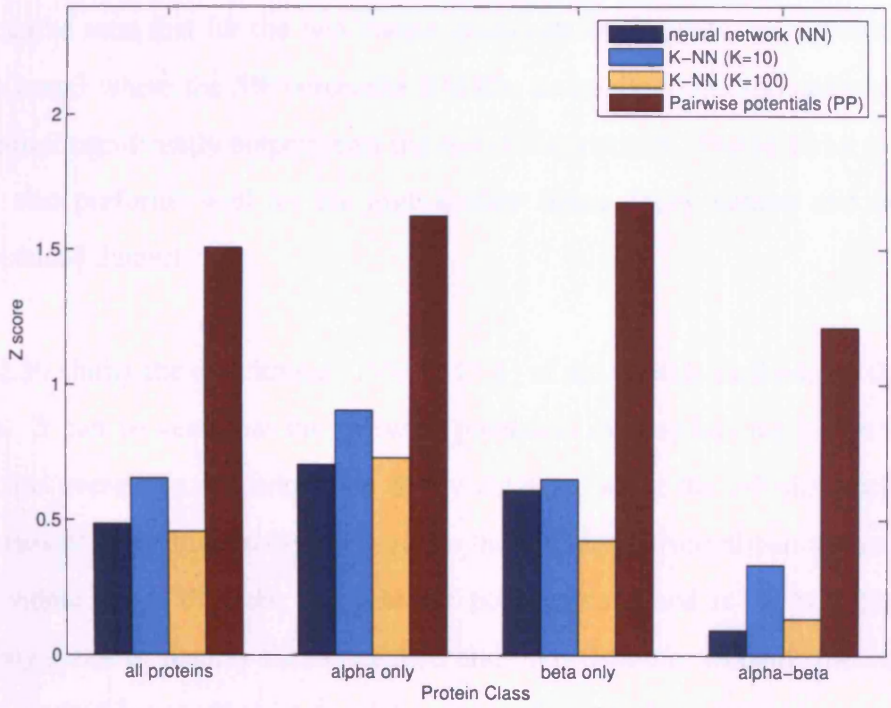


Figure 2.28: Z scores produced by the S combination of the proposed neural network method, the K-Nearest Neighbours methods (K=10, K=100), and the pairwise potentials method on the different secondary structural classes of the Baker decoy dataset

datasets, the NN-dist method performs better than the K-NN method (K=10) for 4 out of the 8 decoy datasets, namely the 4state_reduced, lattice_ssfit, lmds and lmds_v2 datasets, and it does better than the K-NN method (K=100) for all but 3 decoy datasets, which are the 4state_reduced, lattice_ssfit and semfold datasets. The pairwise potentials method has the highest Z scores for the decoy datasets, with the NN-dist method having a comparable Z score to the pairwise potentials method in the lmds_v2 dataset.

Relating the Z scores in Figure 2.29 to the qualities of the decoy datasets in Table 2.6, it can be seen that for the two lowest quality decoy datasets, namely lattice_ssfit and fisa_casp3 where the 5% percentile RMSDs lie at about 7Å, the pairwise potentials method significantly outperforms the rest of the methods. The pairwise potentials method also performs well for the high quality Baker decoy dataset and the small 4state_reduced dataset.

Figure 2.30 shows the enrichment (15% $\times$ 15%) of the various methods on the decoy datasets. It can be seen that the pairwise potentials method has the highest enrichment score overall for the combined decoy datasets, while the NN-dist method has an enrichment score marginally higher than the K-Nearest Neighbours methods. For the individual decoy datasets, the pairwise potentials method is the best for all but two decoy datasets, namely the lattice_ssfit and lmds datasets. In these two cases, the NN-dist method has the highest enrichment score by a small margin.

In general, the NN-dist network method is comparable to the K nearest neighbours method (K=10, K=100) in discriminating the native structure from the decoys (Z score) and associating structures with high scores to low RMSD structures (enrichment), while the tried and tested pairwise potentials method outperforms the proposed NN-dist method in terms of overall Z score and enrichment factor.

In a bid to improve the decoy disrimination process, the next section investigates the inclusion of additional input features, in the form of solvent accessibility values of the residue pairs.

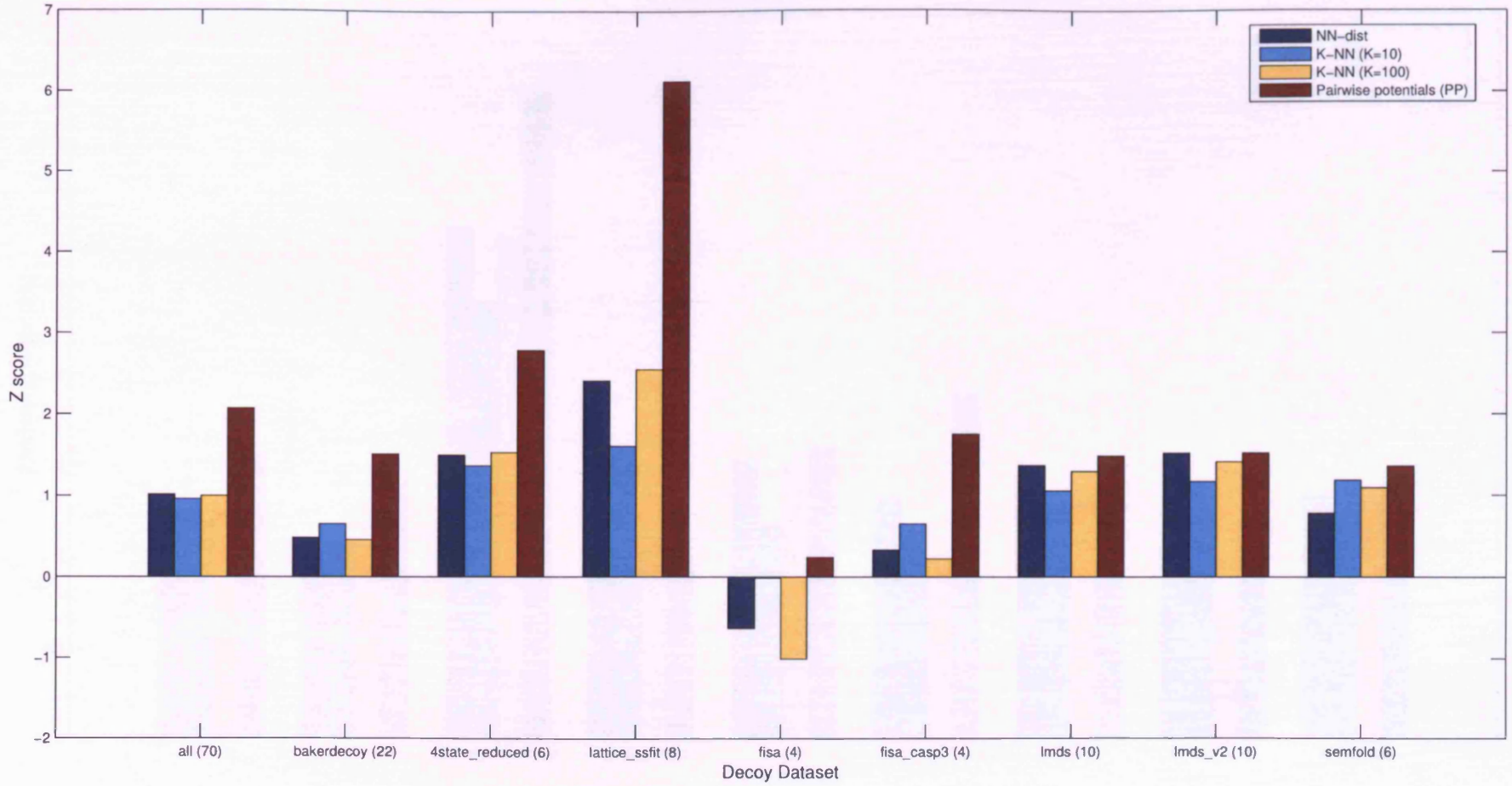


Figure 2.29: Z scores produced by the S combination of the NN-dist method, the K-Nearest Neighbours methods (K=10, K=100) and the pairwise potentials method on the different individual decoy datasets, including the combination of all the individual datasets

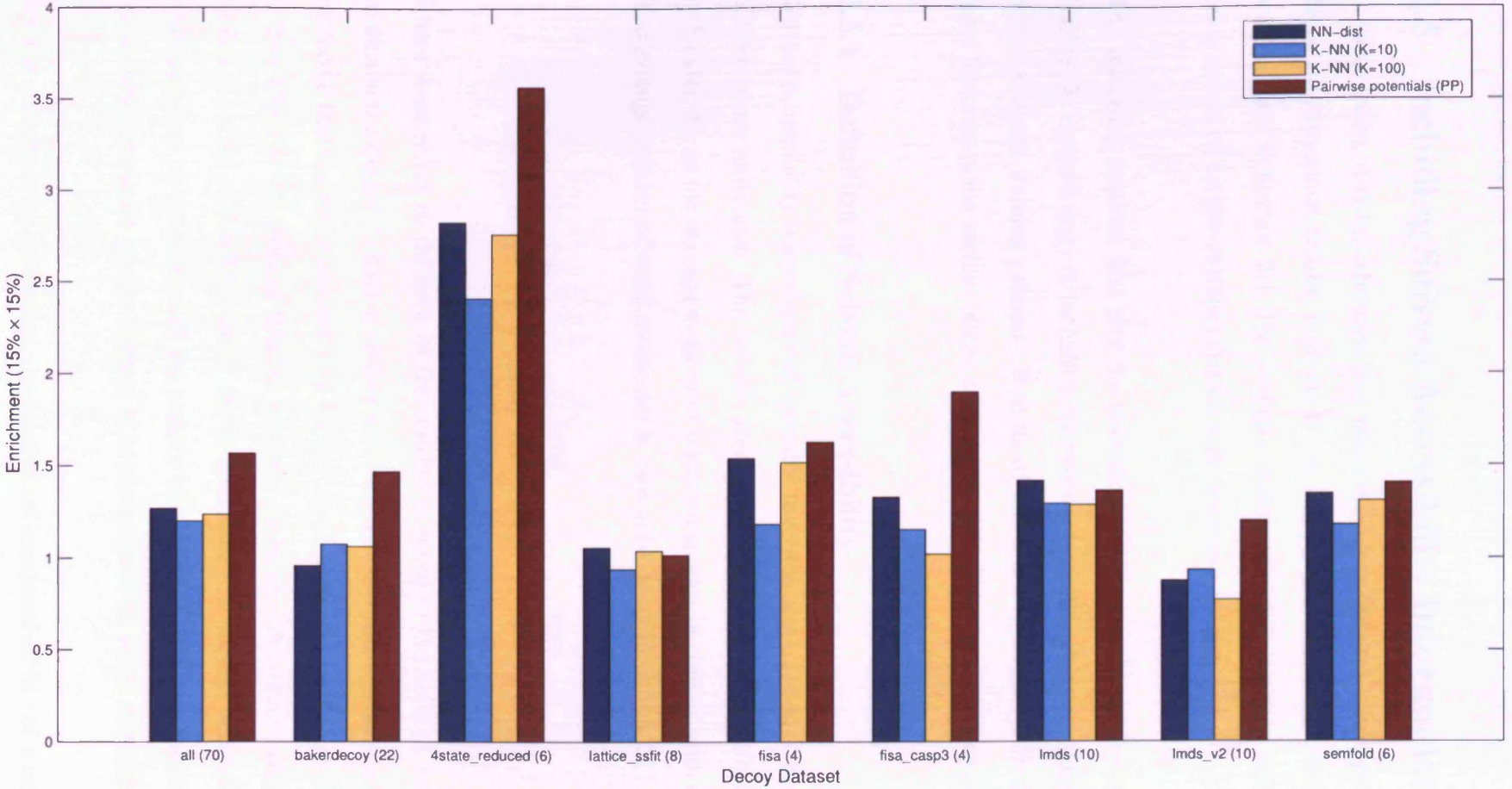


Figure 2.30: Enrichment scores (15% × 15%) produced by the S combination of the NN-dist method, the K-Nearest Neighbours methods (K=10, K=100) and the pairwise potentials method on the different individual decoy datasets, including the combination of all the individual datasets

2.5 Including Solvent Accessibility Information

In this section, it is hypothesized that additional input features can improve previous decoy discrimination results, in terms of the Z score and enrichment measure, which are presented in Section 2.4. The proposed additional features are the solvent accessibility values of the two residues that form the pairwise distance.

The following sections first give the definition of solvent accessibility, and then describes the methodology of including the proposed additional input information in the neural network training process. Results of the neural networks with the enhanced input features on the various decoy datasets are then presented and discussed.

2.5.1 Definition of Solvent Accessibility

Solvent accessibility is a property of a residue that indicates its level of exposure to the solvent water molecules. The solvent accessibility of a residue is defined, according to Sander [8], as the average number of water molecules in contact with each residue. The average number of water molecules is estimated using Equation 2.11 in [8].

$$W = \frac{Area}{V(water\ molecule)^{\frac{2}{3}}} \approx \frac{Area}{10} \tag{2.11}$$

where *Area* refers to the area of the residue exposed to the solvent, and one water molecule is assumed to have a volume of 30Å^3. The estimated surface area of 1 water molecule that can be in contact with the residue is thus $30^{\frac{2}{3}} \simeq 9.65 \approx 10$. The ratio is then taken as the average number of water molecules in contact with the residue. Residues buried within the core of the protein have low solvent accessibility values, while residues on the surface of the protein have high solvent accessibility values because they experience greater degrees of exposure with the water molecules.

For the additional input features, the relative solvent accessibility of a residue is used instead of its absolute value. Absolute values of solvent accessibility of residues in a

Residue	Maximum Solv Acc	Residue	Maximum Solv Acc
ALA (A)	106	MET (M)	188
ASX (B)	160	ASN (N)	157
CYS (C)	135	PRO (P)	136
ASP (D)	163	GLN (Q)	198
GLU (E)	194	ARG (R)	248
PHE (F)	197	SER (S)	130
GLY (G)	84	THR (T)	142
HIS (H)	184	VAL (V)	142
ILE (I)	169	TRP (W)	227
LYS (K)	205	TYR (Y)	222
LEU (L)	164	GLX (Z)	196

Table 2.10: Maximum solvent accessibility values of the 20 residue types

structure are calculated using the DSSP program [8]. The absolute solvent accessibility of each residue is then normalized by dividing by the maximum possible value of solvent accessibility for that residue type (Equation 2.12). The maximum possible solvent accessibility of each residue type X is defined as the maximum exposure surface area of residue X in an extended tripeptide Gly-X-Gly [138]. The 20 residues have different sizes, and hence different maximum exposable surface areas to the solvent. Table 2.10 shows the maximum values of the absolute solvent accessibility of each residue type [139]. The relative solvent accessibility value of a residue would have the value between 0 and 1 inclusive.

$$relative\ solvent\ accessibility = \frac{absolute\ solvent\ accessibility}{maximum\ solvent\ accessibility} \tag{2.12}$$

In Table 2.10, the maximum solvent accessibility values of residue types ASX and GLX are given for the sake of completeness. It can be seen that these values are the average of ASN and ASP, and GLU and GLN respectively. In practice, the residue identities ASX and GLX are never encountered in the structures of the training data

(Table D.1) and decoy test datasets.

2.5.2 Incorporating Additional Inputs in Neural Networks

The additional input information extends the machine learning framework, as previously described in Section 2.2.2. The paradigm of using the neural network output as a likelihood score for assessing the native-like property of a particular structure, and the subsequent ways of combinations of these scores in the results matrix of each decoy (and native) structure, are also used here. The paradigm of using 20 different neural networks for each sequence separation k for $4 \leq k \leq 22$, and one for separations $k >$ 22 is retained.

Figure 2.31 shows the enhanced neural network topology, with the inclusion of relative solvent accessibility information, in addition to the pairwise distance and the residue identities. This new paradigm of using relative solvent accessibility information is referred to as the **NN-solvpairndist** method. It is interesting to see how the neural networks perform without the distance information, and hence the **NN-solvpair** method in Figure 2.32, which only uses the residue identities and the relative solvent accessibility information, is also included for purposes of benchmarking.

The sequence reversal method, as described in Section 2.3.1.2, is used to derive the simulated decoy training dataset. When the residues in the sequence swap positions in the 3D structure, the 3D coordinates of the atoms remain unchanged. Hence the absolute solvent accessibilities of each residue position remains constant, while the relative solvent accessibilities of the residues in the reversed sequence that occupy new positions have changed.

Figure 2.33 shows, in the context of the NN-solvpairndist method, the distribution of the positive training input vectors (representing the native structures) and negative training input vectors (representing the simulated decoy structures) for an Alanine-Alanine residue pair at separation k=4.

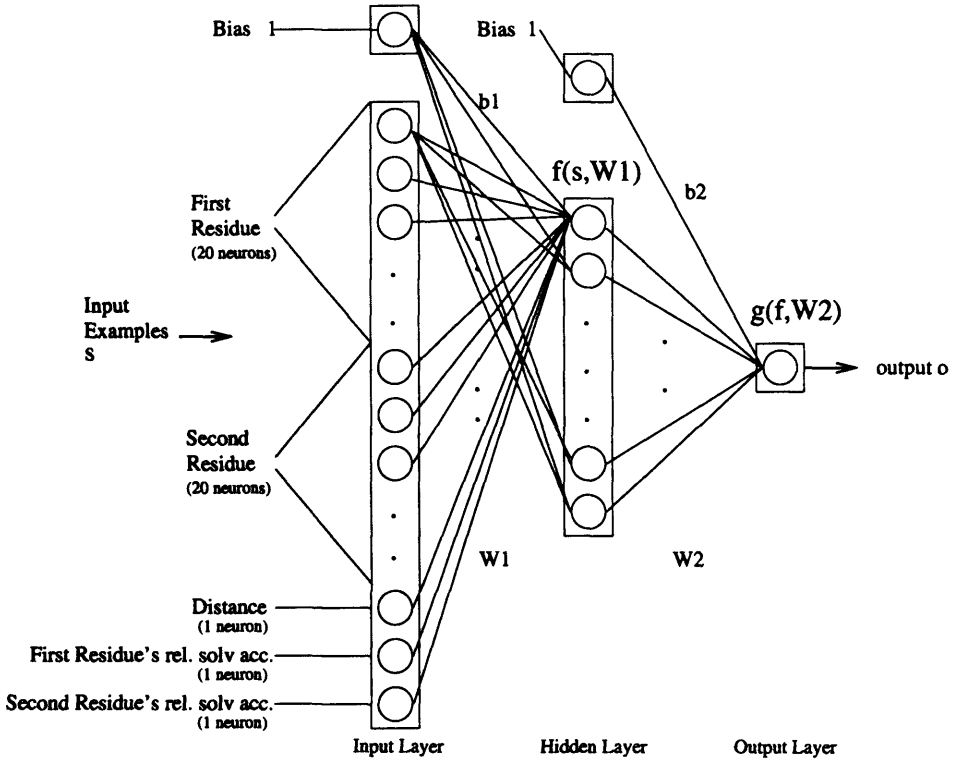


Figure 2.31: Enhanced Neural Network Topology, with relative solvent accessibility information and distance (NN-solvpairndist method)

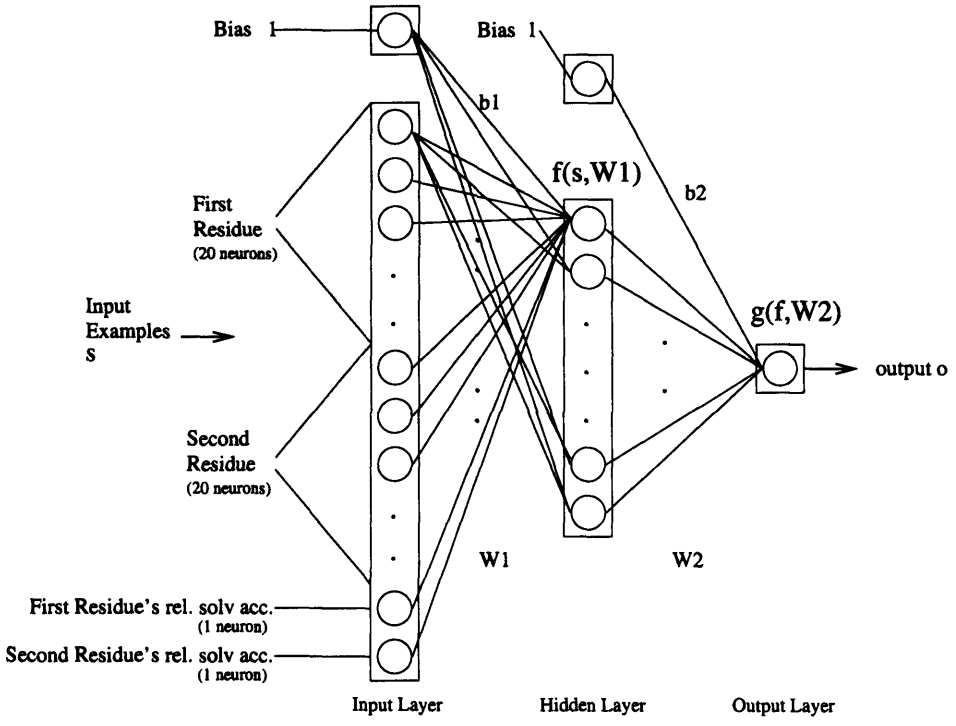


Figure 2.32: Enhanced Neural Network Topology, with relative solvent accessibility information only (NN-solvpair method)

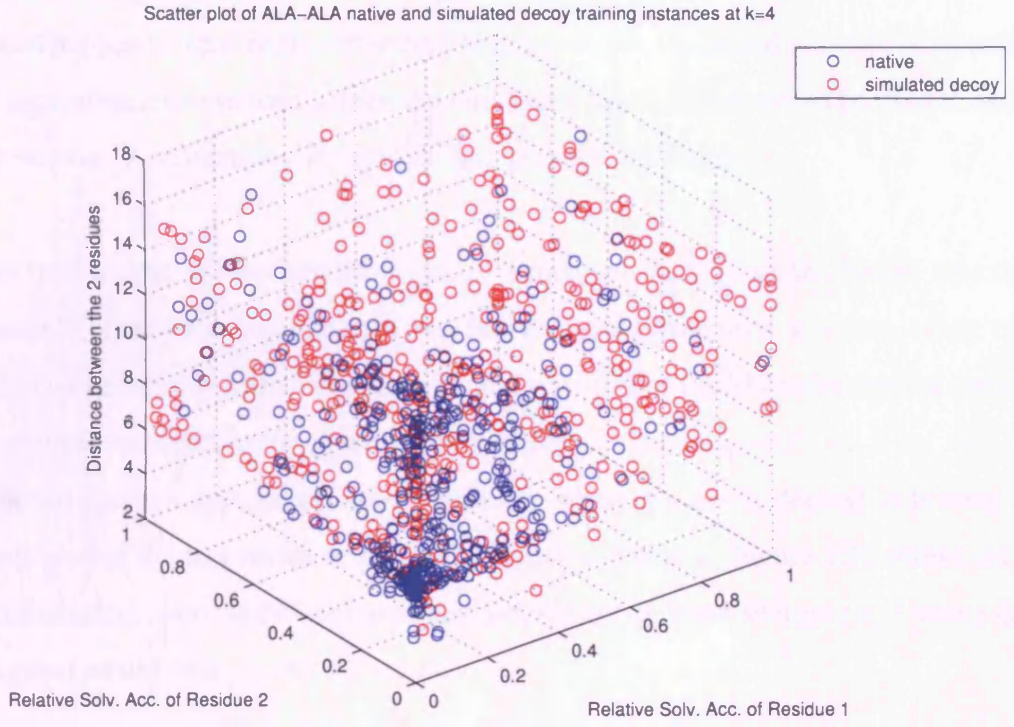


Figure 2.33: Distribution of input training instances, with additional solvent accessibility information, of ALA-ALA at k=4

It can be seen from Figure 2.33 that in the k=4 distribution of input vectors (d, $rs1$, $rs2$) of Alanine-Alanine residues, where d is the pairwise distance and $rs1$, $rs2$ are the relative solvent accessibilities of the residues, a larger concentration of the native instances are near the relative solvent accessibility value of 0 (due to the hydrophobic nature of Alanine), while the simulated decoy instances are more widely scattered across the entire relative solvent accessibility scales. Appendix G shows additional distributions of several types of residue pairs at separation k=6.

Each neural network of a particular separation k is therefore responsible for minimizing the error of the training data, across the likes of Figure 2.33 for all 400 possible residue pairs.

Positive instances representing the native training data are labelled *'1'*s, and negative instances representing the simulated decoy training data are labelled *'0'*s during

the training of the neural network. Because several positive and negative instances of the training data are close in 3D space, the function that the neural network learns would not have a zero error. After the training process, the neural network of a particular separation k would have achieved a non-linear function that would be used to assign test vectors. A validation dataset is used to prevent overfitting.

The test vectors are derived from decoy structures, as well as the native structure. Figure 2.34 shows an input feature map that is derived from each structure, where each diagonal pertains to a particular separation $k = |j - i|$. For the NN-solvpairndist method, the set of input vectors $\{\boldsymbol{a}_{ij}\}$ along each diagonal, where $\boldsymbol{a}_{ij}$ = (*R1 R2 d rs1 rs2*), are then fed into the appropriate neural network, where a score is derived indicating the likelihood of the test vector being part of a native structure. Figure 2.35 shows such a results matrix. For the NN-solvpair method, the set of input vectors $\{\boldsymbol{a}_{ij}\}$ along each diagonal would be $\boldsymbol{a}_{ij}$ = (*R1 R2 rs1 rs2*).

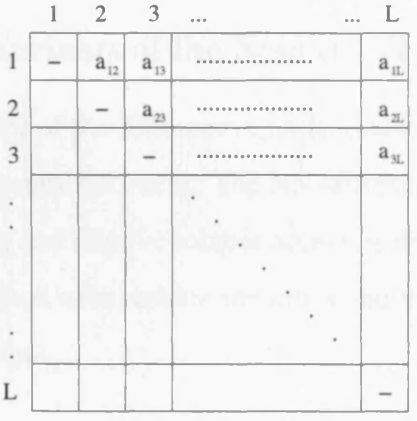


Figure 2.34: Input vector feature map

The scores in the results matrix in Figure 2.35 for each structure are then combined in three possible ways, namely the short range (S) combination ($4 \leq k \leq 10$), the short and medium range (SM) combination ($4 \leq k \leq 22$), and the short, medium and long range (SML) combination ($k \geq 4$). These combined scores are then compared with the results in Section 2.4.

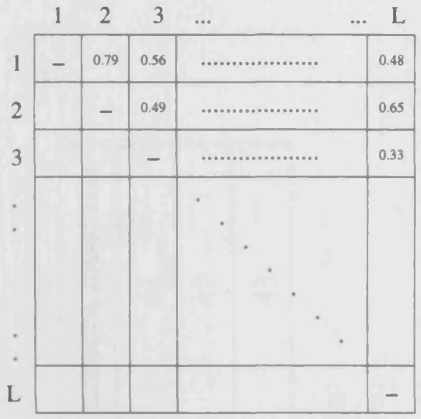


Figure 2.35: Results matrix of each structure

The next section shows a summary of the variants of the proposed neural network discrimination methods.

2.5.3 Summary of Variants of the Neural Network Method

Table 2.11 shows a summary of the different input features of the proposed decoy discrimination method using neural networks. The NN-solvpairndist method uses residue identities, pairwise distance and relative solvent accessibilities of the residues as input, while the NN-solvpair method uses residue identities and the relative solvent accessibilities of the residues as input.

2.5.4 Materials and Methods

This section describes the training and testing methodology that are used with the additional input features. Most of the details in the methodology are similar to that in Section 2.3.

Name	Input Features	No of networks	Network input size
NN-dist	Residue pair identities and Distance	20	41
NN-solvpair	Residue pair identities, and Relative Solvent Accessibilities	20	42
NN-solvpairndist	Residue pair identities, Distance and Relative Solvent Accessibilities	20	43

Table 2.11: A Summary of the Training Paradigms Used for Decoy Discrimination

2.5.4.1 Training and Validation Datasets

The training and validation datasets are the same, as in Tables D.1 and D.2. Negative instances of training data for the simulated decoys are generated using the sequence reversal method, as described in Section 2.3.1.2. The perturbed distance method is not used in this case because there are no structures available for the derivation of solvent accessibilities. The training data would naturally have two extra values, namely the relative solvent accessibilities of the residues, as shown in Table 2.12.

The absolute solvent accessibility values of the training data are obtained from the DSSP program [8] and normalized using Equation 2.12 on page 127. Those residues to which DSSP could not assign any solvent accessibility values, possibly due to incomplete atom information, are discarded from the training data. Solvent accessibility values of the residues in the simulated decoy structures with reversed sequences are also obtained using DSSP, and normalized accordingly. The relative solvent accessibility values in the reversed sequence are capped at the maximum value of 1, for those residues with calculated values > 1. This could occur after the sequence reversal process, when small residues like glycine assume residual positions of high absolute solvent accessibility previously occupied by a large surface residue.

The preliminary test dataset is not used for the testing of this enhanced neural network method because it was primarily for the purpose of testing the viability of the two simulated decoy methods, namely the sequence reversal method and the perturbed distance method. Section 2.4.1 has already shown that the sequence reversal method is better than the perturbed distance method in terms of recognizing native structures from random structures. Hence this test is not repeated here.

2.5.4.2 Neural Network Training Issues

This section describes the neural network training issues of the NN-solvpairndist and NN-solvpair methods. Figure 2.31 and 2.32 show the neural network topologies of the NN-solvpairndist and NN-solvpair methods respectively. Both figures are similar to

Protein	Type	Residue1	Residue2	Separation	Distance	Relative Solv. Acc. of Residue1	Relative Solv. Acc. of Residue2	Output Label
1a32	Native	ALA	SER	4	4.765	0.131	0.566	1
1a32	Native	TRP	GLY	4	6.367	0.988	0.591	1
1a32	Native	THR	TYR	4	8.894	0.724	0.145	1
...	...	...	...	...	...	...	...	..
1a32	Decoy	PHE	TYR	4	7.894	0.655	0.197	0
1a32	Decoy	LEU	ILE	4	9.664	0.677	0.016	0
1a32	Decoy	MET	LEU	4	10.032	0.840	0.309	0
...	...	...	...	...	...	...	...	..

Table 2.12: Example of k=4 training input instances (with relative solvent accessibilities) and their output labels

that of Figure 2.14, except for the input neurons that represent the solvent accessibility values. Table 2.12 shows an example of the training input instances that are presented to the NN-solvpairndist method.

The error function used in both NN-solvpairndist and NN-solvpair methods is the same as that in Equation 2.4 shown on page 98. The transfer function used for the hidden layer is the radial basis function, and the Levenberg-Marquardt training algorithm is again used to minimize the error function. The validation dataset in Table D.2 is used for early stopping.

2.5.4.3 Decoy Datasets and Test Measures

The test decoy datasets are the same ones that are used in Chapter 2. These are the Baker decoy dataset and the Decoys 'R' Us suite in Tables 2.5 and 2.3 respectively. Similarly, in order to obtain the solvent accessibility values of the residues, all the decoy and native structures of each protein in each dataset are put through the DSSP [8] program.

Benchmarking measures used to quantify the effectiveness of the decoy discrimination method with additional input features are, as previously used, the Z score and the enrichment measure. The results of the newly proposed NN-solvpairndist and NN-solvpair methods would be compared to that of the NN-dist method, as well as to that of the pairwise potentials and K-Nearest Neighbours methods.

The statistical tests mentioned in Section 2.3.6 are applied to the various methods. These tests include the top model selection using the Wilcoxon sign-rank tests, the ranking of Spearman correlation coefficients using the Wilcoxon sign-rank tests, and the ROC analysis. The results of these tests would be presented in the following Section 2.5.5.

2.5.5 Results

In this section, the results of both the NN-solvpairndist and NN-solvpair methods are presented, with comparisons to those of the NN-dist method.

This section shows the

- results of different combinations of sequence separations on all the decoy datasets, for the NN-solvpairndist and NN-solvpair methods. This is similar to the results of the NN-dist method which is presented in Section 2.4.2.2.
- Z scores of the NN-solvpairndist and NN-solvpair methods, K-Nearest Neighbours and the pairwise potentials method on the Baker decoy dataset, which is useful since the Baker dataset has a number of proteins of different secondary structural classes.
- Z scores and enrichment of the NN-dist, NN-solvpair and NN-solvpairndist methods, K-Nearest Neighbours and the pairwise potentials method on all decoy datasets.
- results of the Wilcoxon sign-rank tests for the top model selection
- results of the Wilcoxon sign-rank tests for the Spearman correlation coefficients
- results of ROC analysis

2.5.5.1 Comparison of Results Using Different Combinations

In this section, the results of the different combinations of the NN-solvpairndist and NN-solvpair methods are presented. Figures 2.36 and 2.37 show the Z scores of the different ways of combining the results of the different neural networks of separation k on the Baker decoy dataset for the NN-solvpairndist and NN-solvpair methods respectively.

There are two interesting observations that can be noted of the NN-solvpairndist method on the Baker decoy dataset from Figure 2.36. Firstly, the Z scores of k=4, S, SM and SML combinations of the NN-solvpairndist method on the various structural classes of proteins from the Baker decoy dataset are all positive, in contrast to

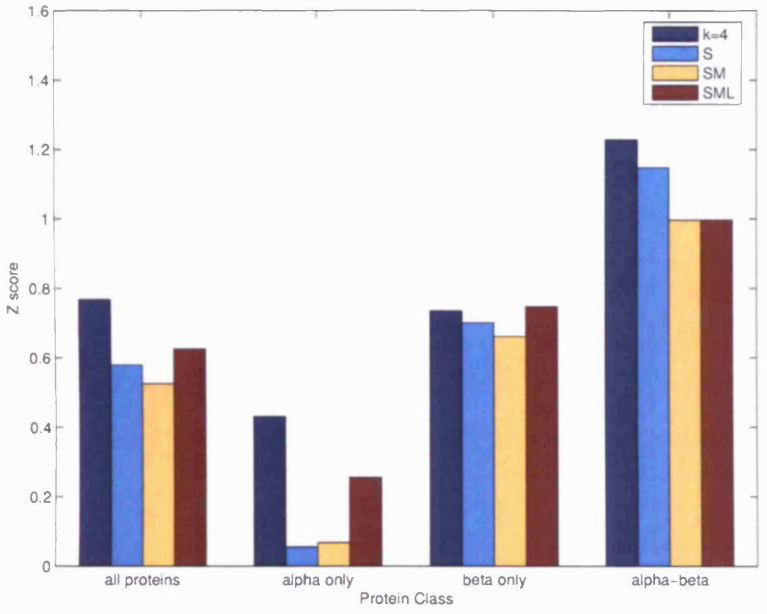


Figure 2.36: Z scores produced by the k=4, S, SM and SML combinations of sequence separations of the NN-solvpairndist method on the different secondary structural classes of the Baker decoy dataset

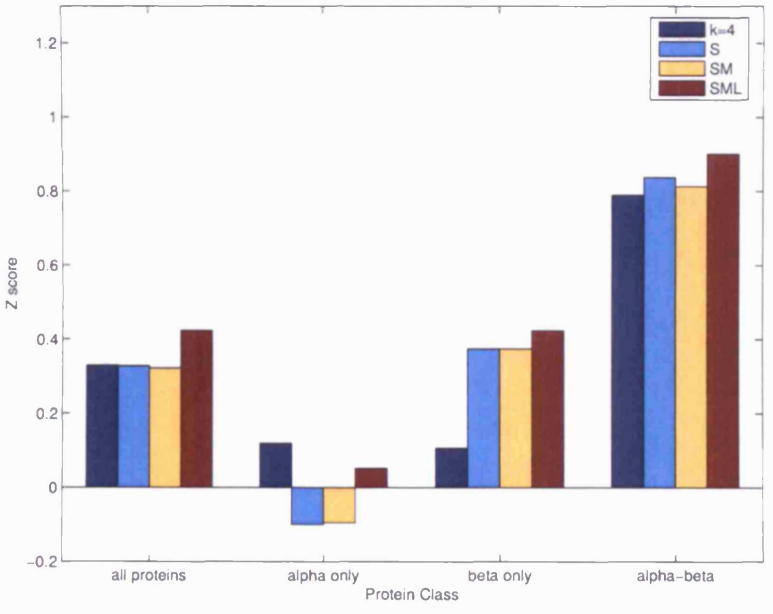


Figure 2.37: Z scores produced by the k=4, S, SM and SML combinations of sequence separations of the NN-solvpair method on the different secondary structural classes of the Baker decoy dataset

the negative SM, SML Z scores from Figure 2.26. The same can almost be said of the NN-solvpair method in Figure 2.37, except that the S and SM combinations for the α-only protein class yield negative Z scores.

While the SM and SML combinations of neural network outputs perform poorly for the NN-dist method in Figure 2.26, the SM and SML combinations are actually comparable to that of k=4 and the S combination in both the NN-solvpairndist and NN-solvpair methods. Further evidence of this can be seen in other decoy datasets in Figure 2.42, where the tests are performed on the Decoys 'R' Us suite of decoys.

The second observation is that the k=4 neural network score performs best across the entire set of proteins. It especially does best in the α-only class, which is not surprising because the helical information of a protein can mostly be captured in the information belonging to pairwise residues of sequence separation k=4. However, the other combination of scores do perform better in the rest of the decoy datasets for the NN-solvpairndist and NN-solvpair methods, as shown in Figures 2.42 and 2.43.

Figures 2.38 to 2.41 show a comparison of the three methods, NN-dist, NN-solvpairndist and NN-solvpair, over the S, SM and SML ways of network score combinations for the different classes of proteins in the Baker decoy dataset. Figures 2.38 to 2.41 are essentially graphical rearrangements of the Z scores for the 3 NN methods shown in Figure 2.26, Figure 2.36 and Figure 2.37.

It can be seen from Figure 2.38 that for all proteins in the Baker decoy dataset, the NN-solvpairndist method performs best, and the performance is rather consistent over all types of combinations of network scores. The NN-solvpairndist method is also the best method of the 3 NN methods for the β-only proteins and the $\alpha\beta$ proteins in the Baker decoy dataset as shown in Figures 2.40 and 2.41 respectively. This suggests that the additional input features of solvent accessibilities contribute positively to decoy discrimination.

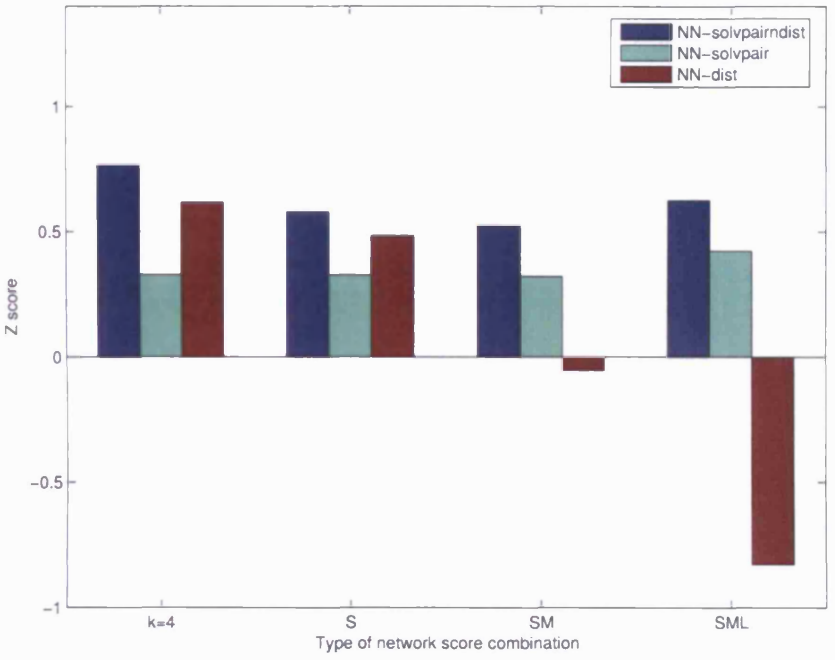


Figure 2.38: Z scores produced by the NN-solvpairndist, NN-solvpair and NN-dist methods on all the proteins in the Baker decoy dataset across the different k=4, S, SM and SML combinations

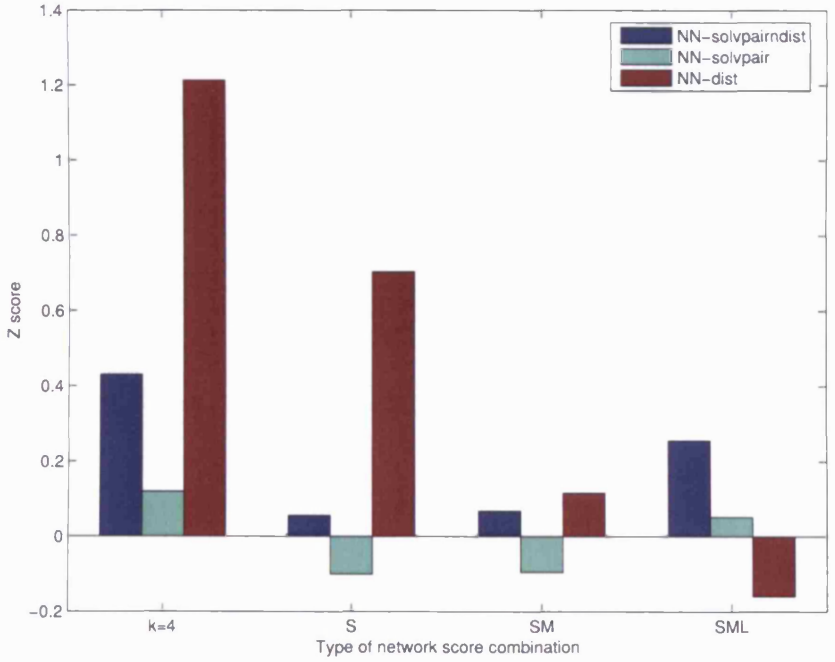


Figure 2.39: Z scores produced by the NN-solvpairndist, NN-solvpair and NN-dist methods on α-only proteins in the Baker decoy dataset across the different k=4, S, SM and SML combinations

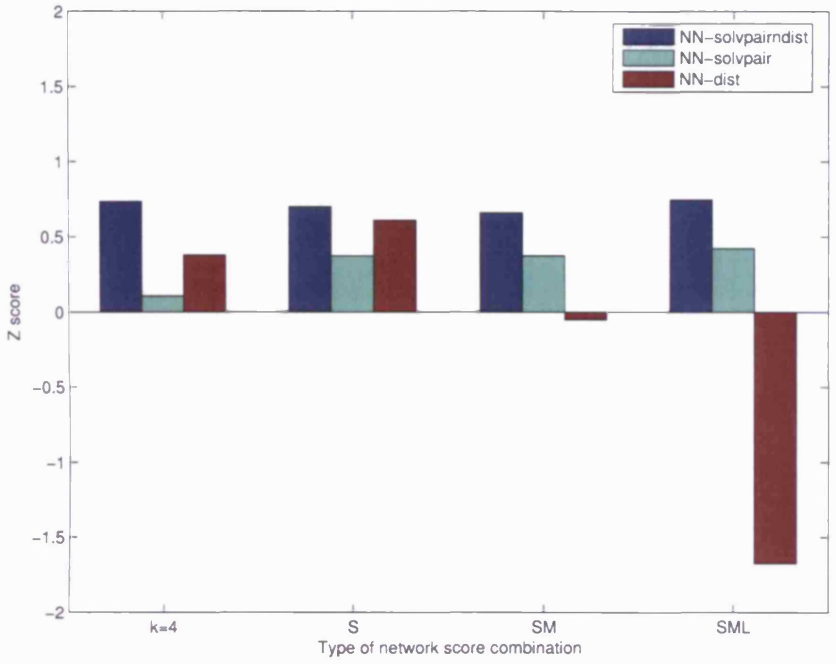


Figure 2.40: Z scores produced by the NN-solvpairndist, NN-solvpair and NN-dist methods on β-only proteins in the Baker decoy dataset across the different k=4, S, SM and SML combinations

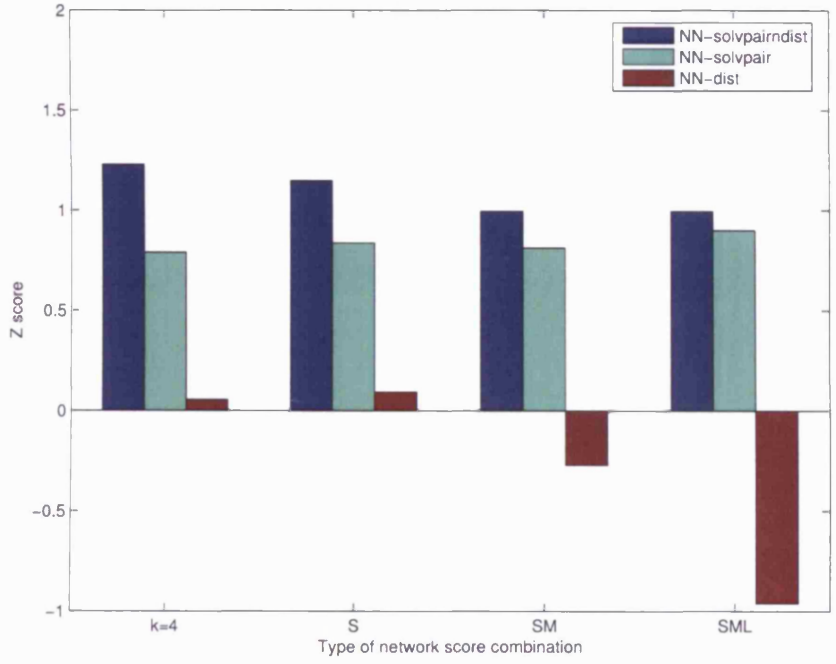


Figure 2.41: Z scores produced by the NN-solvpairndist, NN-solvpair and NN-dist methods on $\alpha\beta$ proteins in the Baker decoy dataset across the different k=4, S, SM and SML combinations

It is interesting to note from Figure 2.39 that for α-only proteins in the Baker decoy dataset, the NN-dist method, with the k=4 and S combination, is the highest. This suggests that the short-range distance information alone may be the most discriminative for α-only proteins.

Figure 2.42 extends the comparison of different ways of combination to the Decoys 'R' Us suite of decoys for the NN-solvpairndist method. It can be seen from Figure 2.42 that for the combined datasets, there is little difference between the k=4, S, SM and SML combinations of sequence separations in terms of the Z score.

Preliminary comparisons to the NN-dist plot in Figure 2.27 suggest that the NN-solvpairndist method yields better Z scores for all the combinations of sequence separations, including the single k=4, S, SM and SML combination. Section 2.5.5.2 would present detailed graphical plots of the comparison between all methods, including the pairwise potentials and the K-Nearest Neighbours methods. Like the earlier observations on the Baker decoy dataset, it is observed that the SM and SML combinations are comparable to that of the S combination. The k=4 single mean score does not do as well as the other combinations for the fisa, lmds and lmds_v2 decoy datasets.

Figure 2.43 shows the results of the Z scores of the NN-solvpair method on the various decoy datasets. Similarly, it can be observed that the different network score combinations are comparable in performance for the various decoy datasets, including the combined dataset of 70 decoy sets.

Since the Z scores of the k=4, S, SM and SML combinations on the Baker decoy dataset are comparable with one another for both the NN-solvpairndist and NN-solvpair methods, the S combination is chosen for further benchmarking purposes in the next section. This is to facilitate an effective comparison with the results of the NN-dist methods, which is most effective when using the S combination, as discussed in Section 2.4.2.2.

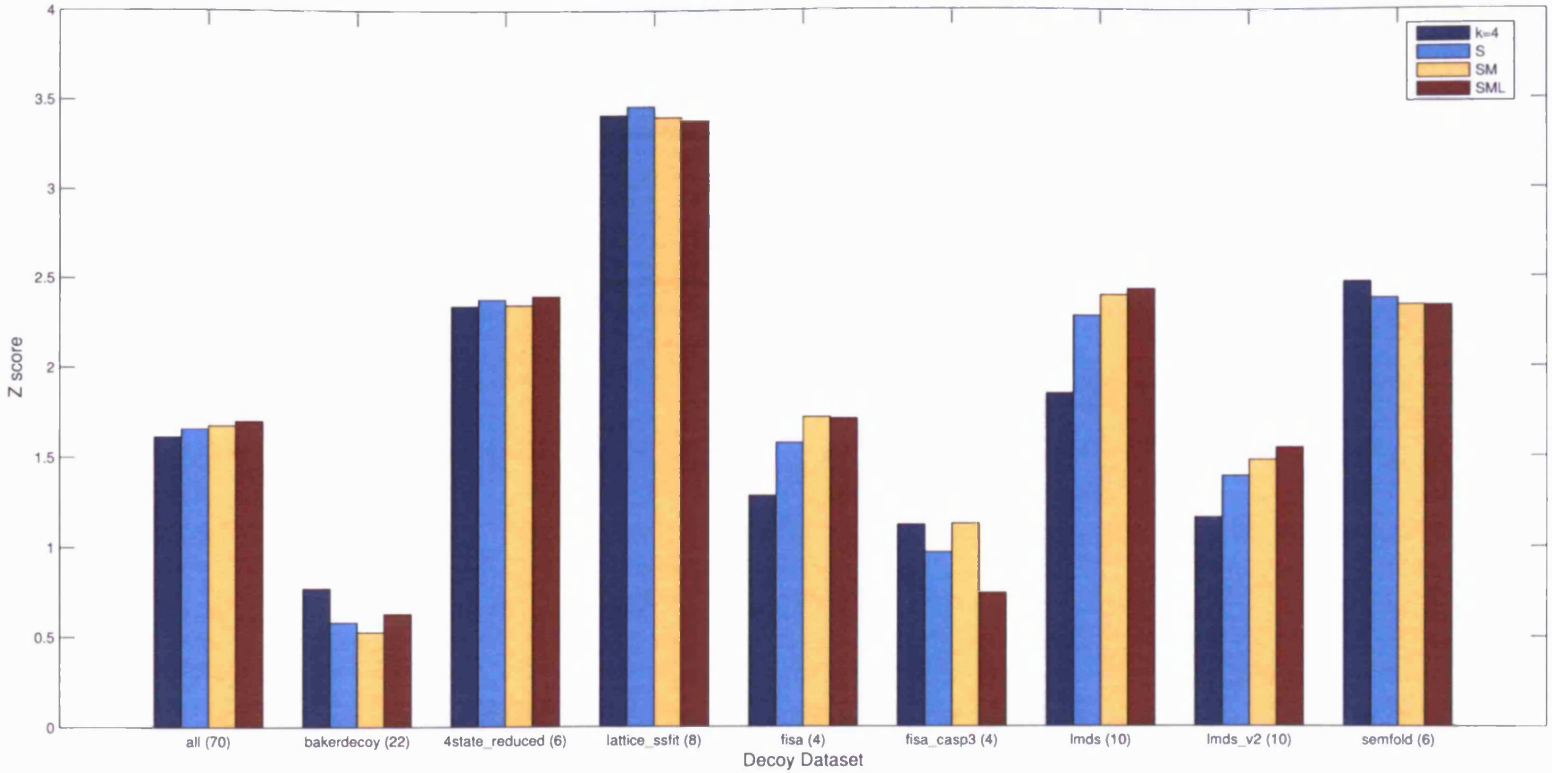


Figure 2.42: Z scores produced by the k=4, S, SM and SML combinations of sequence separations of the NN-solvpairndist method on the different individual decoy datasets, including the combination of all the individual datasets

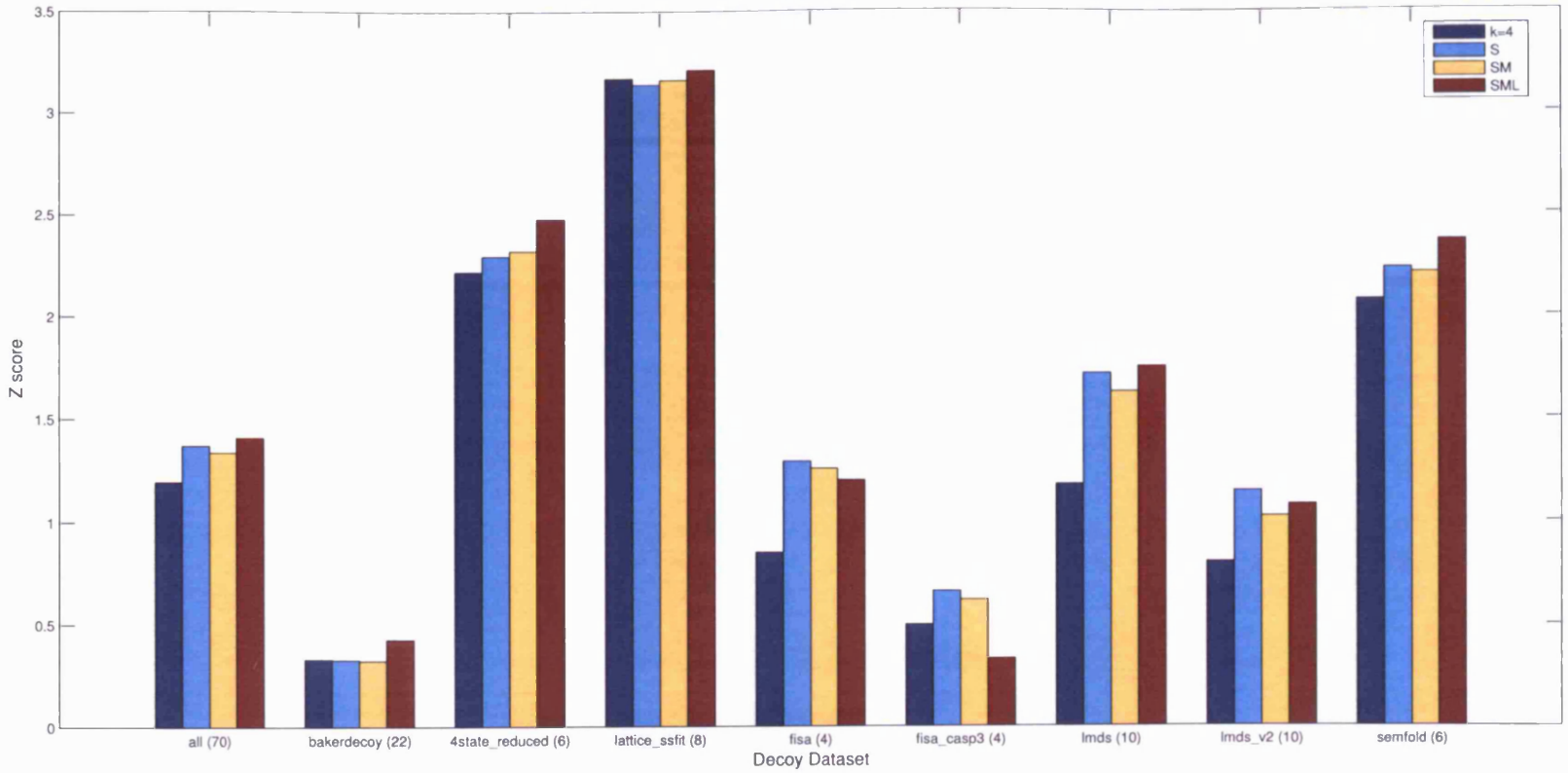


Figure 2.43: Z scores produced by the k=4, S, SM and SML combinations of sequence separations of the NN-solvpair method on the different individual decoy datasets, including the combination of all the individual datasets

2.5.5.2 Comparison of Results Across All Methods

In this section, the results of the NN-solvpairndist and NN-solvpair methods are benchmarked against those of the NN-dist method, the pairwise potentials method and K-Nearest Neighbours (K=10 and K=100) method. Figure 2.44 shows the detailed comparison of the various methods on the Baker decoy dataset, using the S combination.

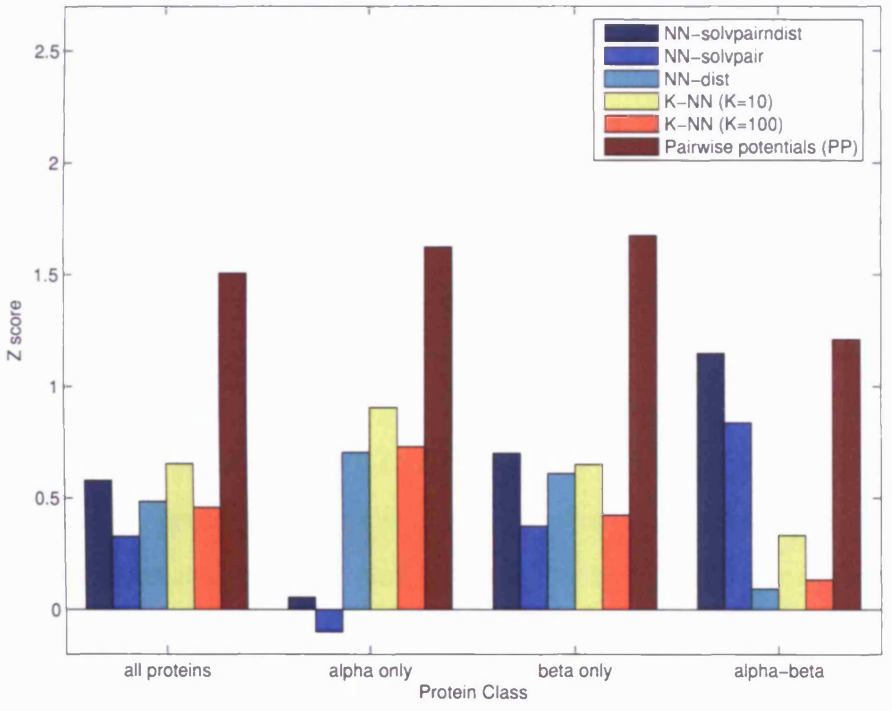


Figure 2.44: Z scores produced by the S combination of the NN-solvpairndist, NN-solvpair, NN-dist methods, the K-Nearest Neighbours methods (K=10, K=100) and the pairwise potentials method on the different secondary structural classes of the Baker decoy dataset

It can be seen from Figure 2.44 that for the S combination, the NN-solvpairndist and NN-solvpair methods do not perform well for α-only proteins in the Baker decoy dataset, compared to the rest of the methods. The reverse is true for $\alpha\beta$ proteins where the NN-solvpairndist and NN-solvpair methods have higher Z scores than the NN-dist method and the K-Nearest Neighbours methods. In all cases, the pairwise potentials method has the highest Z score and it is interesting to note that the NN-solvpairndist

method has a Z score which is only marginally lower than that of the pairwise potentials method for $\alpha\beta$ proteins.

On average, across all proteins, the pairwise potentials method has the highest Z score. The NN-solvpairndist method performs slightly better than the NN-dist method, while the K-Nearest Neighbours method (K=10) has a overall Z score which is slightly higher than the NN-solvpairndist method.

Figure 2.45 shows the Z scores for the S combination of all decoy datasets for the various methods. For the combined datasets, the pairwise potentials method has the highest Z score, while the NN-solvpairndist method has the second highest Z score.

Unlike Figure 2.29, the pairwise potentials method in Figure 2.45 does not have the highest Z score for every dataset. For the fisa, lmds and semfold datasets, the NN-solvpairndist method has the highest Z score instead. The NN-solvpairndist method also has the second highest Z score after the pairwise potentials method in the 4state_reduced, lattice_ssfit and fisa_casp3 datasets. This suggests that the NN-solvpairndist method shows some promise in matching the performance of the pairwise potentials method, if it can be further augmented with additional information.

In all but one case (lmds_v2), the NN-solvpairndist method has a higher Z score than the NN-dist method. The NN-solvpair method also performs better than the NN-dist method in all but two cases, namely the Baker decoy dataset and lmds_v2. The NN-solvpairndist method always has higher Z scores than the NN-solvpair method, which suggests that the additional distance information of the NN-solvpairndist method contributes to the discrimination of native structures.

One notable case is the fisa decoy dataset, where all other methods, except the NN-solvpairndist and NN-solvpair and the pairwise potentials methods, have negative Z scores. It appears that the NN-solvpairndist method does not do as well as the pairwise potentials method in the case of fisa_casp3, which is an α-only dataset. Having said

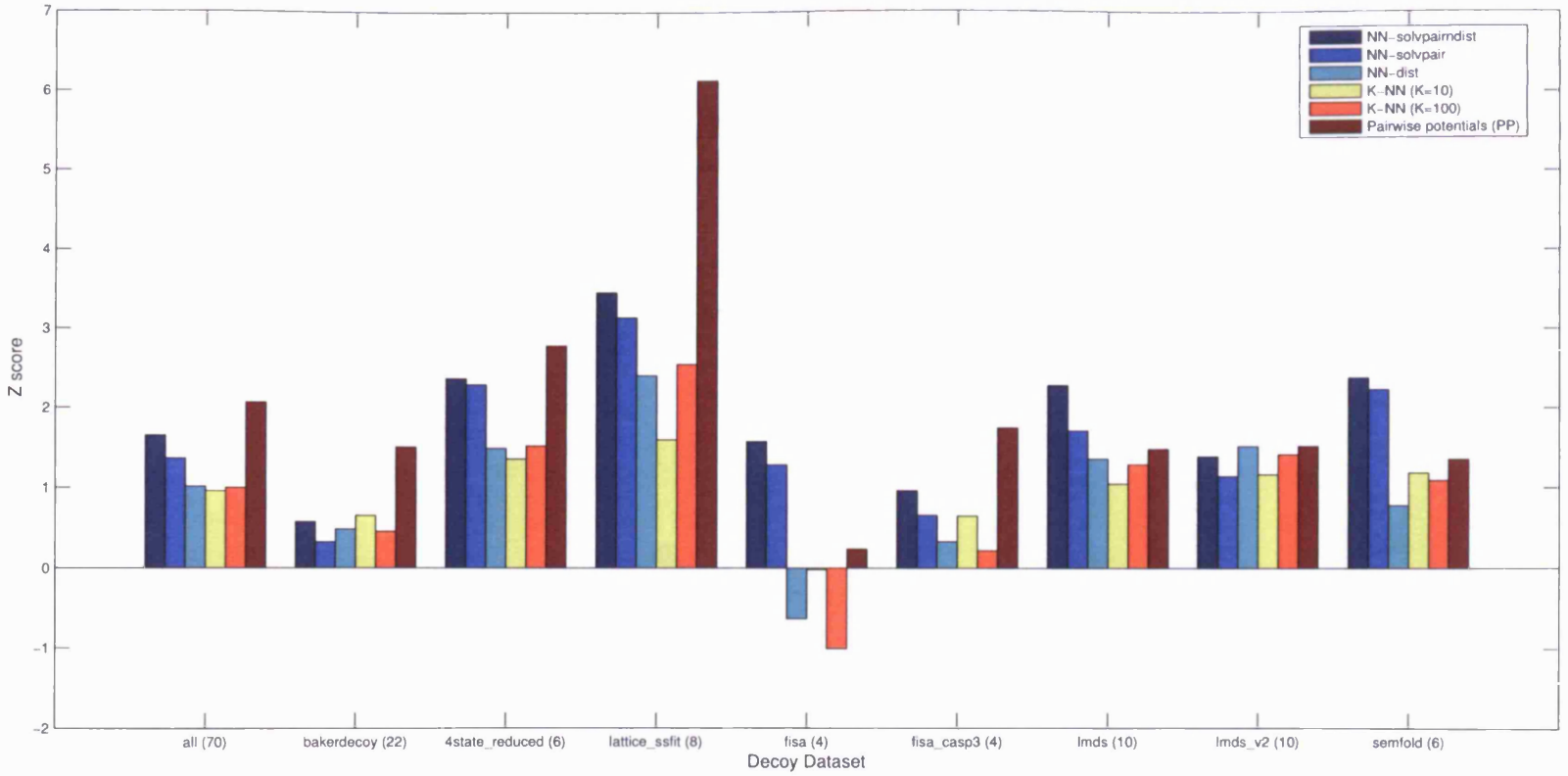


Figure 2.45: Z scores produced by the S combination of the NN-solvpairndist, NN-solvpair, NN-dist methods, the K-Nearest Neighbours methods (K=10, K=100) and the pairwise potentials method on the different individual decoy datasets, including the combination of all the individual datasets

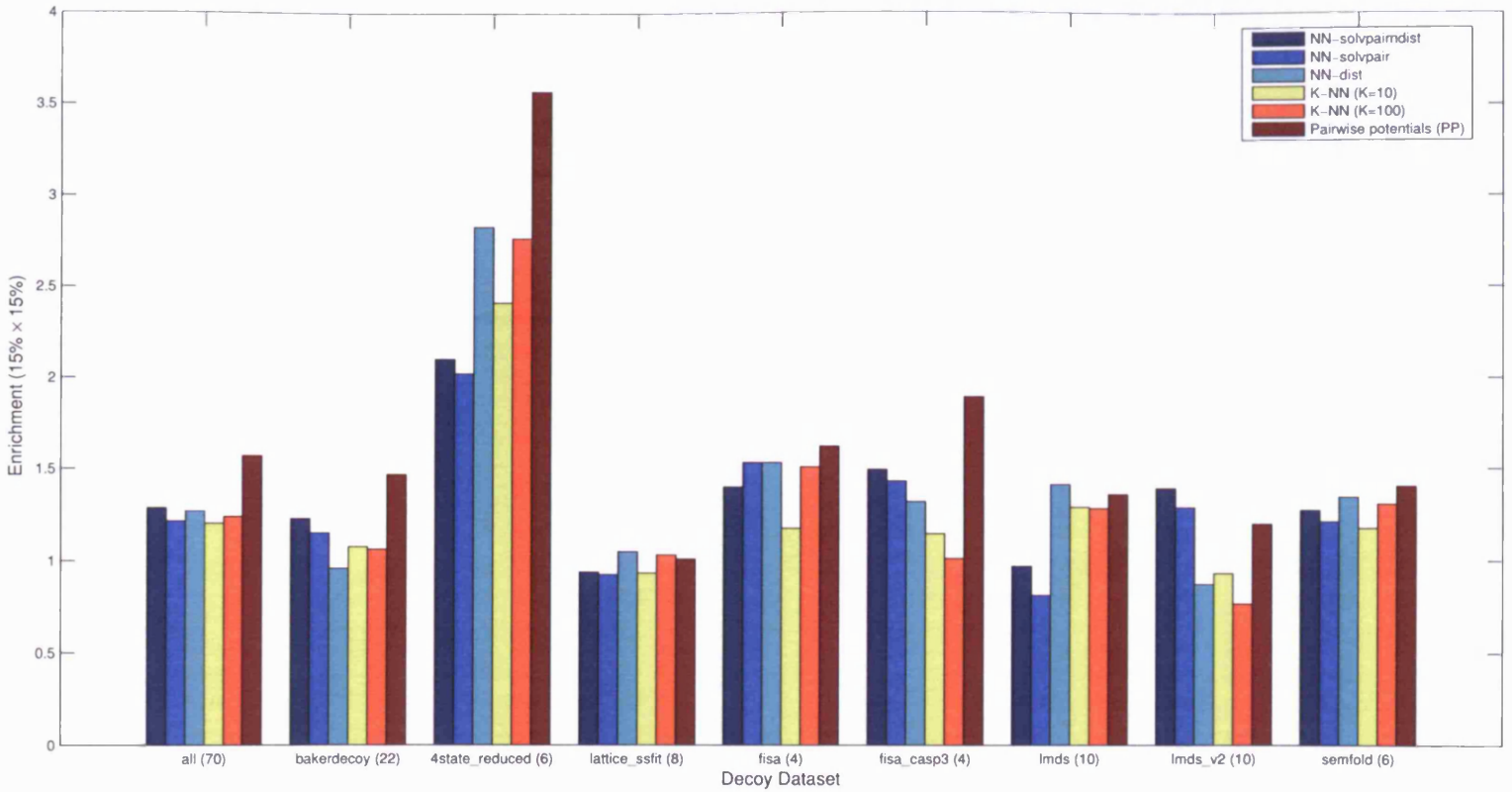


Figure 2.46: Enrichment scores (15% × 15%) produced by the S combination of the NN-solvpairndist, NN-solvpair, NN-dist methods, the K-Nearest Neighbours methods (K=10, K=100) and the pairwise potentials method on the different individual decoy datasets, including the combination of all the individual datasets

that, the pairwise potentials method does not necessarily have the best Z scores for all α-only datasets, like in the case of the fisa decoy dataset where the NN-solvpairndist method is the best (See Table 2.4 for the compositions of Decoys 'R' Us datasets).

For all but 2 decoy datasets (Baker and lmds_v2), the 2 K-NN methods have lower Z scores than the NN-solvpairndist method, although they are comparable to the NN-dist method in terms of Z score. Here it is worth reiterating that the definition of distance in the K-Nearest Neighbours method is restricted to the pairwise distance information only, as mentioned in Section 2.3.8.2. Potentially, the K-NN methods can be extended to include solvent accessibility information, by defining Euclidean distance measures that incorporate the new information. However, it is decided here that the focus is more on benchmarking against the pairwise potentials method.

Figure 2.46 shows the enrichment scores of the S combination across all decoy datasets for the different methods. For the combined datasets, the pairwise potentials method has the highest enrichment score, while the NN-solvpairndist method is comparable to the rest of the other methods. For most of the decoy datasets, there is no clear outstanding method which produces a distinctly high enrichment score, apart from the pairwise potentials method in the Baker, 4state_reduced and fisa_casp3 datasets.

It also seems that there is no significant improvement of the enrichment score (15% $\times$ 15%) for the NN-solvpairndist method over the NN-dist method. In fact, the NN-dist method has higher enrichment scores than the NN-solvpairndist and NN-solvpair methods in 4 out of 8 decoy datasets, namely 4state_reduced, lattice_ssfit, lmds and semfold. In the fisa dataset, the NN-dist method has a higher enrichment score than the NN-solvpairndist method, but performs similarly to the NN-solvpair method. All in all, in the combined dataset, the NN-solvpairndist method and NN-dist method have similar enrichment scores.

This suggests that while the extra solvent accessibility information used in the NN-solvpairndist method yields a noticeable increase in the Z score in the discrimination of native structures in Figure 2.45, it does not seem to increase the enrichment, which

measures the extent of association of low RMSD structures with high network output scores.

The following statistical analysis focuses on comparing the neural network methods against the pairwise potentials methods, and hence the poorly performing K-Nearest Neighbours methods are left out in subsequent analysis.

2.5.6 Results of Wilcoxon Sign-Rank Tests on Top Model Selection

In this section, the results of the one-tailed Wilcoxon sign-rank test on top model selection are presented. As described in Section 2.3.6.1, the null hypothesis is that the median is zero for the distribution of the differences in the structural similarity score (TM-score, GDT-TS or MaxSub) of the highest ranked model produced by the proposed decoy discrimination method (NN-dist, NN-solvpair or NN-solvpairndist) and the pairwise potentials method. The network scores produced by the NN-dist, NN-solvpair and NN-solvpairndist methods are of the S combination.

Tables 2.13, 2.14 and 2.15 show the P-values obtained from the Wilcoxon sign-rank tests with the structural similarity measures defined as TM-score, GDT-TS and Max-Sub respectively.

Each of these tables shows the P-values obtained from the comparison of NN-dist, NN-solvpair and NN-solvpairndist methods with the pairwise potentials method. For the sake of comparison with an existing MQAP method, the in-house MODCHECK [91] MQAP method is also used for hypothesis testing to see if the proposed neural network methods can outperform the competitive MODCHECK MQAP method in top model selection.

Each of these comparisons of a proposed neural network method with either the pairwise potentials method or MODCHECK is done for all decoy datasets, including the entire combined decoy datasets (all), and secondary structural classes of the combined

datasets (α-only, β-only, $\alpha\beta$).

It can be seen from Tables 2.13 to 2.15 that there is no P-value ≤ 0.05. This means that the null hypotheses for each of the structural similarity measures cannot be rejected at 5% significance level. This in turn means that the hypotheses that the median of the distribution of the differences in the structural similarity scores of the highest ranked model produced by each of the proposed neural network methods, and the pairwise potentials method (and MODCHECK) is zero cannot be rejected at 5% significance level.

The significance level is then relaxed to 10% to see if there are any P-values ≤ 0.10, and the results are

- NN-solvpairndist and pairwise potentials, lmds_v2 decoy dataset, TM-score, P-value = 0.0820
- NN-solvpairndist and pairwise potentials, combined (all) dataset, MaxSub, P-value = 0.0773
- NN-solvpair and pairwise potentials, $\alpha\beta$ dataset, GDT-TS, P-value = 0.0989
- NN-dist and MODCHECK, combined (all) dataset, MaxSub, P-value = 0.0685

In Figure 2.46, the NN-solvpairndist method has a higher enrichment score than the pairwise potentials method for the lmds_v2 dataset. Hence the first result is perhaps not too surprising. There is no evidence in the enrichment plots for the rest of the 3 observations, and the low P-values are probably due to chance.

Table 2.16 shows the results of the one-tailed Wilcoxon sign-rank test between the NN-solvpairndist method, and the other two NN-solvpair and NN-dist methods, at a

Decoy Dataset	NN-dist		NN-solvpair		NN-solvpairndist	
	PP	**MODCHECK**	**PP**	**MODCHECK**	**PP**	**MODCHECK**
4state_reduced	0.9375	0.9375	0.9531	0.8438	0.9531	0.8125
bakerdecoys	0.4228	0.4612	0.4935	0.4806	0.4164	0.4935
fisa_casp3	0.9375	0.9375	0.8125	0.5625	0.9375	0.8125
fisa	0.8750	0.6875	0.6875	0.5000	0.6875	0.5000
lattice_ssfit	0.9922	0.8438	0.8125	0.5000	0.8516	0.5000
lmds	0.5771	0.1162	0.8623	0.7539	0.7842	0.4609
lmds_v2	0.7842	0.9678	0.1250	0.7148	**0.0820**	0.1797
semfold	0.2188	0.5000	0.9688	0.9375	0.9688	0.9062
all	0.9855	0.9484	0.9621	0.8836	0.9550	0.2441
$\alpha\beta$	0.7270	0.2344	0.8120	0.2730	0.2344	0.3586
α-only	0.9624	0.8459	0.9912	0.8374	0.9951	0.2981
β-only	0.7695	0.9922	0.5781	0.8750	0.5781	0.8750

Table 2.13: Top Model Selection : P-values of one-tailed Wilcoxon sign-rank test between the NN-dist, NN-solvpair, NN-solvpairndist methods and Pairwise Potentials and MODCHECK, with TM-score as the structural similarity measure

Decoy Dataset	NN-dist		NN-solvpair		NN-solvpairndist	
	PP	MODCHECK	PP	MODCHECK	PP	MODCHECK
4state_reduced	0.8750	0.8438	0.9688	0.9062	0.9688	0.8750
bakerdecoys	0.2633	0.6334	0.5065	0.8390	0.6334	0.8943
fisa_casp3	0.9375	0.8750	0.8750	0.4375	0.9375	0.4375
fisa	0.8125	0.6875	0.6875	0.5625	0.6875	0.5625
lattice_ssfit	0.9883	0.7266	0.5938	0.2383	0.5312	0.1914
lmds	0.1611	0.1611	0.9033	0.9199	0.8623	0.7842
lmds_v2	0.5771	0.9033	0.1504	0.3262	0.1016	0.2129
semfold	0.5781	0.6562	0.9844	0.9375	0.9844	0.9375
all	0.8897	0.9187	0.9682	0.9657	0.9692	0.9350
$\alpha\beta$	0.7895	0.9080	**0.0989**	0.2344	0.1671	0.4046
α-only	0.8599	0.7834	0.9917	0.9082	0.9978	0.8891
β-only	0.7266	0.9922	0.6797	0.9453	0.6797	0.9453

Table 2.14: Top Model Selection : P-values of one-tailed Wilcoxon sign-rank test between the NN-dist, NN-solvpair, NN-solvpairndist methods and Pairwise Potentials and MODCHECK, with GDT-TS as the structural similarity measure

Decoy Dataset	NN-dist		NN-solvpair		NN-solvpairndist	
	PP	**MODCHECK**	**PP**	**MODCHECK**	**PP**	**MODCHECK**
4state_reduced	0.8750	0.8438	0.9688	0.8438	0.9688	0.8750
bakerdecoys	0.4164	0.4935	0.4516	0.4101	0.4677	0.4228
fisa_casp3	0.9375	0.9375	0.8125	0.6875	0.8750	0.7500
fisa	0.8750	0.8125	0.6875	0.4375	0.6875	0.4375
lattice_ssfit	0.9844	0.3359	0.8906	0.2734	0.8906	0.2734
lmds	0.6152	0.2480	0.9033	0.8389	0.6875	0.5391
lmds_v2	0.5000	0.8125	0.2852	0.2852	0.1797	0.2852
semfold	0.3125	0.5000	0.8438	0.7812	0.3438	0.5000
all	0.9844	**0.0685**	0.9669	0.2806	**0.0773**	0.4261
$\alpha\beta$	0.2866	0.4308	0.1918	0.1880	0.1572	0.2163
α-only	0.9921	0.9422	0.9917	0.3688	0.9966	0.4011
β-only	0.6289	0.9727	0.4219	0.8750	0.4219	0.8750

Table 2.15: Top Model Selection : P-values of one-tailed Wilcoxon sign-rank test between the NN-dist, NN-solvpair, NN-solvpairndist methods and Pairwise Potentials and MODCHECK, with MaxSub as the structural similarity measure

Decoy Dataset	NN-solvpairndist					
	NN-solvpair			NN-dist		
	TM-score	GDT-TS	MaxSub	TM-score	GDT-TS	MaxSub
4state_reduced	0.2500	0.2500	0.5000	0.6562	0.6562	0.6562
bakerdecoys	0.5000	0.5000	0.4062	0.2850	0.6456	0.1652
fisa_casp3	0.9375	0.9375	0.6875	0.0625	0.1875	0.1250
fisa	0.5000	0.5000	0.5000	0.4375	0.4375	0.3125
lattice_ssfit	0.5000	0.5000	0.5000	0.2734	0.1562	0.5781
lmds	0.5000	0.2500	0.2500	0.9473	0.9814	0.8516
lmds_v2	0.2500	0.5000	0.5000	**0.0186**	0.0801	0.1250
semfold	0.5000	0.5000	0.2500	0.8438	0.9219	0.5000
all	0.1586	0.1803	0.1494	0.2685	0.6340	0.2219
$\alpha\beta$	0.4727	0.4219	0.2852	0.1236	0.1161	0.1531
α-only	0.3823	0.4492	0.5000	0.2916	0.8505	0.4527
β-only	0.5000	0.5000	0.5000	0.0547	0.1250	**0.0391**

Table 2.16: Top Model Selection : P-values of one-tailed Wilcoxon sign-rank test between the NN-solvpairndist method and the NN-dist, NN-solvpair methods

5% significance level.

The results are

- NN-solvpairndist and NN-dist, lmds_v2 dataset, TM-score, P-value = 0.0186
- NN-solvpairndist and NN-dist, β-only dataset, MaxSub, P-value = 0.0391

For the NN-solvpairndist and NN-solvpair methods, the null hypothesis that the median of the distribution of the differences in structural similarity scores produced by both methods is zero cannot be rejected at 5% significance level.

For the NN-solvpairndist and NN-dist methods, the null hypothesis can be rejected on two cases, as shown above.

2.5.7 Results of Wilcoxon Sign-Rank Tests on Spearman correlation coefficients

In this section, the results of the one-tailed Wilcoxon sign-rank test on the matched pairs of Spearman correlation coefficients produced by various pairs of decoy discrimination methods are presented. As described in Section 2.3.6.2, the null hypothesis is that the median is zero for the distribution of the differences in the Spearman correlation coefficients between a structural similarity score (TM-score, GDT-TS or MaxSub) and the output scores produced by the proposed decoy discrimination method (NN-dist, NN-solvpair or NN-solvpairndist) and the output scores produced by the pairwise potentials method. The network scores produced by the NN-dist, NN-solvpair and NN-solvpairndist methods are of the S combination.

Again, as in Section 2.5.6, the in-house MODCHECK method is also used for hypothesis testing to see if the proposed neural network methods can outperform the competitive MODCHECK MQAP method in terms of the ranking of the models.

Tables 2.17, 2.18 and 2.19 show the P-values obtained from the Wilcoxon sign-rank tests with the structural similarity measures defined as TM-score, GDT-TS and Max-

Sub respectively.

Tables 2.17, 2.18 and 2.19 show the P-values obtained from the one-tailed Wilcoxon sign-rank test with the structural similarity measures defined as TM-score, GDT-TS and MaxSub respectively.

Each of these tables shows the P-values obtained from the comparison of NN-dist, NN-solvpair and NN-solvpairndist methods with the pairwise potentials method. For the sake of comparison with an existing MQAP method, the in-house MODCHECK [91] MQAP method is also used in place of the pairwise potentials for hypothesis testing. Each of these comparisons of a proposed neural network method with either the pairwise potentials method or MODCHECK is done for all decoy datasets, including the entire combined decoy datasets (all), and secondary structural classes of the combined datasets (α-only, β-only, $\alpha\beta$).

It can be seen from Tables 2.17 to 2.19 that there is no P-value ≤ 0.05. This means that for all cases, the null hypotheses for each of the structural similarity measures cannot be rejected at 5% significance level. This in turn means that the hypotheses that the median of the distribution of the differences in Spearman correlation coefficients produced by each of the proposed neural network methods, and the pairwise potentials method (and MODCHECK) is zero cannot be rejected at 5% significance level.

To put it simply, the proposed methods are not better in ranking the models according to their structural similarity to the native structure (as defined by TM-score, GDT-TS and MaxSub) than either the pairwise potentials method or MODCHECK, when tested

Decoy Dataset	NN-dist		NN-solvpair		NN-solvpairndist	
	PP	MODCHECK	PP	MODCHECK	PP	MODCHECK
4state_reduced	0.9219	0.5000	0.9844	0.9844	0.9844	0.9844
bakerdecoys	0.9858	0.9890	0.9922	0.9880	0.8883	0.9001
fisa_casp3	0.9375	0.9375	0.6875	0.6875	0.8125	0.8750
fisa	0.9375	0.8750	0.8125	0.8750	0.8750	0.9375
lattice_ssfit	0.9805	0.9805	0.5273	0.6797	0.3203	0.5781
lmds	0.8623	0.5000	0.9980	0.9971	0.9863	0.9902
lmds_v2	0.9473	0.9033	0.6875	0.8838	0.7217	0.8623
semfold	0.9844	0.9844	0.9688	0.9844	0.9844	0.9844
all	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
$\alpha\beta$	0.9995	0.9995	0.9977	0.9996	0.9875	0.9907
α-only	0.9999	0.9957	0.9997	0.9991	0.9996	0.9991
β-only	0.9727	0.9961	0.8086	0.9961	0.7695	0.9961

Table 2.17: Spearman correlation coefficient : P-values of one-tailed Wilcoxon sign-rank test between the NN-dist, NN-solvpair, NN-solvpairndist methods and Pairwise Potentials and MODCHECK, with TM-score as the structural similarity measure

Decoy Dataset	NN-dist		NN-solvpair		NN-solvpairndist	
	PP	MODCHECK	PP	MODCHECK	PP	MODCHECK
4state_reduced	0.7812	0.5000	0.9844	0.9844	0.9844	0.9844
bakerdecoys	0.9935	0.9915	0.9992	0.9992	0.9907	0.9804
fisa_casp3	0.9375	0.9375	0.8125	0.8125	0.8750	0.8125
fisa	0.9375	0.8750	0.8125	0.8750	0.8750	0.9375
lattice_ssfit	0.9805	0.9805	0.5000	0.7266	0.2734	0.7266
lmds	0.6152	0.4229	0.9951	0.9971	0.9902	0.9863
lmds_v2	0.8838	0.8623	0.5771	0.7842	0.7217	0.7842
semfold	0.9844	0.9844	0.9844	0.9844	0.9844	0.9844
all	1.0000	0.9999	1.0000	1.0000	1.0000	1.0000
$\alpha\beta$	0.9989	0.9989	0.9988	0.9997	0.9960	0.9931
α-only	0.9998	0.9907	1.0000	0.9999	1.0000	0.9999
β-only	0.9258	0.9961	0.8438	0.9961	0.4219	0.9961

Table 2.18: Spearman correlation coefficient : P-values of one-tailed Wilcoxon sign-rank test between the NN-dist, NN-solvpair, NN-solvpairndist methods and Pairwise Potentials and MODCHECK, with GDT-TS as the structural similarity measure

Decoy Dataset	NN-dist		NN-solvpair		NN-solvpairndist	
	PP	MODCHECK	PP	MODCHECK	PP	MODCHECK
4state_reduced	0.9531	0.5000	0.9844	0.9844	0.9844	0.9844
bakerdecoys	0.9922	0.9922	0.9890	0.9899	0.9302	0.9057
fisa_casp3	0.9375	0.9375	0.8125	0.6875	0.8125	0.6875
fisa	0.9375	0.8125	0.8125	0.8750	0.8750	0.9375
lattice_ssfit	0.6289	0.5000	0.9023	0.9727	0.6797	0.8086
lmds	0.5000	0.3477	0.9902	0.9814	0.9814	0.9756
lmds_v2	0.6875	0.6875	0.4229	0.5771	0.4609	0.5391
semfold	0.9844	0.9688	0.9844	0.9844	0.9844	0.9844
all	1.0000	0.9989	1.0000	1.0000	0.9999	0.9999
$\alpha\beta$	0.9907	0.9886	0.9938	0.9964	0.9535	0.9458
α-only	0.9985	0.9658	0.9993	0.9975	0.9990	0.9971
β-only	0.9883	0.9961	0.8086	0.9961	0.8086	0.9961

Table 2.19: Spearman correlation coefficient : P-values of one-tailed Wilcoxon sign-rank test between the NN-dist, NN-solvpair, NN-solvpairndist methods and Pairwise Potentials and MODCHECK, with MaxSub as the structural similarity measure

Decoy Dataset	NN-solvpairndist					
	NN-solvpair			NN-dist		
	TM-score	**GDT-TS**	**MaxSub**	**TM-score**	**GDT-TS**	**MaxSub**
4state_reduced	**0.0312**	**0.0312**	**0.0312**	0.9688	0.9688	0.9688
bakerdecoys	**0.0001**	**0.0001**	**0.0006**	0.0889	0.2132	0.0999
fisa_casp3	0.3125	0.5000	0.3125	0.0625	0.0625	0.0625
fisa	0.9375	0.9375	0.9375	0.1875	0.1875	0.1875
lattice_ssfit	**0.0195**	0.0742	**0.0117**	**0.0117**	**0.0273**	0.6289
lmds	**0.0322**	**0.0137**	**0.0010**	0.8838	0.9346	0.8838
lmds_v2	0.4609	0.5391	0.5771	0.5000	0.5000	0.5771
semfold	0.0781	0.1562	0.0781	0.2812	0.5000	0.4219
all	**4.2e-5**	**9.4e-5**	**8.3e-6**	0.2208	0.4246	0.5869
$\alpha\beta$	**0.0004**	**0.0007**	**0.0001**	0.0727	0.1753	0.1851
α-only	0.0682	0.0540	**0.0357**	0.3052	0.4532	0.4377
β-only	0.0977	0.0742	0.1250	0.3711	0.4219	0.3203

Table 2.20: Spearman correlation coefficient : P-values of one-tailed Wilcoxon sign-rank test between the NN-solvpairndist method and the NN-dist, NN-solvpair methods

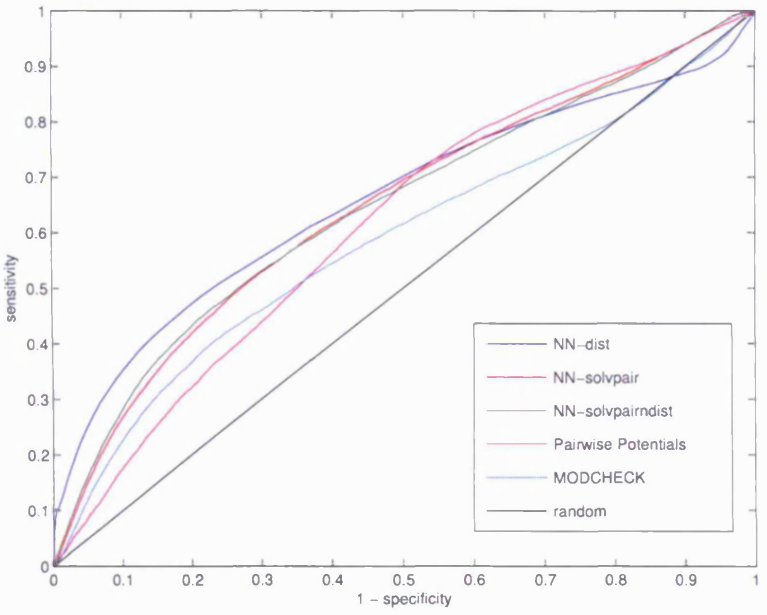


Figure 2.47: ROC plots of the NN-dist, NN-solvpair, NN-solvpairndist methods, Pairwise Potentials and MODCHECK using **RMSD** $\leq$ **6Å** as the threshold for 'true data' on all decoy datasets

with a one-tailed Wilcoxon sign-rank test at 5% significance level.

The NN-solvpairndist method is then subjected to the same Wilcoxon sign-rank test to see if it is better than either the NN-solvpair method or NN-dist method in ranking the decoy models. Table 2.20 shows the P-values obtained from the one-tailed test.

It can be seen from Table 2.20 that in many cases, the NN-solvpairndist method produces higher Spearman correlation coefficients than the NN-solvpair method. In contrast, there are only two cases in the statistical tests where the NN-solvpairndist method produces higher Spearman correlation coefficients than the NN-dist method. This leads to the conclusion that solvent accessibility information alone, in the context of the proposed neural networks method, is not enough to rank good quality decoy models. It appears that pairwise distance information is vital in ranking the decoys according to their quality.

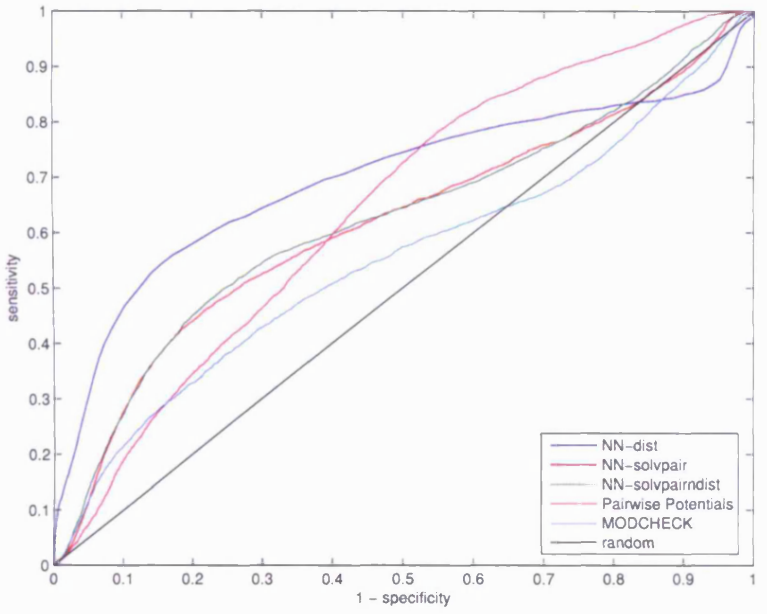


Figure 2.48: ROC plots of the NN-dist, NN-solvpair, NN-solvpairndist methods, Pairwise Potentials and MODCHECK using **RMSD** $\leq$ **4Å** as the threshold for 'true data' on all decoy datasets

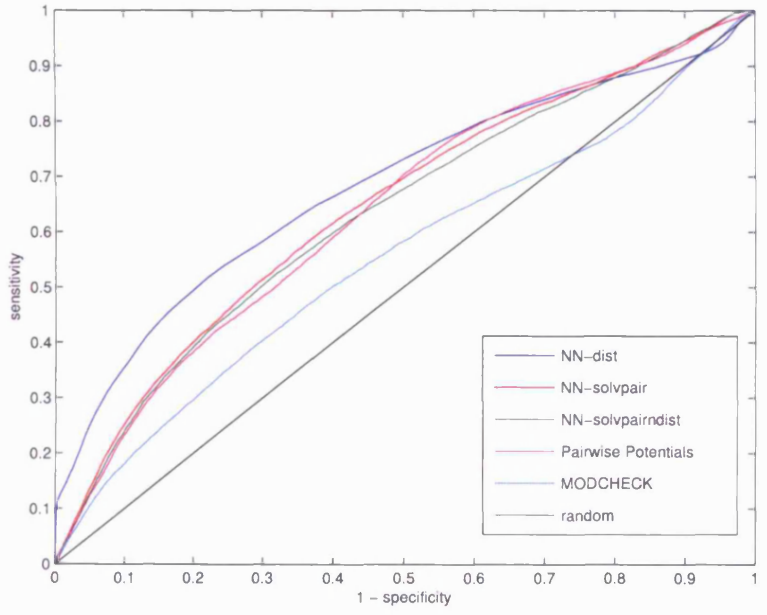


Figure 2.49: ROC plots of the NN-dist, NN-solvpair, NN-solvpairndist methods, Pairwise Potentials and MODCHECK using **TM-score** $\geq$ **0.4** as the threshold for 'true data' on all decoy datasets

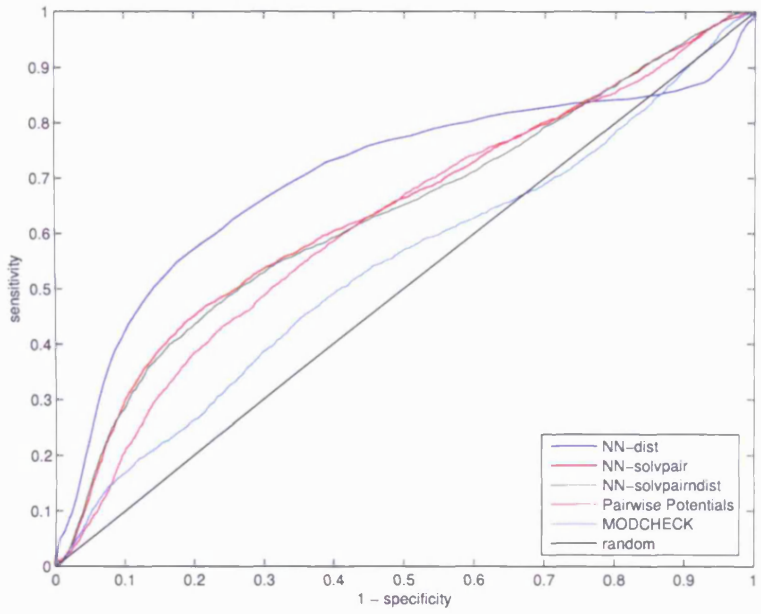


Figure 2.50: ROC plots of the NN-dist, NN-solvpair, NN-solvpairndist methods, Pairwise Potentials and MODCHECK using **TM-score** $\geq$ **0.5** as the threshold for 'true data' on all decoy datasets

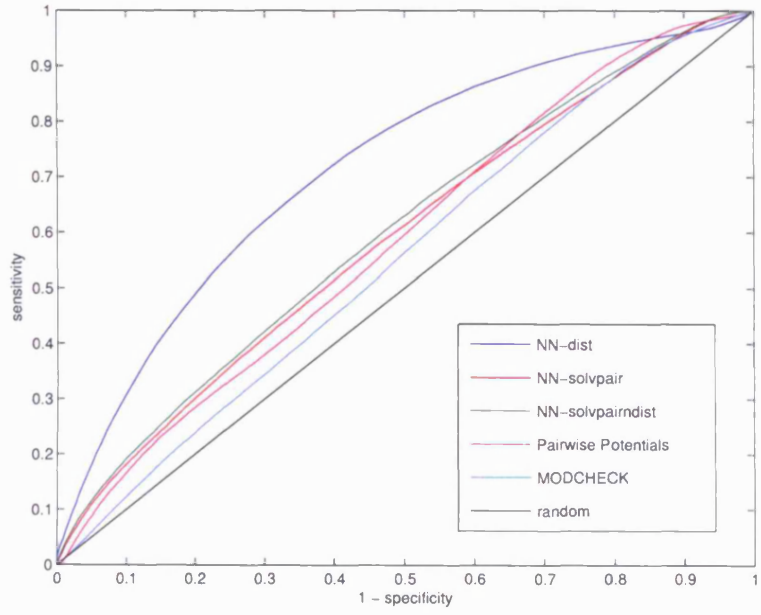


Figure 2.51: ROC plots of the NN-dist, NN-solvpair, NN-solvpairndist methods, Pairwise Potentials and MODCHECK using **GDT-TS score** $\geq$ **0.25** as the threshold for 'true data' on all decoy datasets

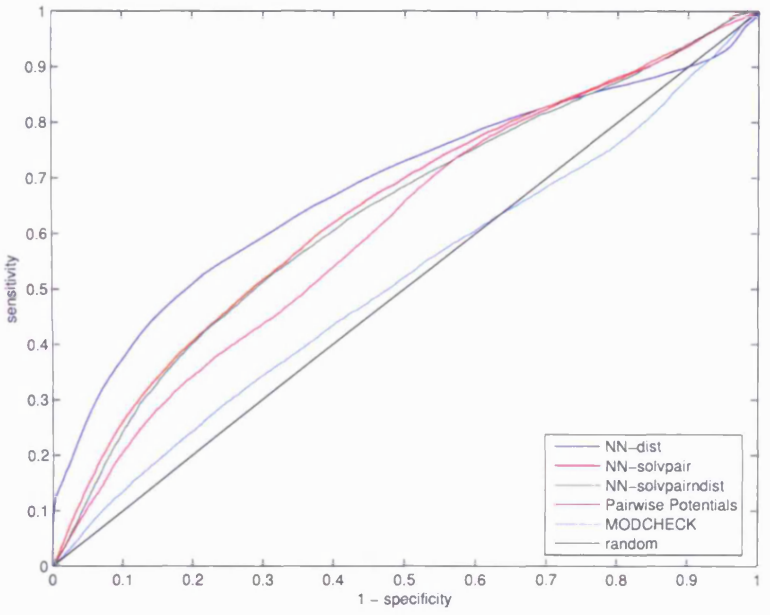


Figure 2.52: ROC plots of the NN-dist, NN-solvpair, NN-solvpairndist methods, Pairwise Potentials and MODCHECK using **GDT-TS score** $\geq$ **0.35** as the threshold for 'true data' on all decoy datasets

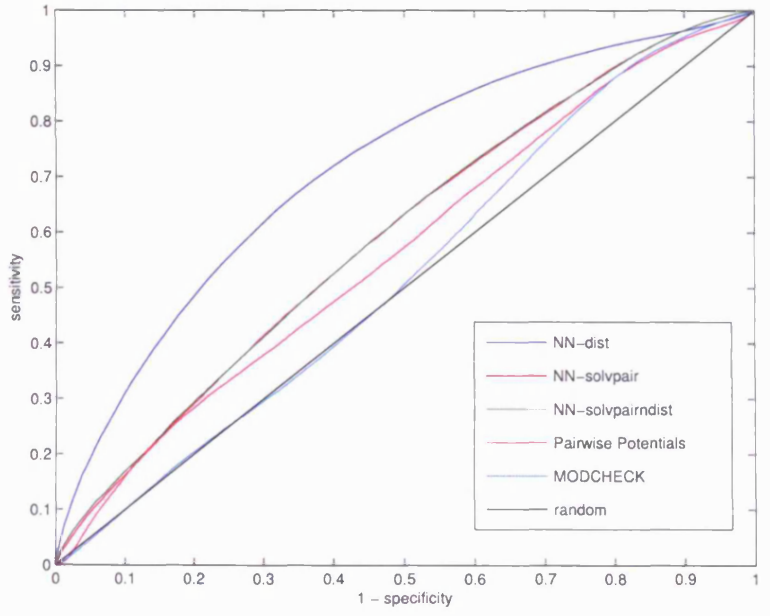


Figure 2.53: ROC plots of the NN-dist, NN-solvpair, NN-solvpairndist methods, Pairwise Potentials and MODCHECK using **MaxSub score** $\geq$ **0.3** as the threshold for 'true data' on all decoy datasets

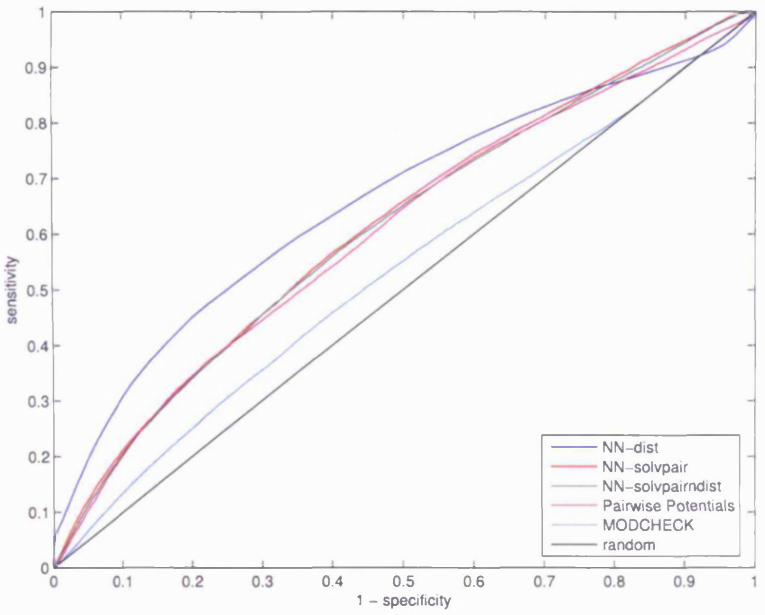


Figure 2.54: ROC plots of the NN-dist, NN-solvpair, NN-solvpairndist methods, Pairwise Potentials and MODCHECK using **MaxSub score** $\geq$ **0.4** as the threshold for 'true data' on all decoy datasets

2.5.8 Results of ROC Analysis

This section investigates how the various neural network decoy discrimination methods, including the pairwise potentials method and MODCHECK, can classify the decoy models, if the available decoy models are dichotomized into 'true' and 'false' classes. The ROC curves are drawn for each structural similarity measure, as shown in Figures 2.47 to 2.54.

As mentioned in Section 2.3.6.3, there are two sets of thresholds for the dichotomy. The first set is 6Å, 0.4, 0.25 and 0.3 for RMSD, TM-score, GDT-TS and MaxSub respectively; the second set is 4Å, 0.5, 0.35 and 0.4 for RMSD, TM-score, GDT-TS and MaxSub respectively. There are altogether 142625 models in the 70 decoy sets from the 8 decoy datasets. All the models whose corresponding structural similarity measures are below the threshold are considered 'false' models, and vice versa.

Figures 2.47 and 2.48 show the ROC plots for RMSD $\leq$ 6Å and RMSD $\leq$ 4Å as the thresholds for 'true data' respectively.

Figures 2.49 and 2.50 show the ROC plots for TM-score $\geq$ 0.4 and TM-score $\geq$ 0.5 as the thresholds for 'true data' respectively.

Figures 2.51 and 2.52 show the ROC plots for GDT-TS $\geq$ 0.25 and GDT-TS $\geq$ 0.35 as the thresholds for 'true data' respectively.

Figures 2.53 and 2.54 show the ROC plots for MaxSub $\geq$ 0.3 and MaxSub $\geq$ 0.4 as the thresholds for 'true data' respectively.

In all figures, the NN-dist method, perhaps somewhat surprisingly, has the highest values of specificities (lowest values of (1-specificity)) for sensitivities of $\leq$ 0.85-0.95, when compared to all other methods. This means that for sensitivities of up to 0.85 to 0.95, the fraction of false positives to 'false' data is the lowest for the NN-dist method. In all figures, it can be seen, across all structural similarity measures, that the NN-solvpair, NN-solvpairndist and the pairwise potentials method have comparable ROC plots, while the MODCHECK method appears to perform among the worst of all methods.

In general, the area under the ROC curve of the NN-dist method is the largest of all the methods for all figures. This suggests that the NN-dist method is better than all other methods, including the pairwise potentials method, in binary classification of decoy structures. The same results are obtained for RMSD as well as TM-score, GDT-TS and MaxSub. This is somewhat surprising because the NN-solvpairndist method and the pairwise potentials method perform better than the NN-dist method for the discrimination of the native structure (Z score) in Figure 2.45. While the NN-solvpairndist method and NN-dist method have similar overall enrichment scores, the pairwise potentials method outperforms the NN-dist method for the enrichment score, as shown in Figure 2.46.

To investigate this observation further, 3D plots of the structural similarity scores against the outputs of the NN-dist, NN-solvpairndist, and pairwise potentials methods

are shown in Figures 2.55 to 2.57.

Figures 2.55, 2.56 and 2.57 show the 3D plots of the RMSDs of the decoy structures against output scores of the NN-dist, NN-solvpairndist and pairwise potentials method respectively. In all 3 figures, one dashed line at 4Å dichotomizes each plot into 'true' and 'false' data, and another dashed line is the varying threshold that yields the ROC curve across the range of sensitivities/specificities.

The arrows in each plot indicates the region of false positives, where decoy models have high RMSD ('false' data) but are assigned high network scores or low energies ('positive' assignment) by the decoy discrimination method. Here, it appears from the distributions that the NN-solvpairndist method in Figure 2.56 yields comparatively higher percentage of false positives than the NN-dist method in Figure 2.55 at approximately median thresholds.

It is interesting to note that the enrichment score in Equation 2.6 on page 103, which captures the top 15% of low RMSD decoys with high network scores or low energies, focuses only on the ratio of the number of 'true positives' identified by the decoy discrimination method to that of a uniform distribution. The enrichment score therefore does not measure the quantity of false positives, and hence the high false positive trends of the NN-solvpairndist method and pairwise potentials method can only be seen from ROC plots.

While the 3D plots in Figures 2.55 to 2.57 can explain the smaller areas of the NN-solvpairndist method in the ROC curves in Figures 2.47 to 2.54, it is perhaps worth noting that in the context of structure prediction experiments such as CASP, the emphasis is to identify the top few 'true positive' models as probable predictions, and hence the issue of large numbers of false positives assigned by a decoy discrimination method is, while relevant, perhaps not very crucial.

In conclusion, the ROC analysis provides another perspective of the performance of the proposed decoy discrimination methods, apart from the Z score and the enrichment.

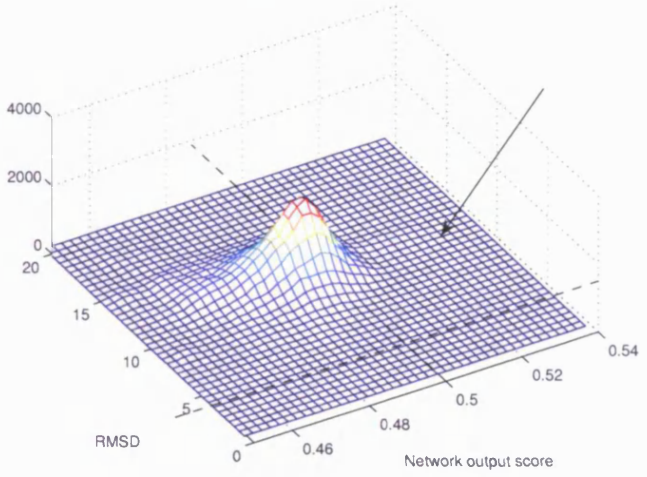


Figure 2.55: 3D plots of the RMSDs of the 142625 decoy structures versus the corresponding S combination of output scores produced by the NN-dist method

While the pairwise potentials method and the NN-solvpairndist method outperforms the NN-dist method in terms of the discrimination of native structure, they also generate higher percentages of false positives for a wide range of sensitivities, as shown in Figures 2.47 to 2.54.

2.6 Summary

This chapter introduces a novel decoy discrimination method using neural networks, which is referred to as the NN-dist method. The neural networks are trained on a set of data that includes native pairwise distances, and non native pairwise distances. The non native pairwise distances are simulated using native structures with their sequences reversed. 19 neural networks are trained on datasets, each representing a particular sequence separation value k, where $4 \leq k \leq 22$, and one network represents the sequence separation range k > 22.

The proposed decoy discrimination method is tested on different publicly available decoy datasets, namely the Baker decoy dataset and the Decoy 'R' Us suite of decoys. Different ways of combining the results of the neural networks are attempted, and it is found that the short range combination of network results ($4 \leq k \leq 10$) is the best for

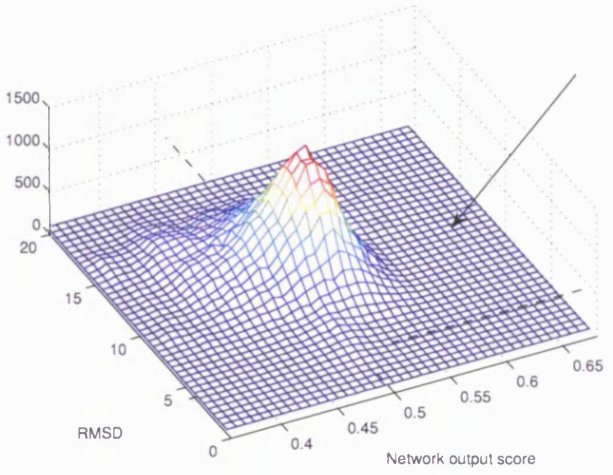


Figure 2.56: 3D plots of the RMSDs of the 142625 decoy structures versus the corresponding S combination of output scores produced by the NN-solvpairndist method

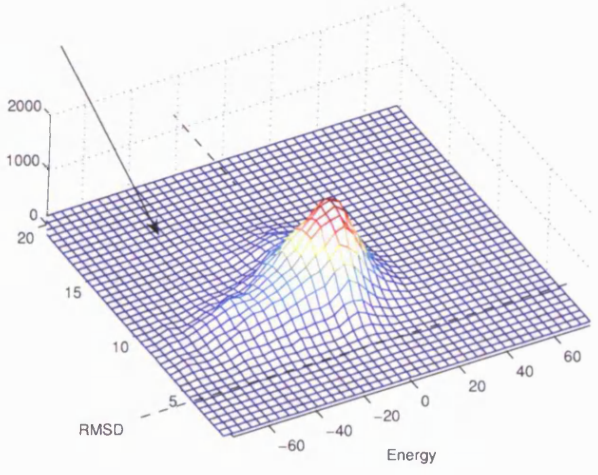


Figure 2.57: 3D plots of the RMSDs of the 142625 decoy structures versus the corresponding S combination of output scores produced by the pairwise potentials method

the NN-dist method.

The proposed methods are benchmarked against the pairwise potentials of mean force method, as well as the K-Nearest Neighbours method, where K is taken to be 10 and 100. The in-house tried and tested pairwise potentials method, which has proven competitive for the past few CASP experiments, is used for benchmarking so that a stringent test can be provided for the proposed neural network methods.

The benchmarking tests include the

- Z score, for measuring how many standard deviations the score of the native structure is away from the mean score of all decoys.
- enrichment, for the degree to which the method can associate low RMSD decoys with high output scores.
- top model selection using the Wilcoxon sign-rank test between each proposed machine learning method and the pairwise potentials method.
- ranking of the decoy models with Spearman rank correlation coefficient, which also uses the Wilcoxon sign-rank test between each proposed machine learning method and the pairwise potentials method.
- ROC analysis

Section 2.5.2 expands on the NN-dist method by introducing additional input features in the form of relative solvent accessibilities of the residue pairs. Two methods, NN-solvpair and NN-solvpairndist, are created; the former replacing the pairwise distance with the relative solvent accessibility values, the latter includes both types of information. For these 2 new methods, the training and validation datasets, the decoy datasets used, training algorithms and test measures remain the same as that in the NN-dist method.

The pairwise potentials method yields the highest Z score for the combined datasets, followed by the NN-solvpairndist method. While the pairwise potentials method has

the highest Z score for 4 out of 8 decoy datasets, as shown in Figure 2.45, the NN-solvpairndist method shows some promise by having the highest Z score for 3 decoy datasets.

The NN-solvpairndist method has higher Z scores compared to the original NN-dist method for all but 1 decoy datasets, as shown in Figure 2.45, for the S combination of network scores. This suggests that as far as the discrimination of the native structure from a set of decoys is concerned, the additional input features of relative solvent accessibilities are useful, in the context of the neural network method of decoy discrimination. In Figure 2.44, results have also suggested that the NN-solvpairndist method works well with $\alpha\beta$ proteins.

The NN-solvpairndist method outperforms the NN-solvpair method, in terms of Z score, for all decoy datasets, as shown in Figure 2.45. This suggests that the additional pairwise distance information, which is the difference between the two methods, does help in the discrimination of native structures from a set of decoys.

The K-Nearest Neighbours methods and the NN-dist method have the lowest Z scores in the combined dataset, as well as in the individual decoy datasets. For the NN-solvpairndist and NN-solvpair methods, the Z scores derived from the SM and SML combinations of the NN-solvpairndist method are comparable to that of the S combination, as shown in Figures 2.42 and 2.43.

For the enrichment measure, that is the association of high scores to low RMSD structures, the pairwise potentials method has the highest enrichment score among all the methods. The NN-solvpairndist method shows no marked improvement over that of the NN-dist method, as shown in Figure 2.46. The difference between the NN-solvpairndist and NN-solvpair methods is also small, as shown in Figure 2.46, suggesting that the additional distance information has little effect on the association of low RMSD models to high scores.

The conclusion of the one-tailed Wilcoxon sign-rank tests involving the top model

selection shows that at 5% significance level, there is no evidence to reject the hypothesis that the proposed neural networks (NN-dist, NN-solvpair, NN-solvpairndist) can perform better model selection than the pairwise potentials method. Different structural similarity scores, TM-score, GDT-TS and MaxSub, are used in the testing.

The same conclusion can be reached of the Wilcoxon sign-rank tests involving the Spearman correlation coefficients. At 5% significance level, there is no evidence to reject the hypothesis that the proposed neural networks (NN-dist, NN-solvpair, NN-solvpairndist) can rank the decoys better than the pairwise potentials method. Different structural similarity scores, TM-score, GDT-TS and MaxSub, are also used for the hypothesis.

The ROC analysis dichotomizes all the decoy models in the datasets into 'true' and 'false' classes. The ROC curves show that the pairwise potentials method performs similarly to the NN-solvpair and NN-solvpairndist methods. For all structural similarity measures, the NN-dist method has a lower false positive rate than the rest of the methods, for a wide range of sensitivities of up to 0.85-0.95, when the NN-dist method starts to have higher false positive rates than the other neural networks and pairwise potentials method.

It turns out that while the NN-solvpairndist and pairwise potentials method yield higher Z scores than the NN-dist method, they also yield higher false positive rates for a wide range of true positive rates. This is not reflected in the enrichment score, which only focus on the ratio of true positives to that of a uniform distribution. Hence, the ROC curves provide another informative perspective to the performance of a decoy discrimination method.

2.7 Conclusion

In this chapter, the proposed decoy discrimination methods, using neural networks and a variety of input features, are compared with the tried and tested pairwise potentials

method using a number of benchmarking measures.

While the various statistical tests show no improvement in the proposed neural network methods over the tried and tested pairwise potentials method in terms of top model selection and model ranking, the high Z scores of the NN-solvpairndist method is encouraging, and hence further work can be done on these methods, through the additional use of evolutionary information, in the next chapter in a bid to improve the performance in the various benchmarking measures.

To summarize, the most promising of the neural networks is the NN-solvpairndist method, which

- has the second highest Z score for the discrimination of native structures, after the pairwise potentials method.
- has the second highest enrichment score, for the association of low RMSD structures with high output scores, after the pairwise potentials method.
- has comparative false positive rates to the pairwise potentials method for all ranges of sensitivities.

Here it is worth mentioning that the basic NN-dist method performs the best in ROC analysis, by yielding highest levels of specificities, compared to other methods, for a wide range of sensitivities. However, in the context of blind structure prediction experiments such as CASP, the emphasis is not on getting the most number of true negatives right, but on the top few best predictions.

It is shown in this chapter that the proposed paradigm of using neural networks for decoy discrimination yields a level of performance that is not as good as the pairwise potentials method, but is nevertheless encouraging and potentially of better performance if it can be further enhanced with additional information. In the next chapter, it is hypothesized that additional evolutionary information used in the proposed neural network method can yield equal or better performance, as measured by the various benchmarks, when compared to that of the pairwise potentials method.

Chapter 3

Using Evolutionary Information in Decoy Discrimination

3.1 Introduction

Chapter 2 attempts to build a decoy discrimination method involving native and decoy distributions of pairwise residues, using the input information of identities of pairwise residues, the physical distance between them and/or the relative solvent accessibilities of both residues. It is shown that the additional input information of the relative solvent accessibility values increases the performance, in the context of the Z score, of discriminating native structures from decoy structures.

In this chapter, it is proposed that evolutionary information be included in the decoy discrimination. Evolutionary information in the form of multiple sequence alignments and derived profiles have been used in several secondary structure prediction methods successfully for the increase of the Q_3 accuracy (Section 1.2.4). Here, the idea of using evolutionary information is suggested for increasing the performance of the neural-network based decoy discrimination method.

Hence, a novel method is proposed for the inclusion of evolutionary information in the context of the neural network methodology used so far. In this method, the neural networks are trained on sequence profiles, instead of the residue identities. In Figures 2.14, 2.31 and 2.32, the neural network topologies are selected in such a way that there are 20 inputs per residue, and an additional 1 to 3 inputs depending on the feature of

interest (pairwise distance and/or relative solvent accessibilities). The 2 input vectors of size 20×1 each in Figures 2.14, 2.31 and 2.32 are for single residue identities, with only 1 out of 20 neurons switched on for each training example during neural network training. Such an input topology is deliberately selected with the eventuality of training evolutionary profiles in mind.

In this proposed method, the input vectors would take in sequence profiles of the residue positions, instead of the residue identities. These profiles are calculated from multiple sequence alignments of the original sequence. The input features of pairwise distance and/or relative solvent accessbilities are retained. This method is labelled as the ***sequence profile method***. There are 3 possible configurations of the sequence profile method, namely the topologies with the input feature of pairwise distance only, relative solvent accessibilities only, and a combination of both the pairwise distance and the relative solvent accessibilities.

Another way of using multiple sequence alignment information is to obtain the homologous sequences of the test protein, apply them to the various neural network methods, and then average the network scores obtained, in a bid to improve the benchmarking measures. This idea is not new and was used by Reva and co-workers [140] to improve the Z scores of the native structures among alternative conformations with the averaging of energies of homologous sequences in gapless threading.

In this averaging method, homologues of the target sequence are first obtained, and then these sequence homologues are threaded onto each and every structure in the decoy set, including the native structure. For example, if the *1hyp* protein of the Baker decoy dataset has 4 homologous sequences, these 4 sequences are threaded onto each of the 1400+1 decoy and native structures. The next step is to evaluate the likelihood of each decoy structure using the trained neural networks in the previous chapter. For each structure, the scores obtained for all homologous sequences, including the original sequence, are averaged to produce a mean score that describes the 'native-like' property of that particular structure. The usual Z score and enrichment can then be applied to these mean scores.

For the sake of convenience, this particular method of using multiple sequence information is referred to as the ***homologue threading*** method. The motivation of the homologue threading method is to reduce the noise of the neural-network based decoy discrimination method by applying it to more sequences, instead of just one sequence, and then averaging the scores obtained. This is done under the assumption that the close homologues adopt similar 3D folds to that of the original sequence. The previous neural networks used for the homologue threading method are the ones shown in Figures 2.14, 2.32 and 2.31, namely NN-dist, NN-solvpair and NN-solvpairndist, as previously mentioned in Table 2.11.

Table 3.1 shows a summary of the additional variants to the neural-network based decoy discrimination method developed in the previous chapter for both the homologue threading and sequence profile methods.

3.2 Materials and Methods

This section describes the procedures and methods used in both the homologue threading and sequence profile methods. The training dataset (Table D.1) and test dataset of decoys (Tables 2.5 and 2.3) remain the same, although the training dataset applies to only the sequence profile method.

The next section describes the use of PSI-BLAST [25] in deriving both the set of sequence homologues and the profiles for use in the homologue threading and sequence profile methods respectively. The following sections after that describe the algorithms peculiar to both the homologue threading and sequence profile methods.

No.	Variant Type	Previous Network Used	No. of input neurons	Name for this Variant	Training Required
1	Homologue Threading (HT)	NN-dist	41	HT-NN-dist	No
2	Homologue Threading (HT)	NN-solvpair	42	HT-NN-solvpair	No
3	Homologue Threading (HT)	NN-solvpairndist	43	HT-NN-solvpairndist	No
4	Sequence Profile (SP)	None	41	SP-NN-dist	Yes
5	Sequence Profile (SP)	None	42	SP-NN-solvpair	Yes
6	Sequence Profile (SP)	None	43	SP-NN-solvpairndist	Yes

Table 3.1: A Summary of the Methods Used for the Inclusion of Evolutionary Information for Decoy Discrimination

3.2.1 Evolutionary Information

The multiple sequence alignment of a target sequence allows homologous sequences to be aligned in such a way that provides useful information about the conserved residues in certain positions in a family of sequences. Homologous sequences are sequences that are evolutionarily related. Figure 3.1 shows an example of a multiple sequence alignment.

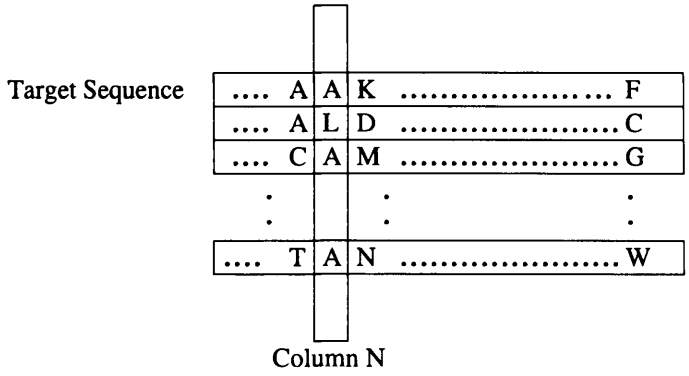


Figure 3.1: An Example of a Multiple Sequence Alignment

In this work, PSI-BLAST [25] is used to identify homologous sequences of a target protein from the sequence databases. This is done for each sequence from the training dataset (Table D.1). The homologous sequences used in the homologue threading methods are taken from the top 10 PSI-BLAST hits of each target sequence.

PSI-BLAST also produces position-specific profiles as intermediate outputs, which encode useful information about the conserved residues in each position of the target sequence. These PSI-BLAST profiles are then used for the training of the sequence profile methods. Such use of PSI-BLAST profiles have been successfully demonstrated in the PSIPRED secondary structure prediction server [15].

For neural network training in the sequence profile method, two column vectors of size 20×1 each, representing the two residue positions of sequence separation k apart, serve as inputs, along with the pairwise distance and/or relative solvent accessibility values. Each of the 20 elements of these column vectors is normalized to values between 0 and 1 according to Equation 3.1.

$$f(x) = 1/(1 + e^{-x}) \tag{3.1}$$

For each sequence in the training dataset, 3 PSI-BLAST iterations are run. The parameters used in PSI-BLAST are 0.001 for the initial and subsequent E-values, and the sequence database used is UniRef50, release 6.7. In the UniRef50 dataset, all the sequences are at most 50% similar in terms of sequence identity, and this helps to prevent homologous sequences from being used together for the generation of profiles. Although some of the sequences in the UniRef50 dataset may, possibly due to convergent evolution, still be structural homologues with one another even though the sequences are nonhomologous, such commonality has little negative effect on the multiple sequence alignments in terms of possible overrepresentation of protein families in the alignments.

The next two sections describe the homologue threading method and the sequence profile method.

3.2.2 Homologue Threading Method

The decoy datasets used are the Baker decoy set (Table 2.5) and the Decoys 'R' Us suite (Table 2.3), excluding the semfold dataset. The semfold dataset is excluded because it has about 11000 decoy structures per sequence (Table 2.4) and the threading of 10 homologues per sequence for such a large amount of decoy structures is computationally too demanding.

Figure 3.2 shows the homologue threading method.

The steps of the homologue threading method are detailed below as follows:

- For each sequence in the decoy dataset, PSI-BLAST is run for 3 iterations and the top 10 sequence homologues with the smallest E-values are threaded onto

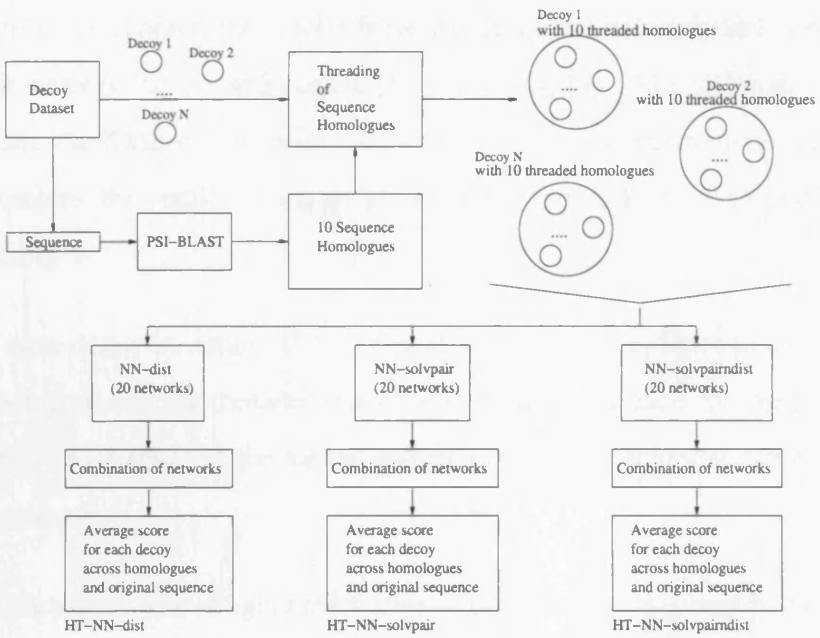


Figure 3.2: Homologue Threading Diagram

each decoy structure in the decoy dataset. The number of decoys for each dataset is shown in Tables 2.5 and 2.3 respectively.

- During threading, each homologous sequence is first aligned with the original sequence, and gaps in the alignments which do not map to the 3D structure of the original sequence are removed. This is done for each set of decoy structures of the original sequence.

- For example, the lattice_ssfit decoy dataset has 8 target proteins, and 2000 decoy structures per target protein. 4 out of 8 of these target proteins have sequence homologues. Therefore, for each of these 4 proteins, there would be a total of $10\times(2000+1) = 20010$ structures for testing, including the native structure. These structures of each of the 4 target sequences would be evaluated by the 3 types of neural networks developed in the previous chapter, namely NN-dist, NN-solvpair and NN-solvpairndist, as shown in Figure 3.2. The DSSP program is run for each threaded structure for obtaining the solvent accessibilities of the residues.

- For each type of neural network method (NN-dist, NN-solvpair, NN-solvpairndist), there are 20 networks, one for each sequence separation k ($4 \leq k \leq 22$, $k > 22$), that are to be tested on each threaded decoy structure. In

the previous chapter, the results from the 20 neural networks are combined in 3 ways, namely the S combination ($4 \leq k \leq 10$), the SM combination ($4 \leq k \leq 22$) and the SML combination ($k \geq 4$). The same combinations are used here to combine the results of the different neural networks of each threaded decoy structure.

- For each decoy structure, the scores of its original sequence and the sequence homologues that are threaded onto it are averaged for each combination (S, SM, SML). The Z score of the native structure and the enrichment can be calculated accordingly.

- The paradigms of averaging the results of the threaded sequence homologues and the results of the original sequence using the previous neural networks NN-dist, NN-solvpair and NN-solvpairndist are referred to as the HT-NN-dist, HT-NN-solvpair and HT-NN-solvpairndist methods respectively.

As mentioned earlier, the homologue threading method, with its 3 subtypes (HT-NN-dist, HT-NN-solvpair, HT-NN-solvpairndist), aims to increase the Z scores of the native structure and the enrichment measure by applying the various neural network methods (NN-dist, NN-solvpair, NN-solvpairndist) to close sequence homologues of the original sequence. It is hoped that the homologue threading method can reduce the noise inherent in the neural networks when only the original sequence is tested with the various decoy (and native) structures. In the case of original sequences without any sequence homologues identified from PSI-BLAST, the Z score and enrichment would remain the same for that sequence.

3.2.3 Sequence Profile Method

Unlike the homologue threading method, the sequence profile method does not use the previous neural network methods (such as NN-dist) as described in Chapter 3. Instead new sets of networks are trained with sequence profiles in place of residue identities, for different combinations of input features, namely pairwise distance between the residues, relative solvent accessibilities of both residues or a combination of both pair-

wise distances and relative solvent accessibilities.

Figure 3.3 shows the topology of the neural network that performs training of sequence profiles with the input feature of pairwise distance. In terms of network architecture, Figure 3.3 is identical to Figure 2.14. The difference between the two lies in the nature of the input examples, where profiles being fed into the input layer in place of residue identities for Figure 3.3. This variant of the neural-network based decoy discrimination method is referred to as SP-NN-dist. Corresponding architectures of SP-NN-solvpair (relative solvent accessibilities) and SP-NN-solvpairndist (relative solvent accessibilities and pairwise distance) are not shown because the architectures are identical to that of Figures 2.32 and 2.31 respectively. Table 3.2 shows a summary of the new sequence profile methods.

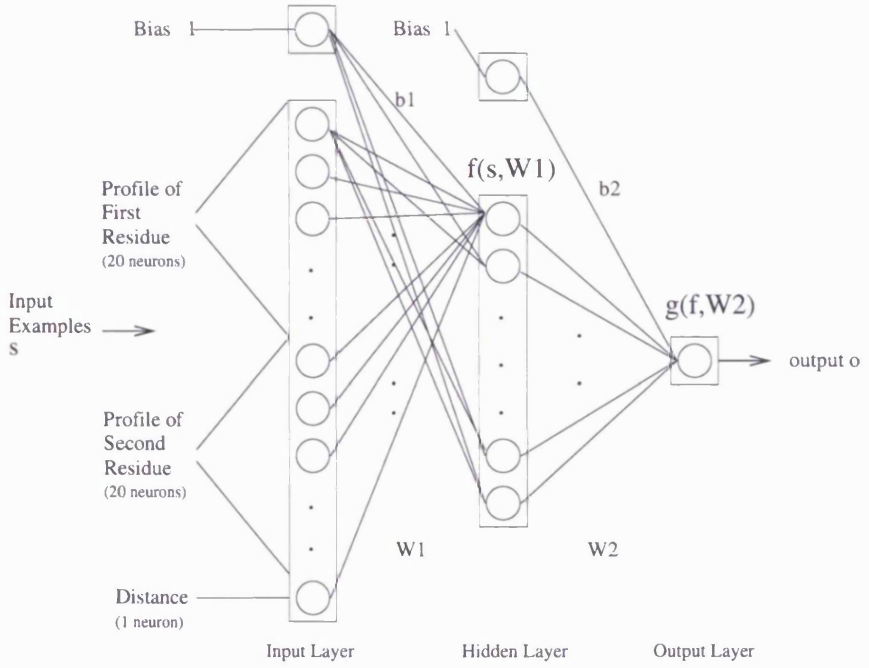


Figure 3.3: Neural Network Topology (SP-NN-dist)

The training dataset for the SP-NN-dist, SP-NN-solvpair and SP-NN-solvpairndist methods is shown in Table D.1. All of the 285 proteins in the training dataset in Table

Name	No. of input neurons	No. of networks	Description of input
SP-NN-dist	41	20	Profiles of residue pair, and distance
SP-NN-solvpair	42	20	Profiles of residue pair, and relative solvent accessibilities
SP-NN-solvpairndist	43	20	Profiles of residue pair, distance and relative solvent accessibilities

Table 3.2: A Summary of the Sequence Profile Methods

Protein	Type	Profile of Residue1	Profile of Residue2	Separation	Distance	Output Label
1a32	Native	[1.00 0.12 0.95 ... 0.12 0.02 0.05]	[0.12 0.02 0.50 ... 0.05 0.01 1.00]	4	4.765	1
1a32	Native	[0.27 0.02 0.27 ... 0.73 0.88 0.01]	[0.05 0.02 0.95 ... 0.12 0 0.02]	4	6.367	1
1a32	Native	[0.88 0.27 0.95 ... 0.05 0.02 1.00]	[0.01 0.88 0.50 ... 0.73 0.05 0.01]	4	8.894	1
...	...	...	...	...	...	..
1a32	Decoy	[1.00 0 0.50 ... 0.02 0.02 1.00]	[0.50 0.50 0.99 ... 0.01 0.73 0.27]	4	7.894	0
1a32	Decoy	[0.73 0.88 0.05 ... 0.05 0.01 1.00]	[0 0.27 0.88 ... 0.50 0.01 0.99]	4	9.664	0
1a32	Decoy	[0.73 0.73 0.12 ... 0.05 0.02 0.88]	[0.05 0.73 0.73 ... 0.27 0 0]	4	10.032	0
...	...	...	...	...	...	..

Table 3.3: Example of SP-NN-dist k=4 training input instances and their output labels

D.1 have multiple sequence alignments as identified by PSI-BLAST, and therefore the number of proteins in the training dataset for the methods trained on evolutionary information and that for the methods trained on residue identities are the same.

The creation of the negative examples for neural network training is done by reversing the sequences, as described in Section 2.3.1.2. During the reversal of the sequence when residues in the structure swap positions, the profile of each residue is swapped together with the residue. Each residue and its profile would occupy a new position in the 3D structure when the sequence is reversed. However, the relative solvent accessibilities of these swapped residues in the structure would be altered due to the difference in identities of the residues occupying the new positions, as mentioned in Section 2.5.2.

For each of the 3 methods in Table 3.2, there are 20 neural networks that are trained, one for each sequence separation k where $4 \leq k \leq 22$, and one for $k > 22$. PSI-BLAST profiles are scaled according to Equation 3.1 on page 180 before being used as inputs to the neural networks. Table 3.3 shows examples of inputs to the SP-NN-dist neural network of separation k=4. The neural network training algorithms and parameters used in the sequence profile methods are the same as that described in Section 2.3.4.1. The Matlab neural network toolbox is used for training; the MSE is used as the error of each network, the transfer function used is the radial basis function (radbas) and the gradient descent algorithm used is the Levenberg-Marquardt algorithm. The validation dataset in Table D.2 is used to prevent overfitting. All 95 proteins in the validation dataset have multiple sequence alignments as well.

It is hoped that neural networks trained with evolutionary information in the form of profiles of residue pairs, along with the usual information of pairwise distance between the residues, relative solvent accessibilities or a combination of both, can discriminate near-native structures from non near-native structures more effectively than that using residue identities.

3.2.4 Differences in the homologue threading methods and sequence profile methods

In Section 3.2.2 and 3.2.3, the homologue threading methods and sequence profile methods are introduced as possible ways of including evolutionary information in the proposed decoy discrimination methods. This section elaborates on the differences in the additional information provided by both types of methods.

The homologue threading methods seek to reduce the noise in the proposed neural network methods by averaging the network scores produced on the top 10 homologous sequences, as well as that of the original target sequence. The assumption is that the close homologues adopt similar 3D folds to that of the original sequence. The additional information lies in the extra network scores of the homologues, which can help reduce noise, through averaging, that may be present in the derivation of the network score of the original sequence.

On the other hand, the sequence profile methods are trained with PSI-BLAST profiles of residue pairs and their associated features of pairwise distance and/or relative solvent accessibilities. The profiles of residue pairs provide additional information of

- the extent of conservation of each residue in its position. For example, the neural networks can implicitly learn that an alanine residue of low relative solvent accessibility at a particular position in the sequence is usually conserved, with plausible mutations to similar residue types such as leucine.

- more importantly, the association of such extent of conservation of two residues in their respective positions with each other. For example, a pair of cysteine residues forming a disulphide bridge can be recognized by the neural networks as such when their positions are highly conserved. Another example is the contact propensities between salt bridges, which are the bonds between the positively-charged and negatively-charged residues in a protein. Pairs of PSI-BLAST profiles can effectively encode the presence of such salt bridges with high conservation scores for positively-charged residues in the first position, and high conservation scores for negatively-charged residues in the second position. More interest-

ingly, for a salt bridge, the conservation scores for negatively-charged residues in the first position can be high as well, provided the conservation scores for positively-charged residues in the second position are high too. This can be a result of mutational events in homologous sequences with the positive and negative charged residues swapping positions, while maintaining the functionality of the salt bridge. Therefore pairs of PSI-BLAST profiles, as training input into a neural network, can effectively encode such information that signifies the presence of salt bridges. Hence, the neural networks can learn, with the usage of PSI-BLAST profiles, to recognize the presence of salt bridges in native or low-RMSD decoy structures.

Such additional information might, in theory, enable sequence profile methods to perform better than the homologue threading methods.

3.3 Results

This section shows the results of both the homologue threading and sequence profile methods. The statistical tests outlined in Section 2.3.6 are also repeated and presented in this section. The results are organized as follows:

- a discussion on the number of homologues used in the homologue threading methods.
- effect of different combinations of sequence separations for the sequence profile methods.
- Z scores and enrichments for both the homologue threading methods and the sequence profile methods.
- results of the Wilcoxon sign-rank test for top model selection.
- results of the Wilcoxon sign-rank test for Spearman correlation coefficient.
- results of ROC analysis

3.3.1 Number of Homologues Used in Homologue Threading Methods

In this section, the number of homologues used for each dataset for the homologue threading methods, HT-NN-dist, HT-NN-solvpair and HT-NN-solvpairndist, are shown. Table 3.4 shows the number of homologues used in the various decoy datasets.

In Table 3.4, for each decoy dataset, the number of proteins with at least 10 homologues found from the sequence database is shown. In some decoy datasets, some of the proteins have less than 10 homologues. In such cases, all the sequence homologues are used for the homologue threading method. Three proteins in the Baker dataset, *1msi*, *1utg* and *1pgx* have less than 10 homologues with 5, 7 and 9 respectively.

For the fisa, lmds and lmds_v2 datasets, there are 1, 2 and 3 proteins respectively with less than 10 homologues.

As mentioned in Section 3.2.2, the semfold decoy dataset is omitted due to the excessive computational demands required of its approximately 11000 decoy structures for each of the 10 sequence homologues per protein.

The Z scores and enrichment measures of the various homologue threading methods (HT-NN-dist, HT-NN-solvpair, HT-NN-solvpairndist) on all the decoy datasets would be presented and discussed together with the SP-NN methods in Section 3.3.3.

The next section will first present the results of the S, SM and SML combinations of the Z scores of the SP-NN methods on only the Baker decoy dataset. Because the Baker decoy dataset has a larger number of proteins than the rest of the decoy datasets, is of better quality, and has proteins of different secondary structural categories, it is informative to see how well the different SP-NN methods perform with each type of category.

Decoy Dataset	Number of proteins			
	Total	**with no homologues**	**with ≥10 homologues**	**with <10 homologues**
bakerdecoy	22	5	14	3
4state_reduced	6	0	6	0
lattice_ssfit	8	1	7	0
fisa	4	0	3	1
fisa_casp3	4	1	3	0
lmds	10	1	7	2
lmds_v2	10	0	7	3

Table 3.4: Number of homologues produced by PSI-BLAST for the native proteins in the various decoy datasets for the homologue threading methods

3.3.2 Comparisons of Different Combinations for the Sequence Profile Methods

In this section, the results of the different combinations of the SP-NN-dist, SP-NN-solvpair and SP-NN-solvpairndist methods are presented. The line of discussion in this section is similar to that in Section 2.5.5.1. Figures 3.4, 3.5 and 3.6 show the Z scores of the different ways of combining the results on the Baker decoy dataset for the SP-NN-dist, SP-NN-solvpair and SP-NN-solvpairndist methods respectively.

It can be seen from Figure 3.4 that the S combination of the SP-NN-dist method

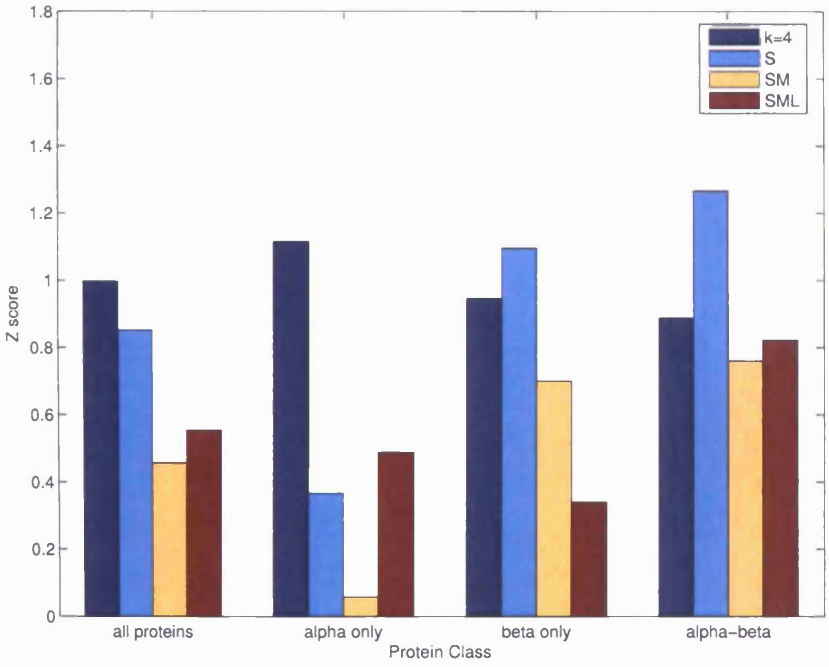


Figure 3.4: Z scores produced by the k=4, S, SM and SML combinations of sequence separations of the SP-NN-dist method on the different secondary structural classes of the Baker decoy dataset

generally has higher Z scores than the SM and SML combinations for all classes of proteins, except for the α-only class of proteins. To compare the Z scores obtained by the NN-dist method in Figure 2.26 with those obtained by the SP-NN-dist method in Figure 3.4, Figure 3.7 shows the two figures combined into a single chart.

From Figure 3.7, it can be seen that for the β-only and $\alpha\beta$ classes of proteins, the

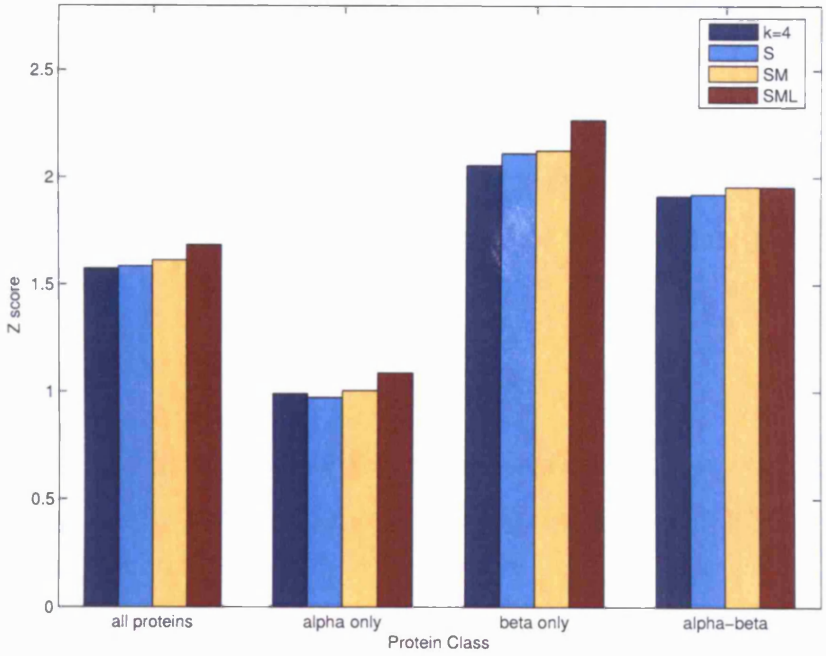


Figure 3.5: Z scores produced by the k=4, S, SM and SML combinations of sequence separations of the SP-NN-solvpair method on the different secondary structural classes of the Baker decoy dataset

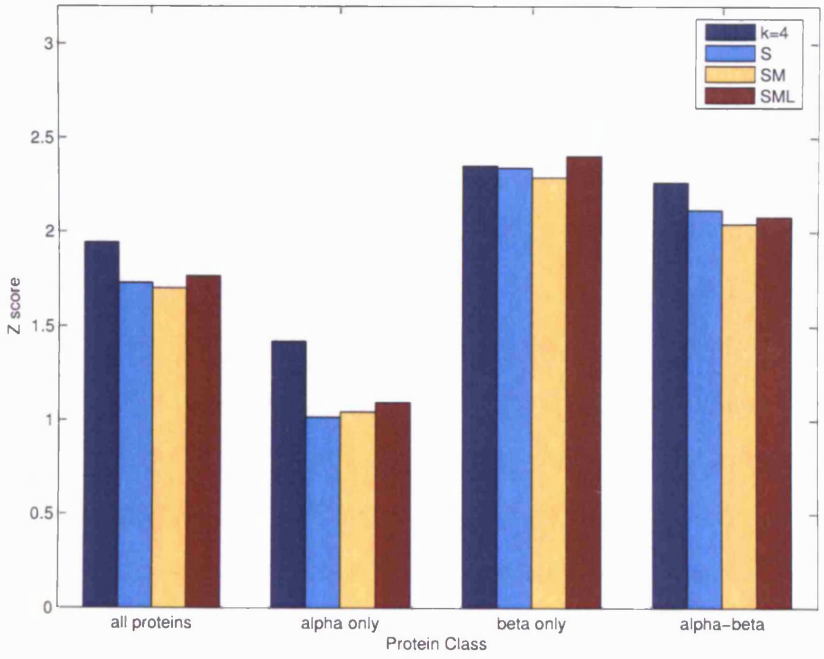


Figure 3.6: Z scores produced by the k=4, S, SM and SML combinations of sequence separations of the SP-NN-solvpairndist method on the different secondary structural classes of the Baker decoy dataset

S combination produces the highest Z-score for both the SP-NN-dist and NN-dist methods. The k=4 Z scores are highest for the combined class of proteins and the α-only class for both methods. The SM and SML combinations have universally lower Z scores than the k=4 and S combinations across all classes for both methods, apart from the SML combination of the SP-NN-dist method for the α-only class.

In contrast, the performances of the different combinations are comparable for both the SP-NN-solvpair and SP-NN-solvpairndist methods, as shown in Figures 3.5 and 3.6 respectively.

Figures 3.8 to 3.11 show a comparison of the three sequence profile methods, SP-NN-dist, SP-NN-solvpair and SP-NN-solvpairndist, over the different classes of proteins in the Baker decoy dataset over the S, SM and SML ways of network score combination. Figures 3.8 to 3.11 are essentially graphical rearrangements of the Z scores for the 3 SP-NN methods shown in Figures 3.4 to 3.6.

It can be seen from Figure 3.8 that for all proteins, the SP-NN-solvpairndist method has the highest Z score among the 3 sequence profile methods. The SP-NN-solvpair method performs marginally poorer than the SP-NN-solvpairndist method.

The best performance of the SP-NN-solvpairndist method is consistent throughout all types of secondary structural classes for all types of combinations, as shown in Figures 3.9 to 3.11. This suggests that the usage of profile information, together with the pairwise distance and relative solvent accessibility information, can help to discriminate native structures better than either of the pairwise distance or solvent accessibility features.

Figures 3.12, 3.13 and 3.14 extend the comparison of the different ways of network score combination to the Decoys 'R' Us suite of decoys for the SP-NN-dist, SP-NN-solvpair and SP-NN-solvpairndist methods respectively.

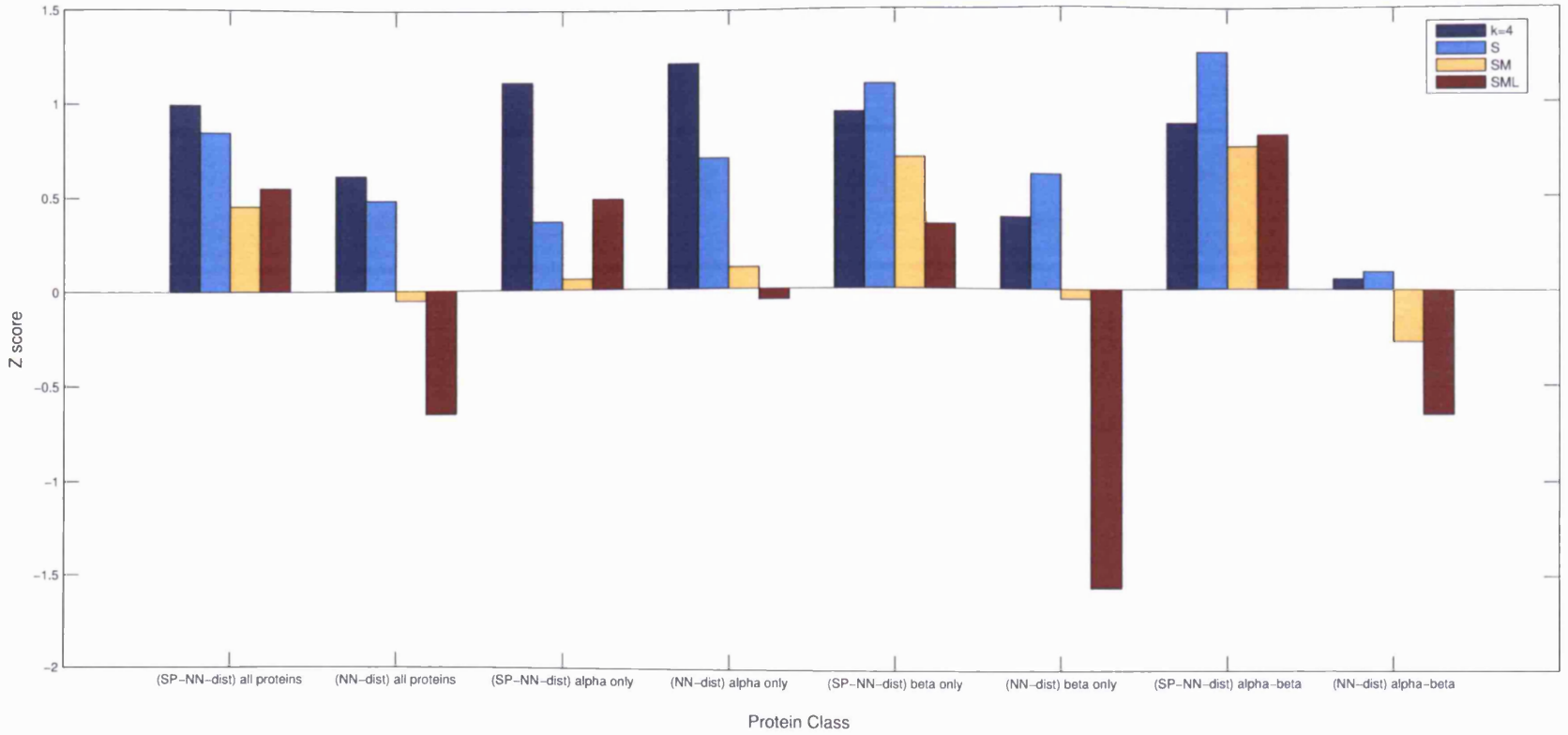


Figure 3.7: Z scores produced by the k=4, S, SM and SML combinations of sequence separations of both the SP-NN-dist and NN-dist methods on the different secondary structural classes of the Baker decoy dataset

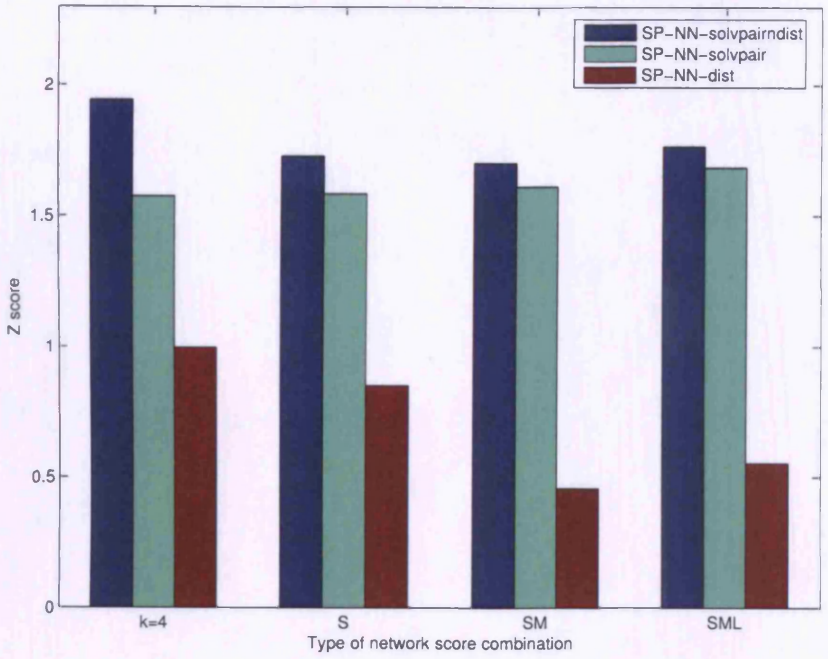


Figure 3.8: Z scores produced by the SP-NN-solvpairndist, SP-NN-solvpair and SP-NN-dist methods on all the proteins in the Baker decoy dataset across the different k=4, S, SM and SML combinations

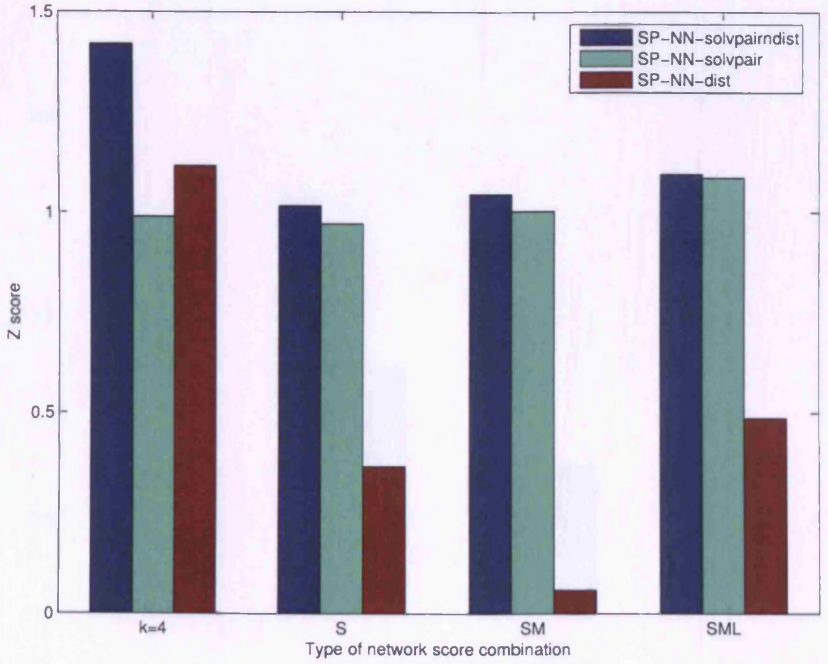


Figure 3.9: Z scores produced by the SP-NN-solvpairndist, SP-NN-solvpair and SP-NN-dist methods on α-only proteins in the Baker decoy dataset across the different k=4, S, SM and SML combinations

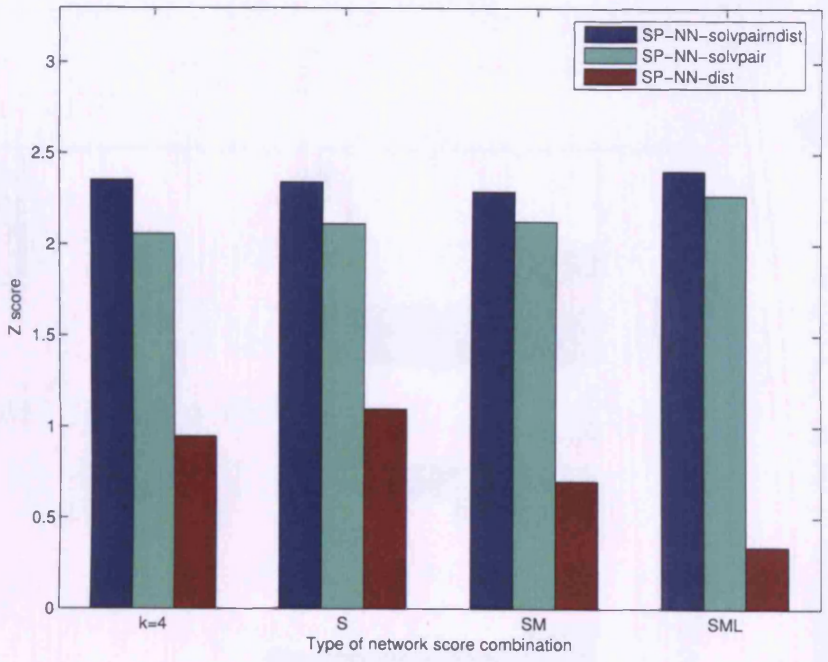


Figure 3.10: Z scores produced by the SP-NN-solvpairndist, SP-NN-solvpair and SP-NN-dist methods on β-only proteins in the Baker decoy dataset across the different k=4, S, SM and SML combinations

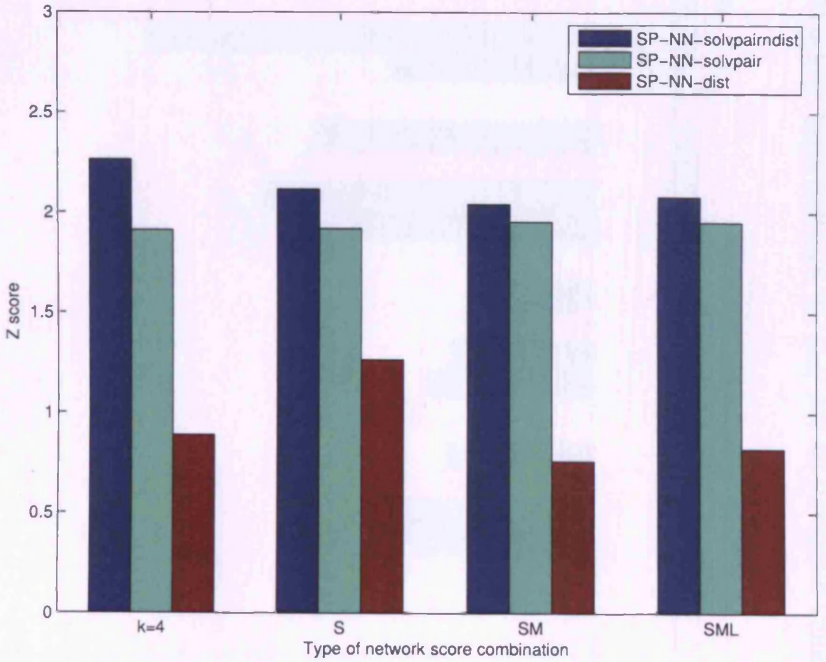


Figure 3.11: Z scores produced by the SP-NN-solvpairndist, SP-NN-solvpair and SP-NN-dist methods on $\alpha\beta$ proteins in the Baker decoy dataset across the different k=4, S, SM and SML combinations

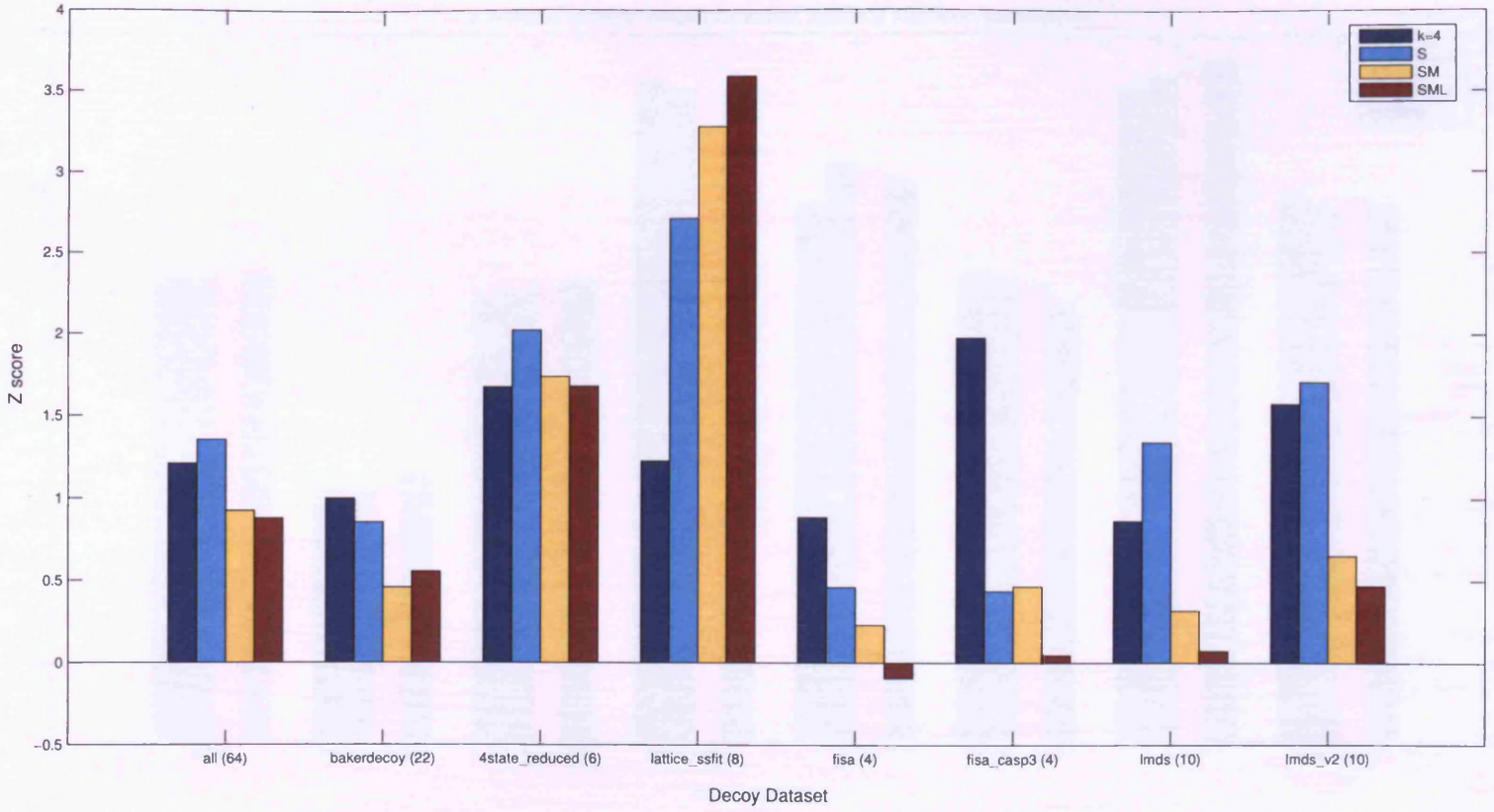


Figure 3.12: Z scores produced by the k=4, S, SM and SML combinations of sequence separations of the SP-NN-dist method on the different individual decoy datasets, including the combination of all the individual datasets

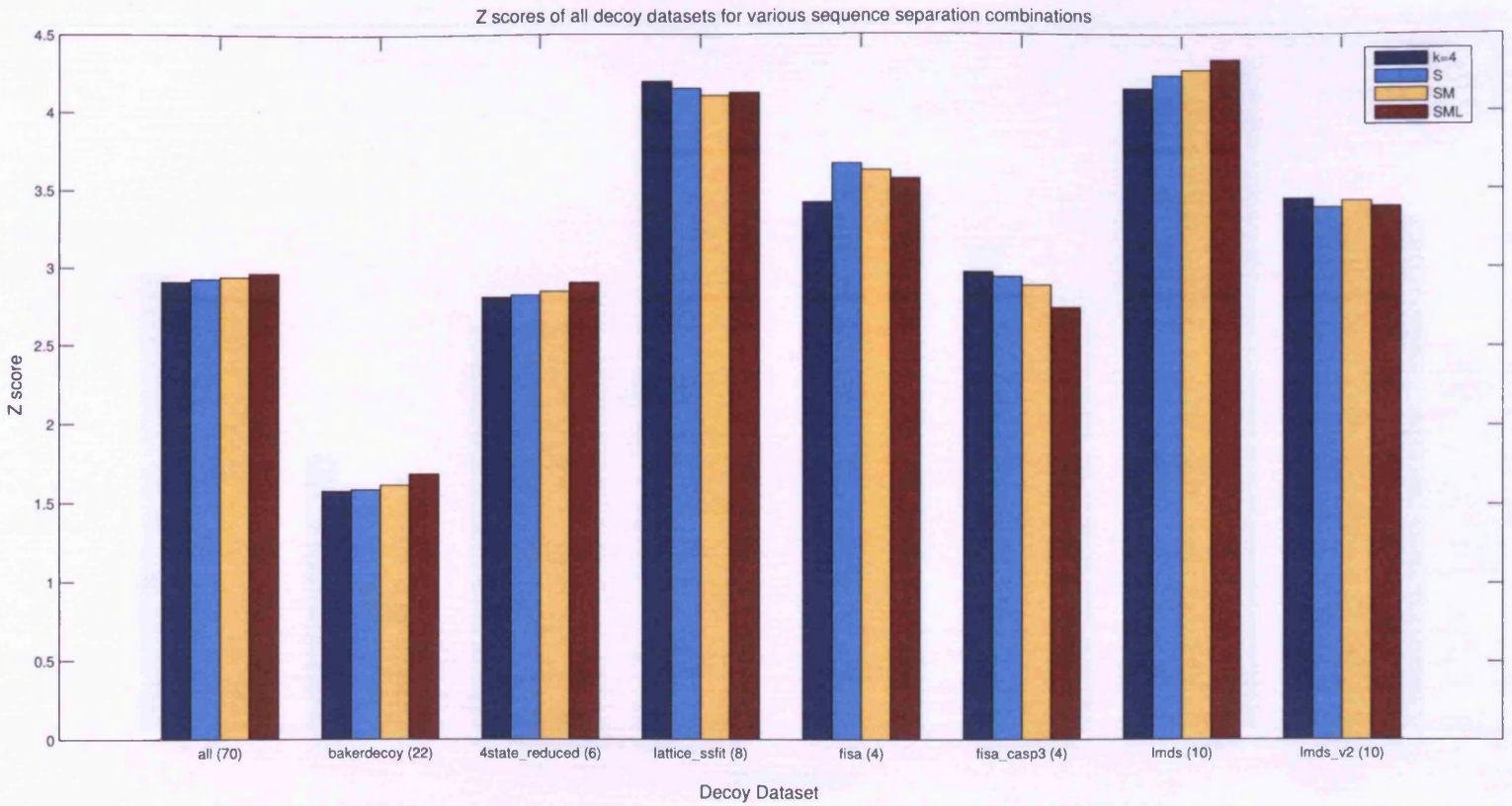


Figure 3.13: Z scores produced by the *k*=4, S, SM and SML combinations of sequence separations of the SP-NN-solvpair method on the different individual decoy datasets, including the combination of all the individual datasets

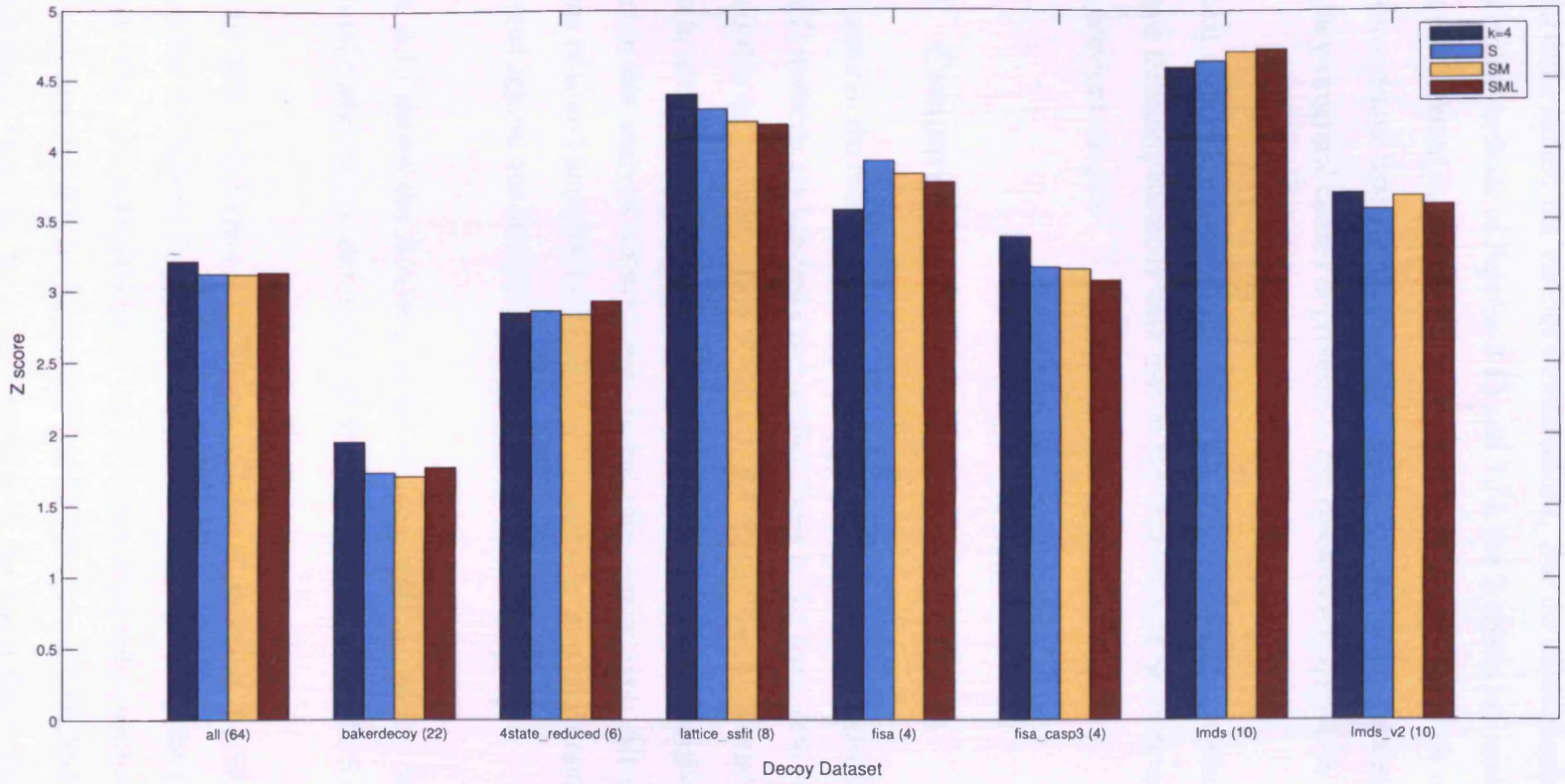


Figure 3.14: Z scores produced by the k=4, S, SM and SML combinations of sequence separations of the SP-NN-solvpairndist method on the different individual decoy datasets, including the combination of all the individual datasets

For the SP-NN-dist method in Figure 3.12, the Z scores for most of the decoy datasets differ across the various combinations. For the SP-NN-solvpair and SP-NN-solvpairndist methods in Figures 3.13 and 3.14, the Z scores are much more consistent across the different combinations for each decoy dataset. These observations are similar to those drawn from the performance of the various SP-NN methods on the different secondary structural classes of proteins in the Baker decoy dataset in Figures 3.4 to 3.6.

The next section compares the performance of the sequence profile methods and homologue threading methods with that of the basic neural network methods developed in the previous chapter.

3.3.3 Comparison of Results Across All Methods

In this section, the results of the sequence profile (SP) methods and homologue threading (HT) methods are benchmarked against those of the basic neural network methods developed in the previous chapter, namely NN-dist, NN-solvpair and NN-solvpairndist methods, and the pairwise potentials method. The K-Nearest Neighbours methods are left out in this analysis because they are not very competitive. Altogether, there are 9 variants of neural networks methods and the pairwise potentials method that are to be compared against one another.

Figure 3.15 shows the Z scores of the various methods on the different secondary structural classes of proteins in the Baker decoy dataset, using the S combination.

It can be seen from Figure 3.15 that overall, for all proteins, the SP-NN-solvpairndist method has the highest Z score among all methods, including the pairwise potentials method. The best performance of the SP-NN-solvpairndist method is also repeated for β-only proteins and $\alpha\beta$ proteins. The SP-NN-solvpair method is a close second in these cases. However, for α-only proteins in the Baker decoy dataset, the pairwise potentials method is still the best, as in the case of Figure 2.44 where only the 3 basic NN methods are being compared.

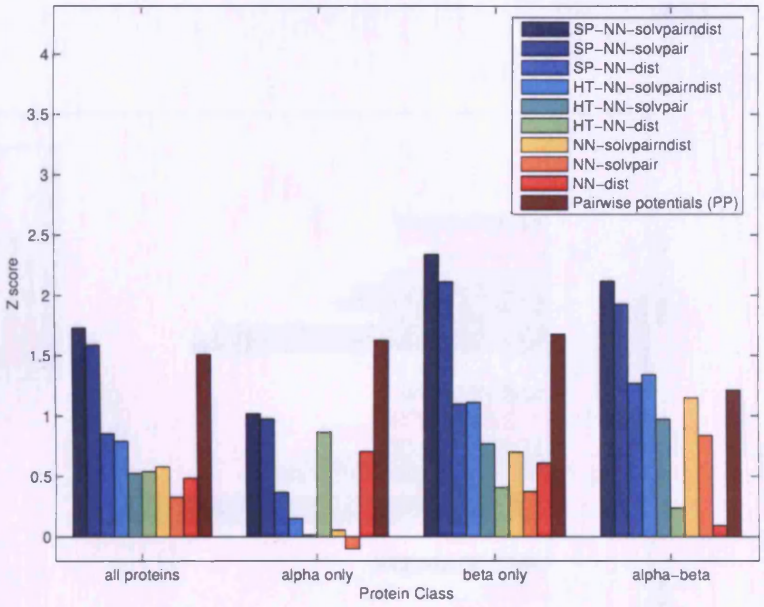


Figure 3.15: Z scores produced by the S combination of the sequence profile (SP) methods, the homologue threading (HT) methods, the basic NN-solvpairndist, NN-solvpair, NN-dist methods, and the pairwise potentials method on the different secondary structural classes of the Baker decoy dataset

For the various secondary structural classes of proteins in the Baker decoy dataset in Figure 3.15, the performance of the homologue threading methods, HT-NN-dist, HT-NN-solvpair and HT-NN-solvpairndist methods, show a modest increase in the Z score over the basic neural network counterparts, with the exception of HT-NN-dist in the β-only class. Figure 3.16 show the Z scores for the S combination of sequence separations across all decoy datasets for all 9 NN methods and the pairwise potentials method.

Figure 3.17 shows the enrichment scores of the S combination for all the methods, including the pairwise potentials method, on the decoy datasets.

It can be seen from Figure 3.16 that the SP-NN-solvpairndist method has the highest Z score compared to the rest of the methods, including the pairwise potentials method for the combined decoy dataset of 70 sets. Apart from the lattice_ssfit dataset where the pairwise potentials method has the highest Z score, the SP-NN-solvpairndist method

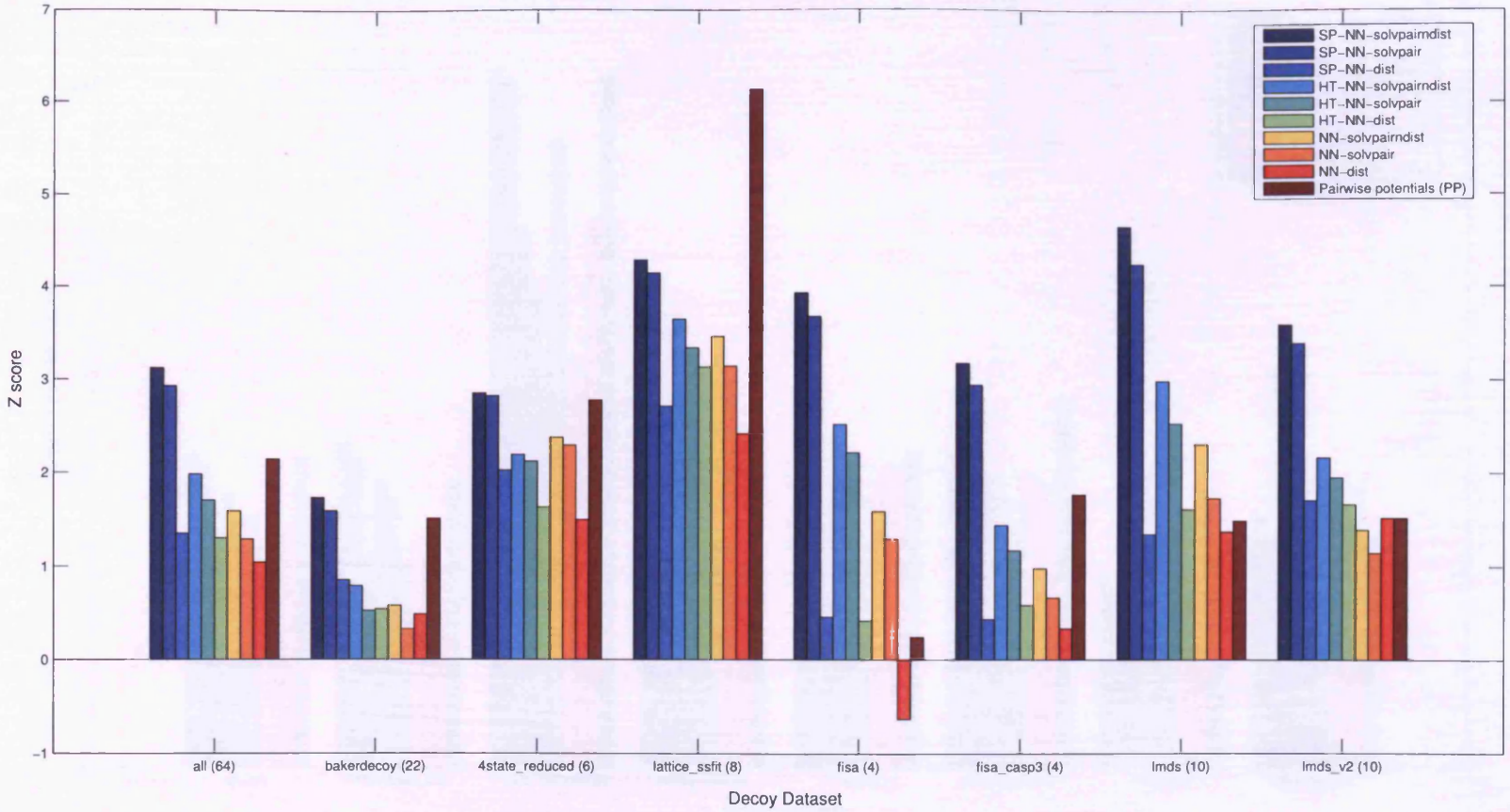


Figure 3.16: Z scores produced by the S combination of the sequence profile (SP) methods, the homologue threading (HT) methods, the basic NN-solvpairndist, NN-solvpair, NN-dist methods, and the pairwise potentials method on the different individual decoy datasets, including the combination of all the individual datasets

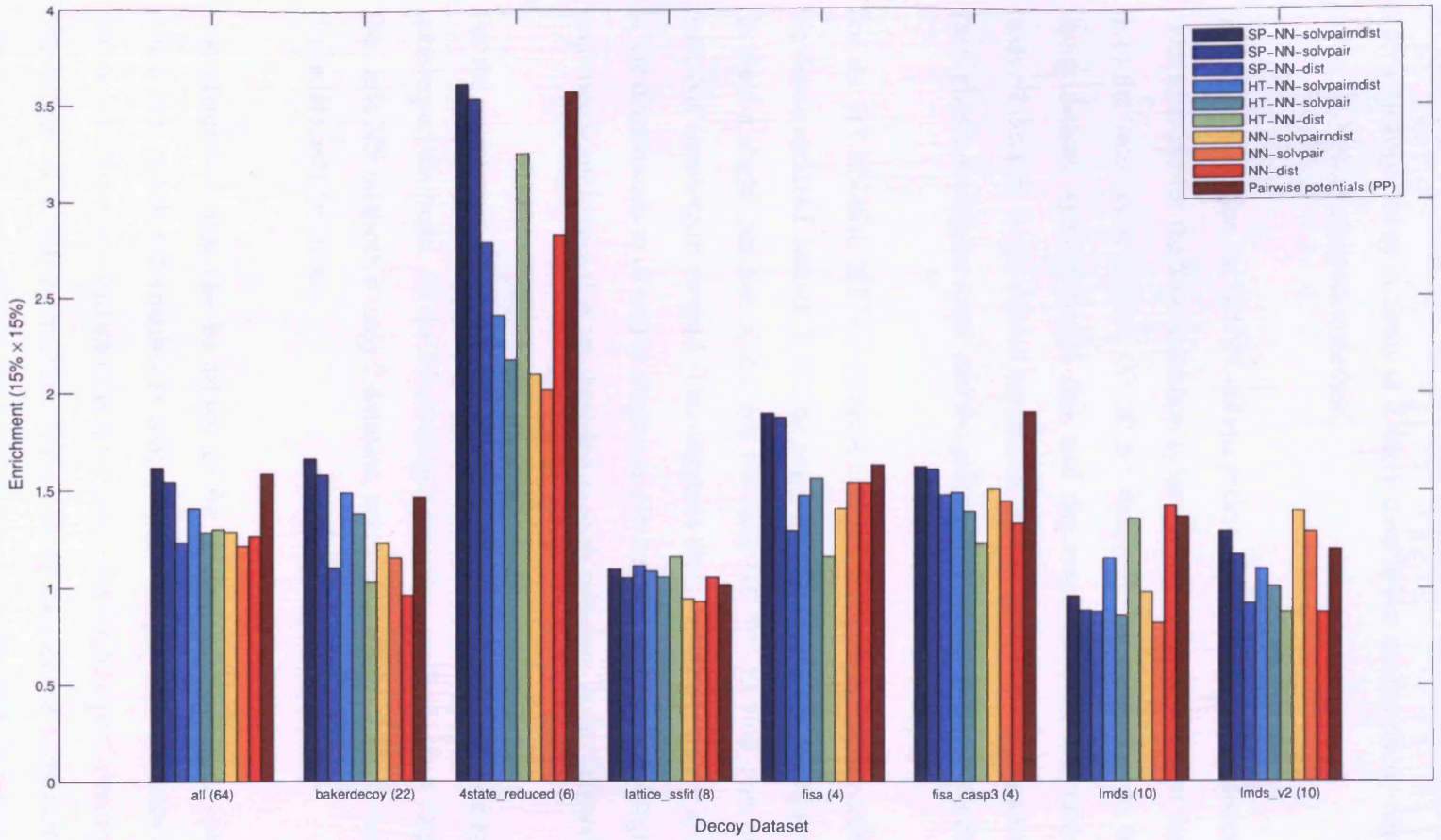


Figure 3.17: Enrichment scores produced by the S combination of the sequence profile (SP) methods, the homologue threading (HT) methods, the basic NN-solvpairndist, NN-solvpair, NN-dist methods, and the pairwise potentials method on the different individual decoy datasets, including the combination of all the individual datasets

has the highest Z scores for all the other decoy datasets. Having said that, it can be seen from Table 2.6 that the lattice_ssfit dataset consists of large numbers of high RMSD decoys. This suggests that the pairwise potentials method still has a slightly higher discriminatory power in terms of Z score over lower quality decoy datasets, compared to the SP-NN-solvpairndist method.

In Figure 3.17, for the SP-NN-solvpairndist method, the performance on the enrichment measure for the S combination is less pronounced than that of the Z score. Apart from the lmds decoy dataset, the SP-NN-solvpairndist method ranks best in the Baker decoy dataset, 4state_reduced, fisa, and fisa_casp3 datasets, and ranks second in the lmds_v2 dataset. In the overall combined dataset, the SP-NN-solvpairndist method has the highest enrichment score, and the pairwise potentials method is a close second best.

For the HT-NN-dist, HT-NN-solvpair and HT-NN-solvpairndist methods, apart from the 4state_reduced dataset, it can be seen that the averaging of sequence homologues do yield a slight increase of Z score for each HT-NN method over its corresponding basic NN counterpart method. This suggests that a modest increase in the performance of the discrimination of native structures can be achieved using averaging the scores of sequence homologues that are threaded to each structure in the decoy dataset.

For the enrichment score, the HT-NN methods show little improvement over the basic counterpart methods. All the 3 homologue threading methods show improvements over the basic NN methods in only 3 datasets, namely the Baker dataset, 4state_reduced and the lattice_ssfit datasets.

One conclusion that can be drawn so far is that the SP-NN-solvpairndist method, which uses profile information in conjunction with pairwise distance and relative solvent accessibility information of residue pairs, has the best performance in terms of the discrimination of native structures for all decoy datasets (Z score) among the various neural network methods and the pairwise potentials method. In terms of selecting the low RMSD decoys (enrichment score), it slightly outperforms the rest of the methods for a number of decoy datasets.

3.3.4 Results of Wilcoxon sign-rank tests for top model selection

In this section, the Wilcoxon sign-rank tests are performed for each of the homologue threading (HT) and sequence profile (SP) methods against the pairwise potentials method, as well as the MODCHECK MQAP method. As in Section 2.3.6.1, three different structural similarity measures are used. The significance level used is 5%. The network scores averaged by the HT-NN-dist, HT-NN-solvpair and HT-NN-solvpairndist methods are of the S combination.

For a given decoy discrimination method and a given structural similarity measure, the null hypothesis states that the median of the distribution of the differences between the structural similarity scores of the top ranked model produced by this particular method and the top ranked model produced by the pairwise potentials method is zero.

For the homologue threading methods, the semfold dataset is not applicable for the analysis. Hence, the combined datasets of all proteins, α-only proteins, β-only proteins, and $\alpha\beta$ proteins do not contain semfold decoy sets. The methods that are compared against the homologue threading methods therefore have their analysis repeated without the decoy sets that belong to the semfold dataset, for the sake of effective comparison.

The following subsections separate the discussion of the P-values obtained for the homologue threading methods from those obtained for the sequence profile methods.

3.3.4.1 P-values of the top model selection test for the Homologue Threading Methods

Tables 3.5, 3.6 and 3.7 show the P-values obtained from the one-tailed Wilcoxon sign-rank test for the comparison of the HT-NN-dist, HT-NN-solvpair and HT-NN-solvpairndist methods, with the pairwise potentials method and MODCHECK, with the structural similarity measures defined as TM-score, GDT-TS and MaxSub respec-

tively.

It can be seen from Tables 3.5 to 3.7 that all the P-values are ≥ 0.05, except for

- HT-NN-dist and pairwise potentials, lmds dataset, GDT-TS, P-value = 0.0137

Relaxing the significance level to 10%, the following cases are observed:

- HT-NN-dist and pairwise potentials, lmds dataset, TM-score, P-value = 0.0527
- HT-NN-dist and MODCHECK, lmds dataset, TM-score, P-value = 0.0801
- HT-NN-dist and MODCHECK, lmds dataset, GDT-TS, P-value = 0.0967
- HT-NN-dist and MODCHECK, lmds dataset, MaxSub, P-value = 0.0801

It can be seen that all cases are observed with the HT-NN-dist method for the lmds dataset.

Table 3.8 shows the P-values of the one-tailed Wilcoxon test for the comparison of the HT-NN-dist, HT-NN-solvpair and HT-NN-solvpairndist methods with the corresponding basic NN counterparts.

At 5% significance level, there are no instances in Table 3.8 where the null hypothesis can be rejected. At 10% significance level, the null hypothesis can be rejected for the following cases:

- HT-NN-solvpairndist and NN-solvpairndist, lmds_v2 dataset, GDT-TS, P-value = 0.0781

Decoy Dataset	HT-NN-dist		HT-NN-solvpair		HT-NN-solvpairndist	
	PP	**MODCHECK**	**PP**	**MODCHECK**	**PP**	**MODCHECK**
4state_reduced	0.8438	0.8906	0.9688	0.7812	0.9219	0.8438
bakerdecoys	0.6811	0.3666	0.3851	0.4419	0.3189	0.2325
fisa_casp3	0.9375	0.9375	0.9375	0.6875	0.9375	0.6875
fisa	0.9375	0.6875	0.8125	0.5000	0.9375	0.5000
lattice_ssfit	0.9961	0.9258	0.9609	0.8125	0.9609	0.8516
lmds	**0.0527**	**0.0801**	0.9814	0.9473	0.8623	0.8389
lmds_v2	0.2852	0.7871	0.6328	0.7148	0.5000	0.7148
all (less semfold)	0.9715	0.9340	0.9941	0.9465	0.9813	0.8776
$\alpha\beta$ (less semfold)	0.7069	0.8371	0.6792	0.8495	0.3208	0.4794
α-only (less semfold)	0.9918	0.8306	0.9992	0.9211	0.9994	0.2463
β-only (less semfold)	0.5781	0.9453	0.3711	0.4219	0.4219	0.8086

Table 3.5: Top Model Selection : P-values of one-tailed Wilcoxon sign-rank test between the HT-NN-dist, HT-NN-solvpair, HT-NN-solvpairndist methods and Pairwise Potentials and MODCHECK, with TM-score as the structural similarity measure

Decoy Dataset	HT-NN-dist		HT-NN-solvpair		HT-NN-solvpairndist	
	PP	MODCHECK	PP	MODCHECK	PP	MODCHECK
4state_reduced	0.8438	0.8438	0.9688	0.7812	0.9531	0.8438
bakerdecoys	0.3306	0.6456	0.2325	0.7260	0.3544	0.6926
fisa_casp3	0.8750	0.6875	0.9375	0.6875	0.9375	0.4375
fisa	0.9375	0.8125	0.8750	0.5000	0.9375	0.5000
lattice_ssfit	0.9961	0.9609	0.8438	0.5000	0.8750	0.5938
lmds	**0.0137**	**0.0967**	0.9756	0.9814	0.8838	0.8838
lmds_v2	0.1016	0.5898	0.1504	0.6328	0.6328	0.8203
all (less semfold)	0.7930	0.8379	0.9788	0.9649	0.9770	0.9544
$\alpha\beta$ (less semfold)	0.5283	0.9262	0.5000	0.8103	0.2346	0.5000
α-only (less semfold)	0.9207	0.3163	0.9982	0.9129	0.9993	0.8467
β-only (less semfold)	0.5273	0.9453	0.4219	0.6797	0.5000	0.8438

Table 3.6: Top Model Selection : P-values of one-tailed Wilcoxon sign-rank test between the HT-NN-dist, HT-NN-solvpair, HT-NN-solvpairndist methods and Pairwise Potentials and MODCHECK, with GDT-TS as the structural similarity measure

Decoy Dataset	HT-NN-dist		HT-NN-solvpair		HT-NN-solvpairndist	
	PP	MODCHECK	PP	MODCHECK	PP	MODCHECK
4state_reduced	0.8438	0.8438	0.9688	0.7812	0.9688	0.8438
bakerdecoys	0.5553	0.4419	0.2633	0.4419	0.3306	0.3131
fisa_casp3	0.9375	0.9375	0.8750	0.6875	0.8750	0.5625
fisa	0.9375	0.8750	0.8125	0.3750	0.6875	0.5000
lattice_ssfit	0.9805	0.8008	0.9023	0.5000	0.9023	0.4688
lmds	0.2158	**0.0801**	0.9902	0.9756	0.9199	0.8125
lmds_v2	0.3262	0.5449	0.4102	0.5898	0.6738	0.8203
all (less semfold)	0.9742	0.8998	0.9903	0.9080	0.9843	0.1479
$\alpha\beta$ (less semfold)	0.4906	0.3880	0.3782	0.1629	0.2190	0.4794
α-only (less semfold)	0.9982	0.9275	0.9978	0.8621	0.9996	0.2059
β-only (less semfold)	0.4219	0.9023	0.2734	0.4219	0.3594	0.8086

Table 3.7: Top Model Selection : P-values of one-tailed Wilcoxon sign-rank test between the HT-NN-dist, HT-NN-solvpair, HT-NN-solvpairndist methods and Pairwise Potentials and MODCHECK, with MaxSub as the structural similarity measure

Decoy Dataset	HT-NN-dist			HT-NN-solvpair			HT-NN-solvpairndist		
	NN-dist			NN-solvpair			NN-solvpairndist		
	TM-score	GDT-TS	MaxSub	TM-score	GDT-TS	MaxSub	TM-score	GDT-TS	MaxSub
4state_reduced	0.1250	0.1250	0.3750	0.1562	0.1562	**0.0938**	0.3125	0.1875	0.1250
bakerdecoys	0.1219	0.2939	0.3424	0.4020	0.3046	0.5000	0.1388	0.1276	0.3177
fisa_casp3	0.2500	0.5000	0.2500	0.7500	0.7500	0.5000	0.5000	0.5000	0.5000
fisa	0.5000	0.5000	0.7500	0.3750	0.5000	0.3750	0.5000	0.4375	0.3125
lattice_ssfit	0.7109	0.8906	0.7188	0.4062	0.6562	0.4062	0.5938	0.8125	0.5000
lmds	0.2188	0.1562	0.1562	0.2812	0.5000	0.1562	0.7109	0.4688	0.8516
lmds_v2	0.1094	0.1094	0.2188	0.2344	0.2891	0.4219	0.9219	**0.0781**	0.1094
all (less semfold)	0.2742	0.3532	0.4054	0.4173	0.4506	0.4397	0.4217	0.4708	0.4626
$\alpha\beta$ (less semfold)	0.4229	0.2783	0.3672	0.1602	0.1602	0.1826	0.4492	0.4829	0.3823
α-only (less semfold)	0.4529	0.3096	0.2770	0.4840	0.4361	0.3437	0.4203	0.2730	0.3096
β-only (less semfold)	0.3125	0.3125	0.2188	0.2344	**0.0547**	0.5938	0.1484	**0.0547**	0.7188

Table 3.8: Top Model Selection : P-values of one-tailed Wilcoxon sign-rank test between the HT-NN-dist, HT-NN-solvpair, HT-NN-solvpairndist methods and the corresponding basic NN methods

- HT-NN-solvpairndist and NN-solvpairndist, β-only dataset, GDT-TS, P-value = 0.0547
- HT-NN-solvpair and NN-solvpair, 4state_reduced dataset, MaxSub, P-value = 0.0938
- HT-NN-solvpair and NN-solvpair, β-only dataset, GDT-TS, P-value = 0.0547

The HT-NN-solvpairndist and HT-NN-solvpair methods show improvement in top model selection for some cases over the NN-solvpairndist and NN-solvpair methods respectively. For the GDT-TS structural similarity measure, the improvement can be noticed in the β-only dataset. The HT-NN-dist method shows no instances of improvement in top model selection over the corresponding NN-dist method, even at 10% significance level.

The next section shows the P-values from the one-tailed Wilcoxon sign-rank test for the sequence profile methods.

3.3.4.2 P-values of the top model selection test for the Sequence Profile Methods

Tables 3.9, 3.10 and 3.11 show the P-values obtained from the one-tailed Wilcoxon sign-rank test for the comparison of the SP-NN-dist, SP-NN-solvpair and SP-NN-solvpairndist methods, with the pairwise potentials method and MODCHECK, with the structural similarity measures defined as TM-score, GDT-TS and MaxSub respectively.

It can be seen from Tables 3.9 to 3.11 that all the P-values are ≥ 0.05. Relaxing the significance level to 10%, the following cases are observed:

- SP-NN-solvpairndist and pairwise potentials, Baker dataset, TM-score, P-value = 0.0838
- SP-NN-solvpairndist and pairwise potentials, Baker dataset, MaxSub, P-value = 0.0789

- SP-NN-solvpair and pairwise potentials, Baker dataset, MaxSub, P-value = 0.0743
- SP-NN-solvpair and MODCHECK, lattice_ssfit dataset, MaxSub, P-value = 0.0781
- SP-NN-solvpairndist and MODCHECK, lattice_ssfit dataset, MaxSub, P-value = 0.0781

For the Baker decoy dataset, the null hypothesis can be rejected for the TM-score and MaxSub structural similarity scores when comparing the SP-NN-solvpairndist method with the pairwise potentials method. This means that, at a 10% significance level, the SP-NN-solvpairndist method can select better top models (TM-score, MaxSub) than the pairwise potentials method for the Baker decoy dataset.

Table 3.12 shows the P-values of the one-tailed Wilcoxon test for the comparison of the SP-NN-dist, SP-NN-solvpair and SP-NN-solvpairndist methods with the corresponding basic NN counterparts.

From Table 3.12, at a 5% significance level, the null hypothesis can be rejected for the following:

- SP-NN-solvpairndist and NN-solvpairndist, Baker dataset, TM-score, P-value = 0.0366
- SP-NN-solvpairndist and NN-solvpairndist, Baker dataset, GDT-TS, P-value = 0.0337
- SP-NN-solvpair and NN-solvpair, Baker dataset, TM-score, P-value = 0.0430
- SP-NN-solvpair and NN-solvpair, Baker dataset, GDT-TS, P-value = 0.0430

Decoy Dataset	SP-NN-dist		SP-NN-solvpair		SP-NN-solvpairndist	
	PP	MODCHECK	PP	MODCHECK	PP	MODCHECK
4state_reduced	0.5000	0.4219	0.5781	0.5000	0.5781	0.5000
bakerdecoys	0.9423	0.9385	0.1057	0.1457	**0.0838**	0.1246
fisa_casp3	0.5000	0.5000	0.5625	0.5000	0.5625	0.5000
fisa	0.9375	0.6875	0.6875	0.5000	0.9375	0.8125
lattice_ssfit	0.9805	0.7695	0.6289	0.2344	0.6797	0.2344
lmds	0.9971	0.8838	0.9805	0.8496	0.8838	0.7217
lmds_v2	0.5449	0.8389	0.6328	0.8750	0.8125	0.9346
semfold	0.9688	0.9844	0.7812	0.8438	0.9844	0.9844
all	0.9998	0.9912	0.7512	0.6289	0.8612	0.7171
$\alpha\beta$	0.9902	0.9955	0.7777	0.9263	0.9164	0.9690
α-only	0.9991	0.9318	0.8111	0.5043	0.8364	0.5043
β-only	0.7266	0.9727	0.3203	0.6797	0.3203	0.6797

Table 3.9: Top Model Selection : P-values of one-tailed Wilcoxon sign-rank test between the SP-NN-dist, SP-NN-solvpair, SP-NN-solvpairndist methods and Pairwise Potentials and MODCHECK, with TM-score as the structural similarity measure

Decoy Dataset	SP-NN-dist		SP-NN-solvpair		SP-NN-solvpairndist	
	PP	MODCHECK	PP	MODCHECK	PP	MODCHECK
4state_reduced	0.5000	0.4219	0.6562	0.5000	0.6562	0.5000
bakerdecoys	0.9211	0.9690	0.1313	0.3189	0.1057	0.2529
fisa_casp3	0.6250	0.5000	0.5625	0.3125	0.5625	0.3125
fisa	0.9375	0.8125	0.6875	0.5000	0.9375	0.5625
lattice_ssfit	0.9453	0.7695	0.5000	0.1094	0.5781	0.2891
lmds	0.9932	0.8838	0.9629	0.8984	0.8838	0.7217
lmds_v2	0.1016	0.6152	0.4551	0.8203	0.7217	0.9033
semfold	0.9219	0.9219	0.7188	0.8438	0.9844	0.9688
all	0.9979	0.9824	0.6442	0.7009	0.7861	0.7551
$\alpha\beta$	0.9779	0.9931	0.7895	0.9505	0.9498	0.9781
α-only	0.9987	0.9194	0.8550	0.4612	0.8680	0.4102
β-only	0.7266	0.9609	0.2852	0.8086	0.2852	0.8086

Table 3.10: Top Model Selection : P-values of one-tailed Wilcoxon sign-rank test between the SP-NN-dist, SP-NN-solvpair, SP-NN-solvpairndist methods and Pairwise Potentials and MODCHECK, with GDT-TS as the structural similarity measure

Decoy Dataset	SP-NN-dist		SP-NN-solvpair		SP-NN-solvpairndist	
	PP	MODCHECK	PP	MODCHECK	PP	MODCHECK
4state_reduced	0.5000	0.4219	0.7188	0.5000	0.7188	0.5000
bakerdecoys	0.8543	0.8304	**0.0743**	0.1445	**0.0789**	0.1293
fisa_casp3	0.6250	0.6875	0.5000	0.5000	0.5000	0.5000
fisa	0.9375	0.8750	0.6875	0.5625	0.8125	0.6875
lattice_ssfit	0.9609	0.5000	0.4727	**0.0781**	0.5273	**0.0781**
lmds	0.9902	0.8838	0.9727	0.8203	0.9199	0.6523
lmds_v2	0.4102	0.8203	0.4551	0.8203	0.7539	0.9033
semfold	0.8906	0.9688	0.5000	0.5781	0.7188	0.8438
all	0.9991	0.9570	0.3661	0.4441	0.7267	0.5237
$\alpha\beta$	0.8784	0.9046	0.5160	0.8010	0.8341	0.8521
α-only	0.9996	0.9687	0.7474	0.3663	0.7217	0.3326
β-only	0.5781	0.9023	0.1562	0.6289	0.1562	0.6289

Table 3.11: Top Model Selection : P-values of one-tailed Wilcoxon sign-rank test between the SP-NN-dist, SP-NN-solvpair, SP-NN-solvpairndist methods and Pairwise Potentials and MODCHECK, with MaxSub as the structural similarity measure

Decoy Dataset	SP-NN-dist			SP-NN-solvpair			SP-NN-solvpairndist		
	NN-dist			NN-solvpair			NN-solvpairndist		
	TM-score	GDT-TS	MaxSub	TM-score	GDT-TS	MaxSub	TM-score	GDT-TS	MaxSub
4state_reduced	0.1562	0.1562	0.1562	0.0938	0.0938	0.0938	0.1562	0.1562	0.1562
bakerdecoys	0.8474	0.8743	0.6553	**0.0430**	**0.0430**	0.0727	**0.0366**	**0.0337**	0.0502
fisa_casp3	0.0625	0.0625	0.0625	0.5625	0.5625	0.3125	0.5000	0.5000	0.5000
fisa	0.5000	0.6250	0.6250	0.5000	0.5000	0.5000	0.6250	0.5000	0.6250
lattice_ssfit	0.3711	0.7266	0.5000	0.1875	0.5312	0.0781	0.1914	0.7266	0.1562
lmds	0.9863	0.9941	0.9512	0.7695	0.3711	0.6289	0.7871	0.2852	0.8496
lmds_v2	0.2158	0.2783	0.5449	0.2891	0.3438	0.4219	0.9180	0.8984	0.8203
semfold	0.8906	0.9219	0.9219	0.3125	0.3125	0.1875	0.9375	0.5000	0.8438
all	0.7256	0.8031	0.6942	0.0651	**0.0258**	**0.0370**	0.2048	0.0926	0.1966
$\alpha\beta$	0.2877	0.5740	0.8428	0.1906	0.8863	0.2507	0.9781	0.9886	0.9560
α-only	0.8408	0.8895	0.7858	0.0614	**0.0185**	**0.0432**	**0.0438**	**0.0115**	**0.0490**
β-only	0.3438	0.6562	0.0781	0.1562	0.1562	0.6289	0.1562	0.1562	0.6289

Table 3.12: Top Model Selection : P-values of one-tailed Wilcoxon sign-rank test between the SP-NN-dist, SP-NN-solvpair, SP-NN-solvpairndist methods and the corresponding basic NN methods

- SP-NN-solvpair and NN-solvpair, all proteins, GDT-TS, P-value = 0.0258
- SP-NN-solvpair and NN-solvpair, all proteins, MaxSub, P-value = 0.0370
- SP-NN-solvpairndist and NN-solvpairndist, α-only proteins, TM-score, P-value = 0.0438
- SP-NN-solvpairndist and NN-solvpairndist, α-only proteins, GDT-TS, P-value = 0.0115
- SP-NN-solvpairndist and NN-solvpairndist, α-only proteins, MaxSub, P-value = 0.0490
- SP-NN-solvpair and NN-solvpair, α-only proteins, GDT-TS, P-value = 0.0185
- SP-NN-solvpair and NN-solvpair, α-only proteins, MaxSub, P-value = 0.0432

The additional evolutionary information in the SP-NN-solvpairndist and SP-NN-solvpair methods seems to help select top ranked models of a higher quality for the Baker decoy dataset, as well as α-only proteins. The SP-NN-dist method shows no improvement in top model selection over the NN-dist method at a 5% significance level.

3.3.5 Results of Wilcoxon Sign-Rank Tests on Spearman correlation coefficients

In this section, the results of the one-tailed Wilcoxon sign-rank test on the matched pairs of Spearman correlation coefficients produced by the homologue threading methods and the sequence profile methods are presented.

As described in Section 2.3.6.2, the null hypothesis is that the median is zero for the distribution of the differences in the Spearman correlation coefficients produced by the proposed neural network decoy discrimination method and the Spearman correlation coefficients produced by the pairwise potentials method (or MODCHECK). The Spearman rank correlation coefficients are calculated between the structural similarity

scores (TM-score, GDT-TS or MaxSub) of the decoys and the output scores of the decoys assigned by a decoy discrimination method.

The network output scores produced by the SP-NN-dist, SP-NN-solvpair and SP-NN-solvpairndist methods, as well as those averaged by the homologue threading methods, are of the S combination.

For the homologue threading methods, the semfold dataset is not applicable for the analysis. Hence, the combined datasets of all proteins, α-only proteins, β-only proteins, and $\alpha\beta$ proteins do not contain semfold decoy sets. The methods that are compared against the homologue threading methods therefore have their analysis repeated without the decoy sets that belong to the semfold dataset, for the sake of effective comparison.

The following subsections separate the discussion of the P-values obtained for the homologue threading methods from those obtained for the sequence profile methods.

3.3.5.1 P-values of the Spearman correlation coefficients for the Homologue Threading Methods

Tables 3.13, 3.14 and 3.15 show the P-values obtained from the one-tailed Wilcoxon sign-rank test for the comparison of the HT-NN-dist, HT-NN-solvpair and HT-NN-solvpairndist methods, with the pairwise potentials method and MODCHECK, with the Spearman correlation coefficients of the output scores of the method, and the structural similarity measures, which are the TM-score, GDT-TS and MaxSub respectively.

It can be seen from Tables 3.13 to 3.15 that all the P-values are ≥ 0.05. Therefore the various null hypotheses are not rejected. This implies that there is no improvement in model ranking for the homologue threading methods, when compared to the pairwise potentials methods or the MODCHECK method.

Decoy Dataset	HT-NN-dist		HT-NN-solvpair		HT-NN-solvpairndist	
	PP	MODCHECK	PP	MODCHECK	PP	MODCHECK
4state_reduced	0.5000	0.1562	0.9844	0.9844	0.9844	0.9844
bakerdecoys	0.9690	0.9752	0.5709	0.5452	0.1774	0.1313
fisa_casp3	0.9375	0.9375	0.8125	0.8125	0.8125	0.8750
fisa	0.8750	0.8750	0.8750	0.8750	0.8750	0.8750
lattice_ssfit	0.9258	0.9453	0.6289	0.8086	0.5781	0.8086
lmds	0.8389	0.5000	0.9990	0.9951	0.8838	0.9199
lmds_v2	0.9033	0.8125	0.5000	0.6523	0.5000	0.7217
all (less semfold)	0.9999	0.9976	0.9974	0.9986	0.9742	0.9661
$\alpha\beta$ (less semfold)	0.9968	0.9916	0.8952	0.9187	0.6563	0.6735
α-only (less semfold)	0.9964	0.9392	0.9935	0.9762	0.9916	0.9691
β-only (less semfold)	0.9609	0.9961	0.5781	0.9258	0.5273	0.4219

Table 3.13: Spearman correlation coefficient : P-values of one-tailed Wilcoxon sign-rank test between the HT-NN-dist, HT-NN-solvpair, HT-NN-solvpairndist methods and Pairwise Potentials and MODCHECK, with TM-score as the structural similarity measure

Decoy Dataset	HT-NN-dist		HT-NN-solvpair		HT-NN-solvpairndist	
	PP	MODCHECK	PP	MODCHECK	PP	MODCHECK
4state_reduced	0.5000	0.1562	0.9844	0.9844	0.9844	0.9844
bakerdecoys	0.9771	0.9642	0.8226	0.2961	0.5581	0.2039
fisa_casp3	0.9375	0.9375	0.6875	0.5625	0.8750	0.5625
fisa	0.8750	0.8750	0.8750	0.8750	0.8750	0.8750
lattice_ssfit	0.9609	0.9453	0.7695	0.9023	0.7266	0.9023
lmds	0.6875	0.3848	0.9990	0.9951	0.9346	0.9473
lmds_v2	0.8125	0.8623	0.5771	0.7217	0.5391	0.7842
all (less semfold)	0.9997	0.9948	0.9997	0.9997	0.9957	0.9902
$\alpha\beta$ (less semfold)	0.9926	0.9877	0.9692	0.9488	0.8094	0.8220
α-only (less semfold)	0.9926	0.8906	0.9983	0.9905	0.9983	0.9847
β-only (less semfold)	0.9453	0.9961	0.5000	0.9023	0.5273	0.5273

Table 3.14: Spearman correlation coefficient : P-values of one-tailed Wilcoxon sign-rank test between the HT-NN-dist, HT-NN-solvpair, HT-NN-solvpairndist methods and Pairwise Potentials and MODCHECK, with GDT-TS as the structural similarity measure

Decoy Dataset	HT-NN-dist		HT-NN-solvpair		HT-NN-solvpairndist	
	PP	MODCHECK	PP	MODCHECK	PP	MODCHECK
4state_reduced	0.5000	0.1094	0.9844	0.9844	0.9844	0.9844
bakerdecoys	0.9788	0.9833	0.6576	0.6576	0.2850	0.2039
fisa_casp3	0.9375	0.9375	0.6875	0.5625	0.6875	0.5000
fisa	0.9375	0.9375	0.8750	0.8750	0.8750	0.8750
lattice_ssfit	0.3711	0.5781	0.8438	0.9023	0.7695	0.9023
lmds	0.3848	0.2158	0.9756	0.9756	0.8125	0.7842
lmds_v2	0.6523	0.6875	0.3477	0.2158	0.2783	0.3125
all (less semfold)	0.9935	0.9742	0.9925	0.9917	0.9285	0.8804
$\alpha\beta$ (less semfold)	0.9488	0.9187	0.8341	0.8220	0.5094	0.5283
α-only (less semfold)	0.9640	0.1752	0.9899	0.9674	0.9847	0.9365
β-only (less semfold)	0.9961	0.9961	0.5781	0.9023	0.4219	0.6289

Table 3.15: Spearman correlation coefficient : P-values of one-tailed Wilcoxon sign-rank test between the HT-NN-dist, HT-NN-solvpair, HT-NN-solvpairndist methods and Pairwise Potentials and MODCHECK, with MaxSub as the structural similarity measure

Table 3.16 shows the P-values of the one-tailed Wilcoxon test, where the model ranking ability of the homologue threading methods are compared to that of the corresponding basic neural network counterparts.

It can be seen that, at a 5% significance level, the null hypothesis can be rejected for

- HT-NN-dist and NN-dist, 4state_reduced dataset, all similarity measures
- HT-NN-solvpairndist and NN-solvpairndist, Baker dataset, all similarity measures
- HT-NN-solvpair and NN-solvpair, Baker dataset, all similarity measures
- HT-NN-dist and NN-dist, all proteins, all similarity measures
- HT-NN-solvpair and NN-solvpair, all proteins, TM-score and MaxSub
- HT-NN-solvpairndist and NN-solvpairndist, all proteins, all similarity measures
- HT-NN-solvpairndist and NN-solvpairndist, β-only proteins, all similarity measures
- HT-NN-solvpairndist and NN-solvpairndist, lmds dataset, GDT-TS and MaxSub
- HT-NN-solvpair and NN-solvpair, lmds dataset, MaxSub
- HT-NN-solvpairndist and NN-solvpairndist, α-only dataset, GDT-TS

All in all, at 5% significance level, the homologue threading methods can rank models better than their basic NN counterparts in the combined dataset, and in several other individual datasets.

Decoy Dataset	HT-NN-dist			HT-NN-solvpair			HT-NN-solvpairndist		
	NN-dist			NN-solvpair			NN-solvpairndist		
	TM-score	GDT-TS	MaxSub	TM-score	GDT-TS	MaxSub	TM-score	GDT-TS	MaxSub
4state_reduced	**0.0156**	**0.0156**	**0.0156**	0.2812	0.2812	0.2812	0.2188	0.2812	0.2812
bakerdecoys	0.3265	0.1906	0.3437	**0.0028**	**0.0028**	**0.0065**	**0.0015**	**0.0007**	**0.0028**
fisa_casp3	0.1250	0.1250	0.2500	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000
fisa	0.4375	0.4375	0.4375	0.8125	0.8125	0.6875	0.6875	0.6875	0.6875
lattice_ssfit	0.4062	0.4688	0.2891	0.5938	0.7656	0.7656	0.7109	0.8125	0.7656
lmds	0.2852	0.2852	0.1016	0.1797	0.1797	**0.0488**	0.0645	**0.0488**	**0.0371**
lmds_v2	0.3477	0.5000	0.5000	0.2461	0.3477	0.1875	0.1377	0.2461	0.1611
all (less semfold)	**0.0141**	**0.0173**	**0.0094**	**0.0315**	0.0688	**0.0215**	**0.0072**	**0.0117**	**0.0055**
$\alpha\beta$ (less semfold)	0.3574	0.4516	0.3804	0.1206	0.1479	0.1083	0.0969	0.0969	0.0969
α-only (less semfold)	0.0766	**0.0172**	0.0580	0.2839	0.3985	0.4319	0.2648	0.3446	0.3767
β-only (less semfold)	0.5938	0.5938	0.4062	0.1094	0.0781	0.0781	**0.0391**	**0.0234**	**0.0391**

Table 3.16: Spearman correlation coefficient : P-values of one-tailed Wilcoxon sign-rank test between the HT-NN-dist, HT-NN-solvpair, HT-NN-solvpairndist methods and the corresponding basic NN methods

3.3.5.2 P-values of the Spearman correlation coefficients for the Sequence Profile Methods

Tables 3.17, 3.18 and 3.19 show the P-values obtained from the one-tailed Wilcoxon sign-rank test for the comparison of the SP-NN-dist, SP-NN-solvpair and SP-NN-solvpairndist methods, with the pairwise potentials method and MODCHECK, with the Spearman correlation coefficients of the output scores of the method, and the structural similarity measures, which are the TM-score, GDT-TS and MaxSub respectively.

It can be seen from Tables 3.17 to 3.19 that all the P-values are ≥ 0.05 except for the following cases.

- SP-NN-solvpairndist and pairwise potentials, Baker dataset, TM-score, P-value = 0.0412
- SP-NN-solvpairndist and MODCHECK, Baker dataset, TM-score, P-value = 0.0333
- SP-NN-solvpairndist and MODCHECK, Baker dataset, GDT-TS, P-value = 0.0473

The SP-NN-solvpairndist, which has the highest Z score among all the methods as shown in Figure 3.16, appears to perform model ranking in the Baker dataset better than the pairwise potentials/MODCHECK method for the TM-score/GDT-TS score, as shown in the above cases.

Table 3.20 shows the P-values of the one-tailed Wilcoxon test for the comparison of the SP-NN-dist, SP-NN-solvpair and SP-NN-solvpairndist methods with the corresponding basic NN counterparts.

Decoy Dataset	SP-NN-dist		SP-NN-solvpair		SP-NN-solvpairndist	
	PP	**MODCHECK**	**PP**	**MODCHECK**	**PP**	**MODCHECK**
4state_reduced	0.6562	0.5781	0.7812	0.2188	0.5000	0.2188
bakerdecoys	0.8883	0.8820	0.1313	0.0838	**0.0412**	**0.0333**
fisa_casp3	0.9375	0.9375	0.4375	0.4375	0.4375	0.4375
fisa	0.6875	0.6875	0.3125	0.1250	0.4375	0.3125
lattice_ssfit	0.9453	0.9023	0.1914	0.2734	0.1250	0.2734
lmds	0.8838	0.6523	0.9678	0.9678	0.9346	0.8838
lmds_v2	0.9199	0.9033	0.6152	0.7539	0.6523	0.8125
semfold	0.9844	0.9844	0.9219	0.9219	0.9219	0.8906
all	0.9999	0.9980	0.6095	0.4662	0.4131	0.2440
$\alpha\beta$	0.9817	0.9323	0.8341	0.8764	0.6994	0.7724
α-only	1.0000	0.9989	0.4223	0.2653	0.4070	0.2653
β-only	0.4219	0.6289	0.4727	0.5781	0.2734	0.3711

Table 3.17: Spearman correlation coefficient : P-values of one-tailed Wilcoxon sign-rank test between the SP-NN-dist, SP-NN-solvpair, SP-NN-solvpairndist methods and Pairwise Potentials and MODCHECK, with TM-score as the structural similarity measure

Decoy Dataset	SP-NN-dist		SP-NN-solvpair		SP-NN-solvpairndist	
	PP	**MODCHECK**	**PP**	**MODCHECK**	**PP**	**MODCHECK**
4state_reduced	0.5781	0.5781	0.7188	0.2188	0.5000	0.2188
bakerdecoys	0.8883	0.8052	0.2325	0.1948	0.0789	**0.0473**
fisa_casp3	0.9375	0.8750	0.4375	0.3125	0.4375	0.3125
fisa	0.6875	0.6875	0.3125	0.1250	0.3125	0.1875
lattice_ssfit	0.8750	0.9609	0.2734	0.4219	0.2734	0.4219
lmds	0.7842	0.4609	0.9580	0.9580	0.9473	0.9033
lmds_v2	0.8125	0.8838	0.6523	0.7842	0.6523	0.7539
semfold	0.9844	0.9688	0.9688	0.9219	0.9688	0.9219
all	0.9991	0.9892	0.8351	0.3682	0.6362	0.3928
$\alpha\beta$	0.9690	0.9105	0.9273	0.8978	0.8341	0.7943
α-only	0.9999	0.9896	0.8163	0.3548	0.8111	0.3476
β-only	0.3203	0.5781	0.3711	0.7266	0.2734	0.3203

Table 3.18: Spearman correlation coefficient : P-values of one-tailed Wilcoxon sign-rank test between the SP-NN-dist, SP-NN-solvpair, SP-NN-solvpairndist methods and Pairwise Potentials and MODCHECK, with GDT-TS as the structural similarity measure

Decoy Dataset	SP-NN-dist		SP-NN-solvpair		SP-NN-solvpairndist	
	PP	MODCHECK	PP	MODCHECK	PP	MODCHECK
4state_reduced	0.7188	0.5781	0.7812	0.2812	0.7188	0.2188
bakerdecoys	0.9302	0.9162	0.2325	0.1610	0.1117	0.0506
fisa_casp3	0.9375	0.9375	0.4375	0.3125	0.4375	0.3125
fisa	0.6875	0.6875	0.3125	0.1250	0.3125	0.1875
lattice_ssfit	0.4727	0.5000	0.4219	0.4219	0.2734	0.2734
lmds	0.2461	0.5000	0.9580	0.9756	0.9033	0.8389
lmds_v2	0.8125	0.8389	0.5000	0.5771	0.4229	0.5391
semfold	0.9844	0.9844	0.9531	0.8906	0.9219	0.8906
all	0.9984	0.9897	0.7467	0.4778	0.5477	0.1693
$\alpha\beta$	0.8247	0.7609	0.8686	0.7835	0.5886	0.5886
α-only	0.9999	0.9975	0.6879	0.2783	0.6452	0.2166
β-only	0.5000	0.6797	0.4727	0.7695	0.3711	0.4219

Table 3.19: Spearman correlation coefficient : P-values of one-tailed Wilcoxon sign-rank test between the SP-NN-dist, SP-NN-solvpair, SP-NN-solvpairndist methods and Pairwise Potentials and MODCHECK, with MaxSub as the structural similarity measure

Decoy Dataset	SP-NN-dist			SP-NN-solvpair			SP-NN-solvpairndist		
	NN-dist			NN-solvpair			NN-solvpairndist		
	TM-score	GDT-TS	MaxSub	TM-score	GDT-TS	MaxSub	TM-score	GDT-TS	MaxSub
4state_reduced	0.5781	0.5781	0.5781	**0.0156**	**0.0156**	**0.0156**	**0.0156**	**0.0156**	**0.0156**
bakerdecoys	0.2961	0.2529	0.2961	**0.0003**	**0.0001**	**0.0008**	**0.0010**	**0.0004**	**0.0017**
fisa_casp3	0.3125	0.1875	0.4375	0.1250	0.1250	0.1250	0.0625	0.0625	0.0625
fisa	0.0625	0.0625	0.0625	0.1250	0.1250	0.1250	0.1250	0.1250	0.1250
lattice_ssfit	0.2305	0.2305	0.3711	0.0977	0.2305	0.0547	0.1250	0.2305	0.0977
lmds	0.5771	0.8389	0.7217	0.0967	0.0527	0.1377	0.5000	0.3477	0.4229
lmds_v2	0.7217	0.6875	0.7217	0.4229	0.7217	0.4609	0.3848	0.4609	0.4609
semfold	0.0781	0.1094	0.0781	0.0781	0.0781	**0.0469**	0.2812	0.1562	0.1562
all	0.1090	0.1157	0.1508	**1.9e-7**	**2.5e-7**	**3.9e-7**	**6.4e-6**	**4.3e-6**	**6.6e-6**
$\alpha\beta$	0.0727	0.1567	0.0584	**0.0062**	**0.0032**	**0.0029**	**0.0261**	**0.0200**	**0.0337**
α-only	0.5853	0.5234	0.8412	**0.0001**	**0.0001**	**0.0010**	**0.0018**	**0.0008**	**0.0020**
β-only	**0.0391**	**0.0391**	**0.0391**	**0.0391**	**0.0391**	**0.0391**	**0.0273**	**0.0195**	**0.0273**

Table 3.20: Spearman correlation coefficient : P-values of one-tailed Wilcoxon sign-rank test between the SP-NN-dist, SP-NN-solvpair, SP-NN-solvpairndist methods and the corresponding basic NN methods

It can be seen that for all proteins, $\alpha\beta$ proteins, α-only proteins and β-only proteins, the SP-NN-solvpair and SP-NN-solvpairndist methods can rank the decoy models better than the NN-solvpair and NN-solvpairndist methods, at a 5% level of significance. The same is true of the individual Baker and 4state_reduced datasets. This suggests that the extra evolutionary information in these methods has added value in the ranking of decoy models, in the context of the proposed neural network methodology of decoy discrimination. For the SP-NN-dist method, the extra evolutionary information helps only in the β-only class of proteins.

3.3.6 Results of ROC Analysis

This section investigates how the various neural network decoy discrimination methods, including the pairwise potentials method and MODCHECK, can classify the decoy models, if the available decoy models are dichotomized into 'true' and 'false' classes. The ROC curves are drawn for each structural similarity measure, as shown in Figures 3.18 to 3.25.

As mentioned in Section 2.3.6.3, there are two sets of thresholds for the dichotomy. The first set is 6Å, 0.4, 0.25 and 0.3 for RMSD, TM-score, GDT-TS and MaxSub respectively; the second set is 4Å, 0.5, 0.35 and 0.4 for RMSD, TM-score, GDT-TS and MaxSub respectively. There are altogether 64405 models in the 64 decoy sets from the 7 decoy datasets, with the semfold dataset excluded for the sake of effective comparison with the homologue threading methods. All the models whose corresponding structural similarity measures are below the threshold are considered 'false' models, and vice versa.

Figures 3.18 and 3.19 show the ROC plots for RMSD $\leq$ 6Å and RMSD $\leq$ 4Å as the thresholds for 'true data' respectively.

Figures 3.20 and 3.21 show the ROC plots for TM-score $\geq$ 0.4 and TM-score $\geq$ 0.5 as the thresholds for 'true data' respectively.

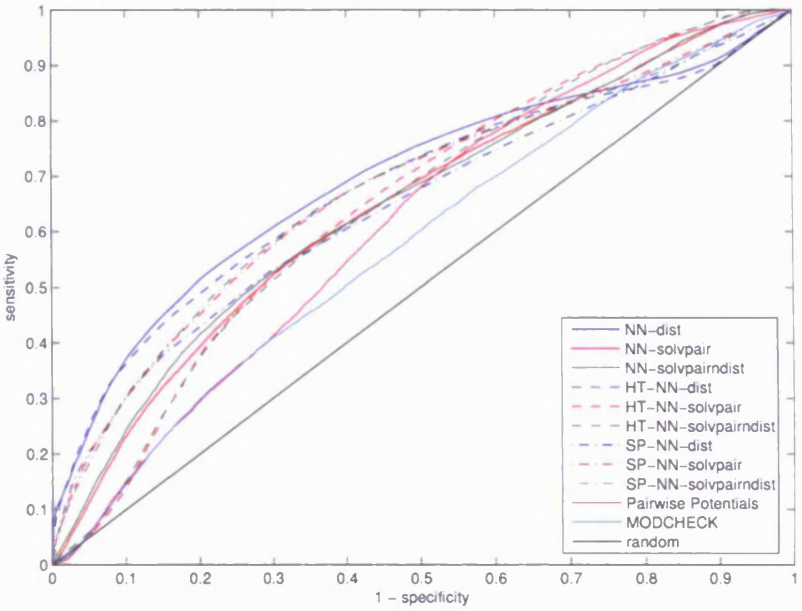


Figure 3.18: ROC plots of the NN-dist, NN-solvpair, NN-solvpairndist methods, the homologue threading (HT) methods, the sequence profile (SP) methods, Pairwise Potentials and MODCHECK using **RMSD** $\leq$ **6Å** as the threshold for 'true data' on all decoy datasets

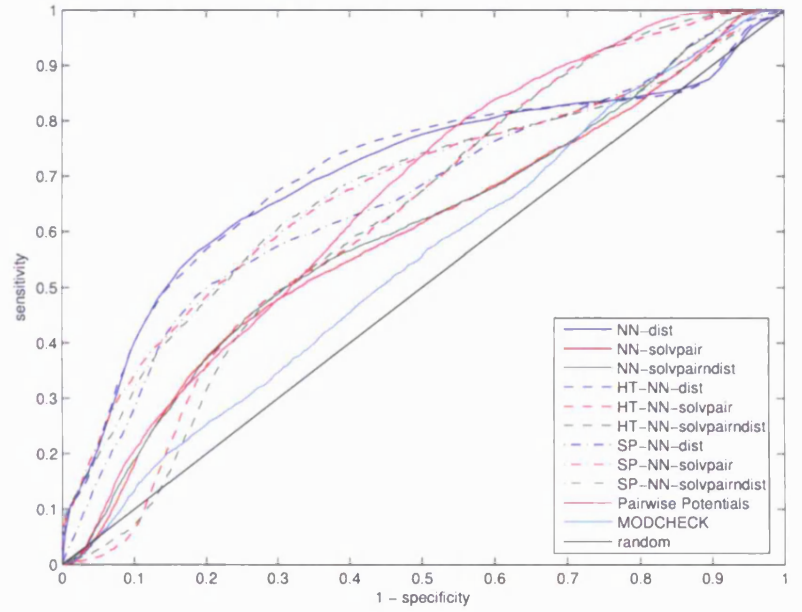


Figure 3.19: ROC plots of the NN-dist, NN-solvpair, NN-solvpairndist methods, the homologue threading (HT) methods, the sequence profile (SP) methods, Pairwise Potentials and MODCHECK using **RMSD** $\leq$ **4Å** as the threshold for 'true data' on all decoy datasets

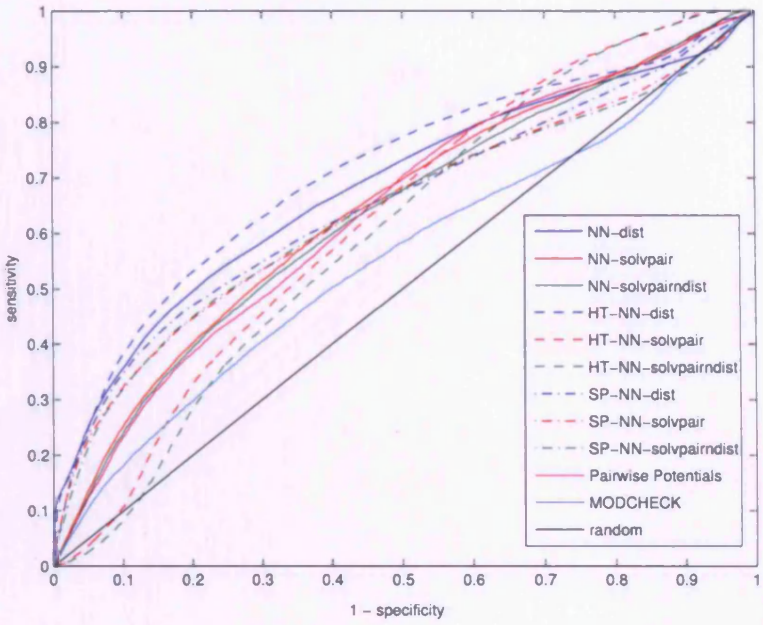


Figure 3.20: ROC plots of the NN-dist, NN-solvpair, NN-solvpairndist methods, the homologue threading (HT) methods, the sequence profile (SP) methods, Pairwise Potentials and MODCHECK using **TM-score** $\geq$ **0.4** as the threshold for 'true data' on all decoy datasets

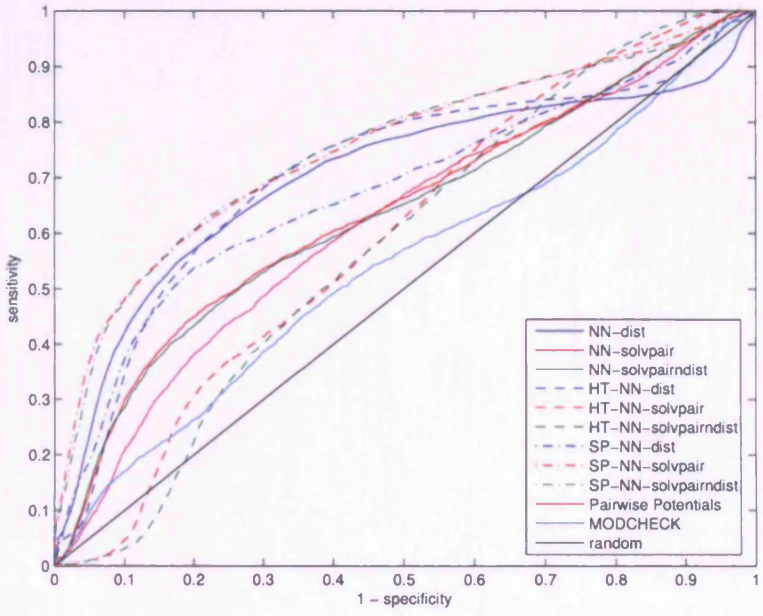


Figure 3.21: ROC plots of the NN-dist, NN-solvpair, NN-solvpairndist methods, the homologue threading (HT) methods, the sequence profile (SP) methods, Pairwise Potentials and MODCHECK using **TM-score** $\geq$ **0.5** as the threshold for 'true data' on all decoy datasets

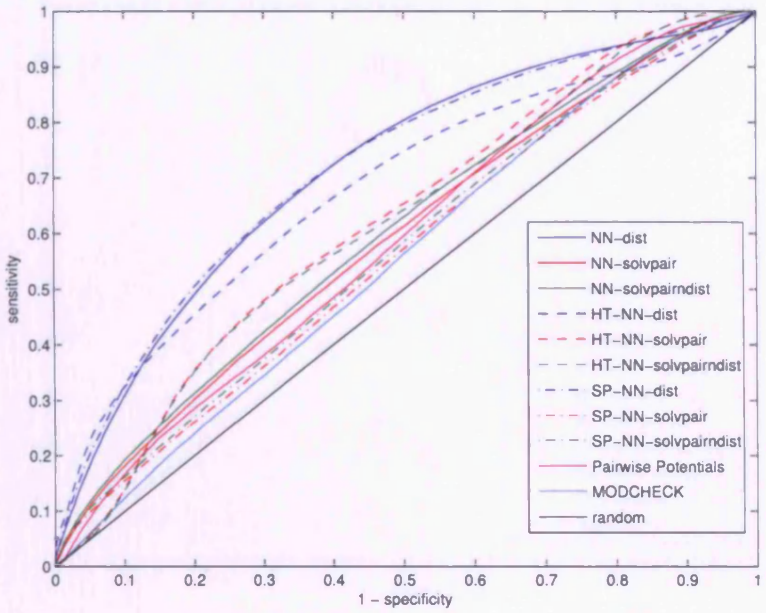


Figure 3.22: ROC plots of the NN-dist, NN-solvpair, NN-solvpairndist methods, the homologue threading (HT) methods, the sequence profile (SP) methods, Pairwise Potentials and MODCHECK using **GDT-TS** $\geq$ **0.25** as the threshold for 'true data' on all decoy datasets

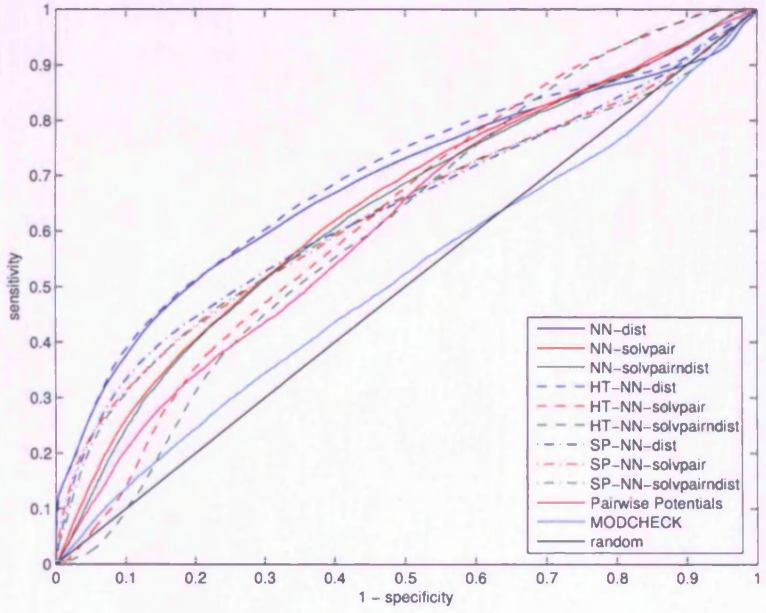


Figure 3.23: ROC plots of the NN-dist, NN-solvpair, NN-solvpairndist methods, the homologue threading (HT) methods, the sequence profile (SP) methods, Pairwise Potentials and MODCHECK using **GDT-TS** $\geq$ **0.35** as the threshold for 'true data' on all decoy datasets

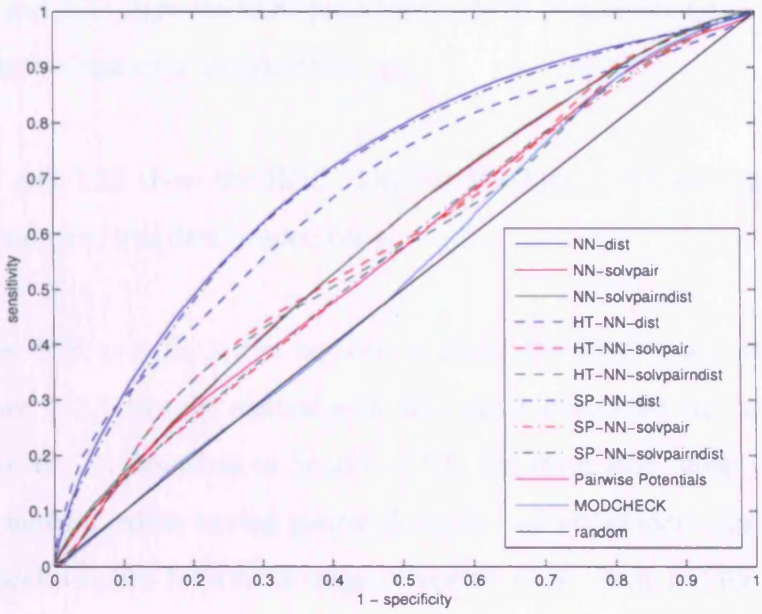


Figure 3.24: ROC plots of the NN-dist, NN-solvpair, NN-solvpairndist methods, the homologue threading (HT) methods, the sequence profile (SP) methods, Pairwise Potentials and MODCHECK using **MaxSub** $\geq$ **0.3** as the threshold for 'true data' on all decoy datasets

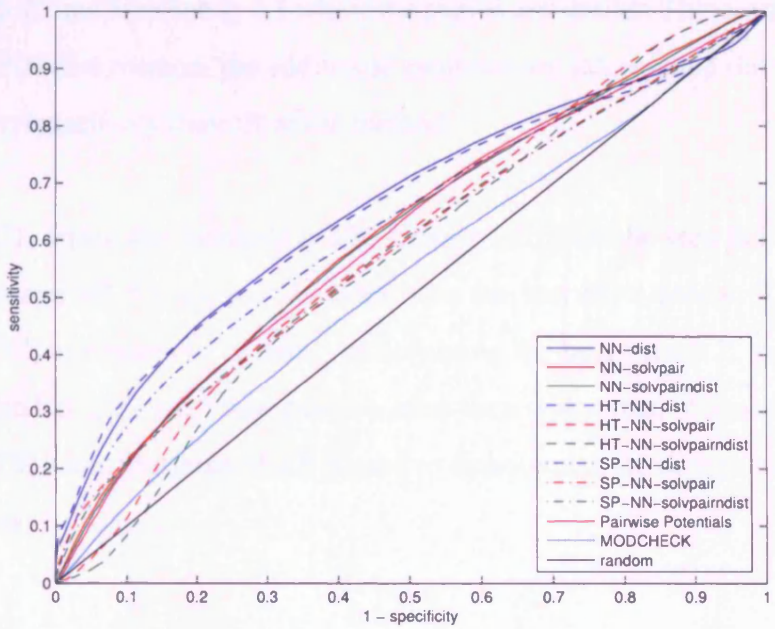


Figure 3.25: ROC plots of the NN-dist, NN-solvpair, NN-solvpairndist methods, the homologue threading (HT) methods, the sequence profile (SP) methods, Pairwise Potentials and MODCHECK using **MaxSub** $\geq$ **0.4** as the threshold for 'true data' on all decoy datasets

Figures 3.22 and 3.23 show the ROC plots for GDT-TS $\geq$ 0.25 and GDT-TS $\geq$ 0.35 as the thresholds for 'true data' respectively.

Figures 3.24 and 3.25 show the ROC plots for MaxSub $\geq$ 0.3 and MaxSub $\geq$ 0.4 as the thresholds for 'true data' respectively.

From Figures 3.18 to 3.25, it can be seen in all of the ROC plots, except the one with TM-score $\geq$ 0.5, that the method with the largest area under the curve is still the NN-dist method. As described in Section 2.5.8, for these ROC plots, it seems that the NN-dist method, while having poorer Z scores and enrichment scores, generates lower false positive rates for a wide range of sensitivities. There is little difference in the performance of the ROC plots between the other methods, including the pairwise potentials method, across the various structural similarity measures.

One interesting observation is that the SP-NN-dist method performs worse than the NN-dist method in terms of the area under the ROC curve for all thresholds, except GDT-TS $\geq$ 0.25 and MaxSub $\geq$ 0.3 where the curves are similar. This seems to suggest that for the NN-dist method, the additional evolutionary information did not improve the sensitivity/specificity tradeoff of the method.

In Figure 3.21, where the threshold is TM-score $\geq$ 0.5, it can be seen that the SP-NN-solvpairndist and SP-NN-solvpair methods have the best ROC curves. This suggests that the SP-NN-solvpairndist method, while having the best overall Z scores and enrichment, also has the lowest false positive rates for a wide range of sensitivities when a stringent TM-score threshold of 0.5 is used to dichotomize the decoys into 'true' and 'false' classes.

3.4 Summary

In this chapter, additional input information in the form of multiple sequence information have been added to the NN-dist, NN-solvpair and NN-solvpairndist methods

developed in the previous chapter. The new methods developed to accommodate multiple sequence information can be classified into two types of variants, namely the homologue threading (HT) methods and the sequence profile (SP) methods.

The homologue threading (HT) methods select the top 10 sequence homologues from the results of a PSI-BLAST search for each protein. For each decoy dataset, each of these homologues is then threaded to every decoy structure, including the native structure. The network scores evaluated on each structure from these 10 homologues, together with the original sequence, are then averaged to yield a mean score for each structure in the decoy dataset. The homologue threading methods consist of HT-NN-dist, HT-NN-solvpair and HT-NN-solvpairndist methods.

The sequence profile (SP) methods consist of using PSI-BLAST profiles as inputs, for both training and testing. This is possible because the design of the NN-dist, NN-solvpair and NN-solvpairndist neural networks have 2 20×1 vectors for sequence identities, and could readily accommodate PSI-BLAST profiles. In fact, the designs of the basic NN methods are done with the eventuality of including PSI-BLAST profile information in mind. The sequence profile methods are SP-NN-dist, SP-NN-solvpair and SP-NN-solvpairndist.

The benchmarking tests, as mentioned in Section 2.6, include

- Z score, for measuring how many standard deviations the score of the native structure is away from the mean score of all decoys.
- enrichment, for the degree to which the method can associate low RMSD decoys with high output scores.
- top model selection using the Wilcoxon sign-rank test between each proposed machine learning method and the pairwise potentials method.
- ranking of the decoy models with Spearman rank correlation coefficient, which also uses the Wilcoxon sign-rank test between each proposed machine learning method and the pairwise potentials method.

- ROC analysis

In the statistical tests, the competitive MODCHECK MQAP method is also tested against the homologue threading and sequence profile methods. This is done to see if the neural network methods can outperform the MODCHECK method in both the top model selection and the ranking of models.

The homologue threading methods have a modest success in terms of the Z score and enrichment over the basic NN methods. It is therefore suggested that some noise can be reduced by using the homologue threading methods, but not by much.

It turns out that the SP-NN-solvpairndist method is the most promising of the additional methods developed in this chapter. Firstly, it outperforms all other methods, including the in-house tried and tested pairwise potentials method, in terms of the discrimination of native structure from a set of decoys. This is highlighted in the performance of the Z score in Figure 3.16. From Figure 3.17, the SP-NN-solvpairndist method also has the highest enrichment score among all other methods although the increase over the other methods is much less pronounced than that of the Z score.

For the top model selection using the one-tailed Wilcoxon sign-rank test, the null hypothesis is that the median is zero in the distribution of the differences in the structural similarity scores of the top ranked model produced by the particular NN method and the top ranked model produced by the pairwise potentials method.

It turns out that at a 5% significance level, apart from one isolated case, there is no evidence to reject the null hypothesis that the median of the distribution of differences between the structural similarity scores of the top ranked models produced by the homologue threading methods and those produced by the pairwise potentials method (and MODCHECK) is zero. In other words, this means that the homologue threading methods show no improvement over the pairwise potentials method or MODCHECK in terms of top model selection. The same could be said of the sequence profile methods as well. At a 5% significance level, the hypothesis that there is no difference between the sequence profile methods and the pairwise potentials method (and MODCHECK)

in the structural similarity scores of their top ranked models cannot be rejected. The conclusions are the same across the three structural similarity measures for both types of methods.

The top model selection test is also performed between the homologue threading methods and the corresponding basic NN methods. It turns out that at a 5% significance level, there is no improvement in top model selection between the homologue threading methods and the corresponding basic NN method.

The top model selection test is also performed between the sequence profile methods and the corresponding basic NN methods. At a 5% significance level, for the α-only dataset, the SP-NN-solvpairndist method outperforms the NN-solvpairndist method in top model selection for all the structural similarity measures. The null hypothesis is also rejected in some other cases as shown in Table 3.12. This demonstrates that evolutionary information is useful in improving the selection of the top model, as measured by any of the structural similarity measures, in the context of the NN-solvpairndist method.

The decoy models are also ranked with the outputs of various methods using the Spearman rank correlation coefficient. The one-tailed Wilcoxon sign-rank test is again used here to test the null hypothesis that there is no difference in the distributions of the difference in Spearman rank correlation coefficients produced by the particular neural network method and the pairwise potentials method (or MODCHECK). A 5% significance level is used.

At 5% significance level, there is no evidence to reject the above null hypothesis for the homologue threading methods. This means that there is no improvement for the homologue threading methods over the pairwise potentials method and MODCHECK in terms of the ranking of the decoy models measured using Spearman rank correlation coefficient. The same could be said of the sequence profile methods, although there are isolated cases of the hypothesis being rejected at a 5% significance level for the Baker

decoy dataset with the SP-NN-solvpairndist method as shown in Tables 3.17 and 3.18.

For the comparison of the homologue threading methods with the corresponding basic NN methods in the ranking of models, there are many instances of the null hypothesis being rejected, as shown in Table 3.16. For the dataset of all proteins, almost all combinations of the homologue threading methods and the structural similarity measures used reject the null hypothesis at the 5% significance level. The HT-NN-solvpair and HT-NN-solvpairndist methods also have P-values lower than 0.05 in the Baker dataset for all structural similarity measures.

For the comparison of the sequence profile methods with the corresponding basic NN methods, the SP-NN-solvpairndist and SP-NN-solvpair methods consistently outperform their basic NN counterparts in the ranking of models in the dataset of all proteins, the α-only dataset, β-only dataset, $\alpha\beta$ datasets. The increase in performance is also found in the individual 4state_reduced and Baker decoy datasets. Therefore, in the context of the NN-solvpair and NN-solvpairndist methods, additional evolutionary information does help in the ranking of decoy models.

In the ROC analysis, decoy models are dichotomized into 'true' and 'false' classes, depending on the structural similarity measure and its threshold. The performances of the ROC curves of the neural network methods, including the homologue threading and sequence profile methods, are similar to that of the pairwise potentials and MODCHECK methods, except for the NN-dist method which yields lower false positive rates for a wide range of sensitivities for different structural similarity measures and thresholds. One notable case is that of the TM-score with a threshold of 0.5 in Figure 3.21, where the SP-NN-solvpairndist method yields the lowest false positive rate among all other methods for a wide range of sensitivities.

3.5 Conclusion

The best method developed in this chapter is the SP-NN-solvpairndist method. Although the Wilcoxon sign-rank tests show no improvement in top model selection and model ranking over the tried and tested pairwise potentials method, the SP-NN-solvpairndist method has higher Z scores and enrichment over the pairwise potentials method. In other words, it has the best performance in the discrimination of native structures, and also the association of low RMSD decoys to high network scores.

It is also shown that in the context of the neural network methodology, the use of evolutionary information can substantially increase the model ranking, top model selection and discrimination of native structures among a set of decoys.

To summarize, it is demonstrated in this chapter that the hypothesis of using neural networks with evolutionary information for decoy discrimination, in comparison with the pairwise potentials method, does work. The best neural network, the SP-NN-solvpairndist method, has

- the highest Z score, for the discrimination of native structures.
- the highest enrichment score, for the association of low RMSD structures with high output scores.
- in the case of TM-score ≥ 0.5 for definition of 'true' data, the lowest false positive rates for a wide range of sensitivities.

when compared to other neural network methods, and the pairwise potentials method.

The next chapter will summarize the findings of this thesis and outlines the future work that can be undertaken on top of the current work.

Chapter 4

Conclusion

4.1 Summary and Conclusions of Work

This thesis consists of two main ideas, namely

- the novel idea of using machine learning for the decoy discrimination problem.
- the novel idea of using evolutionary information, in a machine learning context, to improve the decoy discrimination process.

The problem of decoy discrimination has to be represented in a suitable form for the application of machine learning. In this thesis, neural networks are used, and the input features to the neural networks are represented as pairwise residues along the protein sequence, the sequence separation between the two residues, the pairwise distance and the relative solvent accessibilities of the two residues.

Positive and negative training examples are required in any machine learning problem, and in this case, native structures are used as positive training examples. Negative training examples are simulated by decoy structures which are created from native structures using the sequence reversal method. In the sequence reversal method, the sequence of each native structure in the set of positive examples is reversed and threaded back onto the native structure. As demonstrated by the results obtained, the sequence reversal method seems to be a reasonable approximation of decoy structures, for the purpose of providing the set of negative training examples to the neural networks.

Depending on the amount of input information used, various types of neural net-

works are trained, and tested on publicly available decoy datasets, namely the Tsai decoy dataset from David Baker's laboratory and the Decoys 'R' Us suite of decoy datasets. The Z score is used to measure the extent of which the native structure can be discriminated from its set of decoys, while the enrichment measure is used to measure the correlation of high network scores to low RMSD decoys. The methods developed are tested against the pairwise potentials of mean force, as well as a K Nearest Neighbours algorithm.

The inclusion of evolutionary profile information as inputs to the neural networks helps to improve the decoy discrimination process, in terms of the Z scores and enrichment measure.

The best neural network method (SP-NN-solvpairndist) has input features comprising of the position-specific sequence profile information of residue pairs, together with the relative solvent accessibility of the residues and the pairwise distance between these residues. The SP-NN-solvpairndist method

- is the best among all the methods tested in discriminating native structures from the corresponding set of decoy structures, as demonstrated by the highest Z scores it has in all the decoy datasets in Figure 3.16.
- is the best in approximately half of the decoy datasets tested for the enrichment in Figure 3.17.
- shows no improvement over the pairwise potentials method in top model selection, at a 5% significance level in a one-tailed Wilcoxon sign-rank test.
- shows no improvement over the pairwise potentials method in the ranking of models, at a 5% significance level in a one-tailed Wilcoxon sign-rank test.
- has the lowest false positive rates for a wide range of sensitivities, when the decoy models are dichotomized into 'true' and 'false' classes using a TM-score ≥ 0.5 for 'true' data.

The conclusion is that the idea of applying machine learning for the decoy discrimination problem, in context of using neural networks and the proposed way of representing

the required training examples, is indeed feasible, as demonstrated in this thesis. Furthermore, decoy discrimination, in particular the identification of the native structure, can be greatly improved by using evolutionary information in the form of PSI-BLAST profiles.

The best neural network method is also shown to perform better than the tried and tested pairwise potentials method in the discrimination of native structures. This particular paradigm of using neural networks for decoy discrimination can be expanded to use more high-resolution decoy models, in place of the sequence reversal method, to provide further discriminatory power now that the basic paradigm has been shown to be feasible.

4.2 Future Work

The following points are proposed to be viable extensions to the work presented in this thesis.

- For a start, it would be interesting to further benchmark the most effective method, SP-NN-solvpairndist, against energy functions using all-atom potentials for decoy discrimination. Of course, this can only be performed against high quality decoy datasets where there are full mainchain atoms in the backbones for all decoy structures.

- It might also be worthwhile to compare the SP-NN-solvpairndist method against other MQAP methods, besides MODCHECK [91], which have been used for the evaluation of fold recognition models, even though the neural network methods are originally developed for New Fold candidate models in mind.

- In Section 3.2.4, it is mentioned that the sequence profile methods can, in theory, learn to recognize native features such as salt bridges and disulphide bridges in native structures. It would certainly be interesting to further examine the extent to which the network scores produced by the SP-NN-solvpairndist method, the best of the sequence profile methods, correlate with the presence of such features.

- It might be possible to extend this machine learning paradigm to involve the discrimination of decoys of better ($\leq$ 1Å) resolution from native structures. The reversed sequence paradigm of providing negative training examples works reasonably well for presently available decoy datasets, but is limited in the discrimination of native structures from high resolution decoys ($\leq$ 1Å). A suitable model for the negative training examples could be MODELLER [84] outputs.

- Since the SP-NN-solvpairndist method is the most effective method in discriminating native structures from a set of decoys, it can be used as a component of an energy function for the refinement of protein structures. At some part of the structure assembly or fold recognition pipeline, the SP-NN-solvpairndist method can be used to evaluate the native-like property of candidate structures, as part of the process to create a more directed search in the 3D fold space.

- In Section 2.3.7.1, it is mentioned that the S combination of network results is defined as $4 \leq k \leq 10$. Different boundaries of the S combination can be tried to see if better results can be obtained. The perturbed distance measure mentioned in Section 2.3.1.2 can also be modified to use a variety of standard deviation values other than σ=2Å.

- In Section 3.2.1, the UniRef50 sequence database is used. In future, it might be interesting to use the UniRef70 or UniRef90 databases, where sequences are at most 70% or 90% similar respectively, in the Homologue Threading methods to see how the results obtained might differ.

Appendix A

Native Residue Pair Distance Distributions (NRPDs)

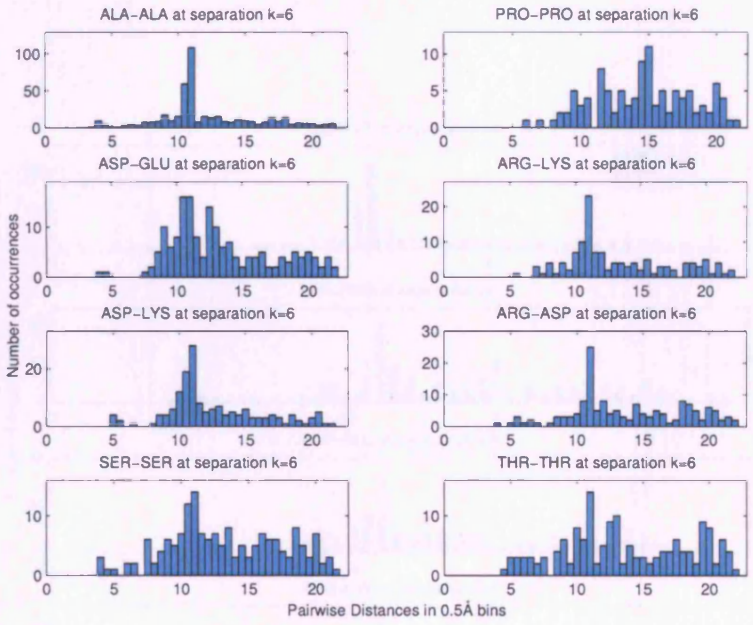


Figure A.1: Histograms of native pairwise distances of different types of residues pairs at k=6

Figure A.1 shows the native distance distributions of residue pairs of different types, at separation k=6, derived from the training dataset in Table D.1. These different types include hydrophobic residue pairs (ALA-ALA, PRO-PRO), similarly charged residue pairs (ASP-GLU, ARG-LYS), opposite charged residue pairs (ASP-LYS, ARG-ASP) and polar residue pairs (SER-SER, THR-THR).

Appendix B

Native and Decoy Residue Pair Distance Distributions (NRPDs and DRPDs)

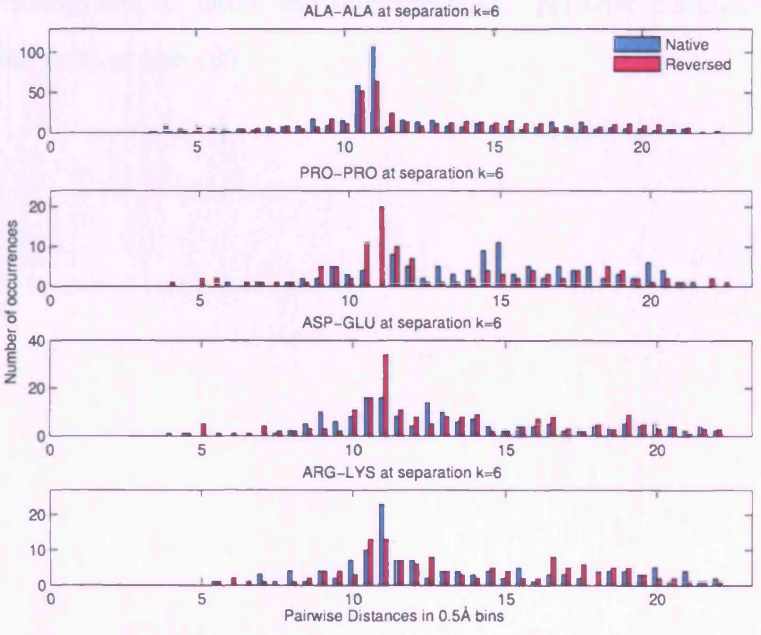


Figure B.1: Histograms of native and reversed decoy pairwise distances of different types of residue pairs at k=6 (A)

This appendix shows the decoy residue pair distance distributions (DRPDs) obtained by reversing the sequence and threading the reversed sequence to the native structure (Section 2.3.1.2). The individual plots shown in Figures B.1 and B.2 are one-to-one correspondences to the plots in Figure A.1 in Appendix A.

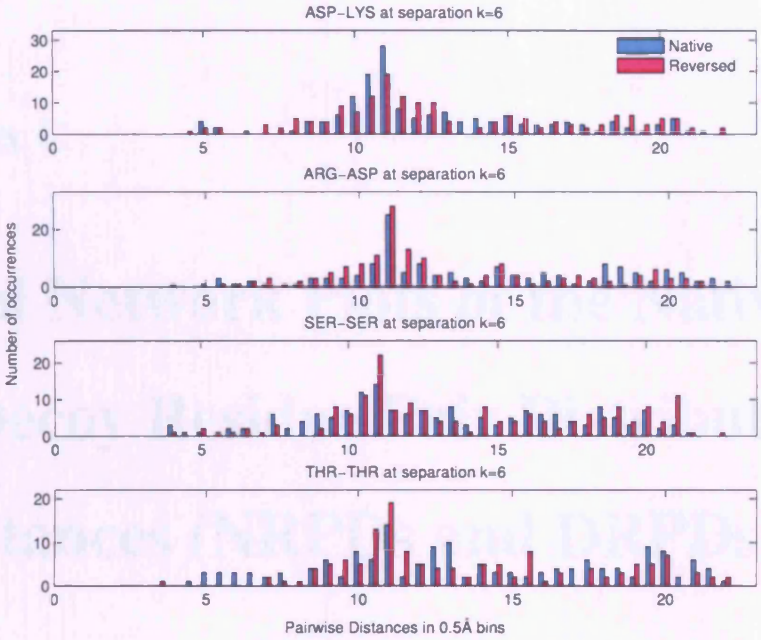


Figure B.2: Histograms of native and reversed decoy pairwise distances of different types of residue pairs at k=6 (B)

Appendix C

Neural Network Plots of the Native and Decoy Residue Pair Distributions of Distances (NRPDs and DRPDs)

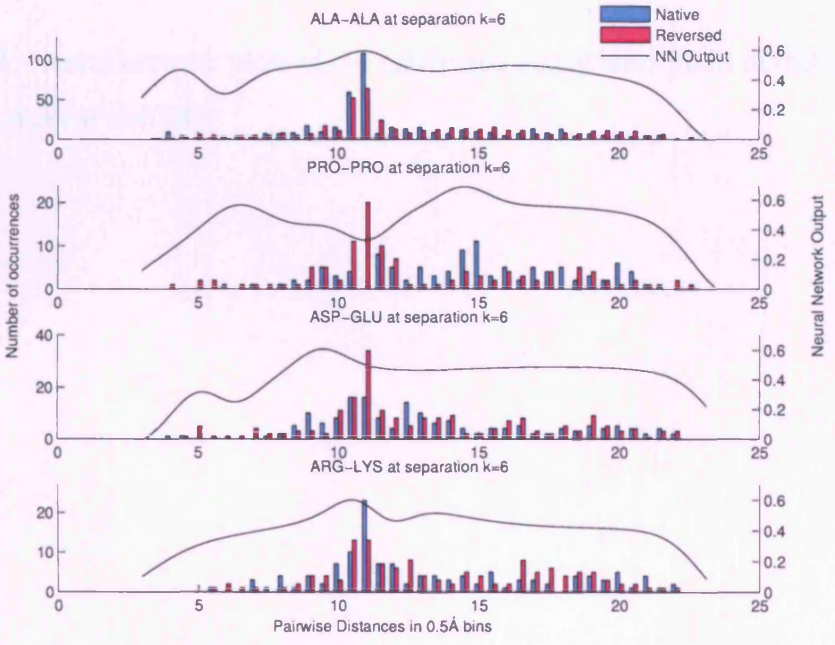


Figure C.1: Neural network plots of the native and decoy histograms of different types of residue pairs at k=6 (A)

This appendix shows the plots obtained from the k=6 neural network after it has been trained with the training dataset in Table D.1. Figures C.1 and C.2 show the neural network plots of the native and reversed decoy distance distributions of different types of residue pairs at separation k=6. These plots correspond to the NRPDs and DRPDs

shown in Figures B.1 and B.2 respectively.

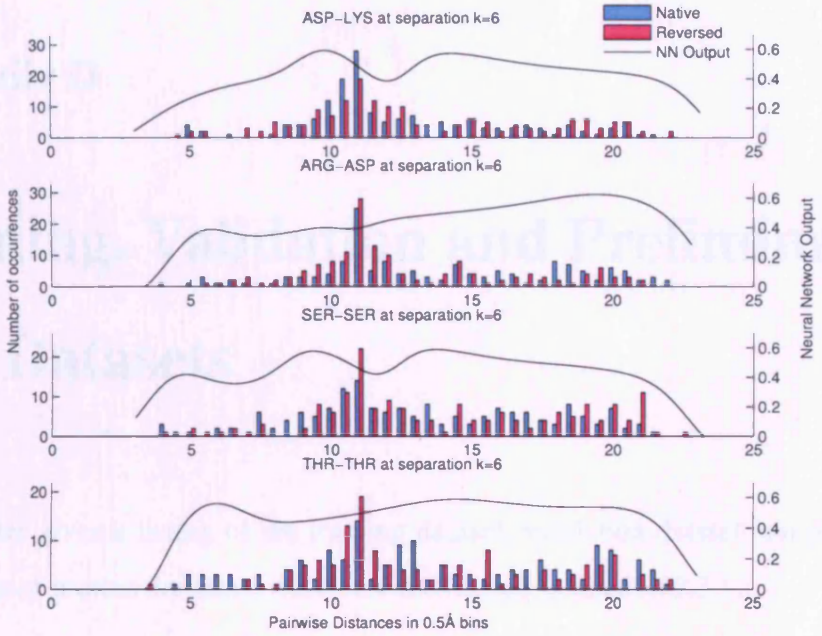


Figure C.2: Neural network plots of the native and decoy histograms of different types of residue pairs at k=6 (B)

Appendix D

Training, Validation and Preliminary Test Datasets

This chapter gives a listing of the training dataset, validation dataset and preliminary test dataset of protein domains, which are mentioned in Section 2.3.1.

Table D.1 shows the set of 285 protein domains used in training; Table D.2 shows the set of 95 protein domains used in validation during neural network training, and Table D.3 shows the set of 95 protein domains used in preliminary testing.

Training Dataset : Protein{:Chain}{:Domain Boundaries}			
1a6q:297-368	1lok:A	1etx:A	1nj4:A:263-355
1ako	1lqp:A	1ew4:A	1nls
1ayl:228-540	1ltz:A	1eyq:A	1nzi:A:1-117
1ayo:A	1m1n:A	1eyv:A	1o08:A
1bx4:A	1mf7:A	1f60:A:241-334	1o1x:A
1byq:A	1mgp:A	1g61:A	1oi7:A:122-288
1c5k:A:35-162	1mn8:A	1gs9:A	1qnf:205-475
1c97:A:2-528	1mv8:A:203-300	1gso:A:-2-103	1qre:A
1cip:A:61-181	1nf9:A	1hbn:A:2-269	1qtn:A
1cuk:156-203	1nvm:A:291-341	1hx0:A:404-496	1who

Training Dataset (cont'd)			
1d8c:A	1o98:A:77-310	1i1w:A	2pth
1ejx:A	1oo0:A	1iom:A	2uag:A:298-437
1evl:A:533-642	1orv:A:39-508	1j3a:A	3grs:364-478
1f46:A	1p3d:A:107-321	1jat:A	4eug:A
1fye:A	1qhp:A:577-686	1jhf:A:73-198	16pk
1g8l:A:327-409	1qna:A:17-115	1jkx:A	1bkr:A
1gdn:A	1rl6:A:7-81	1k3x:A:125-213	1c8z:A
1gpj:A:303-404	1tfe	1k3y:A:81-222	1cs0:B:2-152
1h4a:X:1-85	4ubp:B	1k7k:A	1dg6:A
1heu:A:164-339	1af7:11-91	1kbl:A:377-509	1di6:A
1hp1:A:363-550	1aie	1kmt:A	1dmh:A
1hqk:A	1bd8	1lam:1-159	1dqe:A
1ir1:S	1cqm:A	1ld8:A	1duv:G:1-150
1kb0:A:1-573	1dl2:A	1m15:A:2-95	1dw9:A:87-156
1kqp:A	1dto:A	1me4:A	1dy5:A
1l8a:A:701-886	1e39:A:360-505	1me8:A:2-101	1e58:A
1l8b:A	1e4c:P	1mg4:A	1eaz:A
1ldg:164-329	1ekr:A	1moo:A	1ez3:A
1lkk:A	1elk:A	1muw:A	1f8n:A:6-149
1lqt:A:2-108	1exm:A:313-405	1mvl:A	1fdr:101-248
1m1g:A:132-190	1fwx:A:8-451	1n55:A	1fsg:A
1m26:A	1gmi:A	1n60:B:7-146	1fx2:A
1mj4:A	1gs5:A	1n61:C:1-177	1g8m:A:4-200
1n1b:A:271-598	1gtk:A:220-313	1n8k:A:1-163	1gkm:A
1n3l:A	1hf8:A	1nm8:A:9-385	1gmx:A
1n62:A:82-163	1hlr:A:311-907	1o26:A	1got:G
1ns5:A	1hq1:A	1ofd:A:1240-1507	1gqz:A:1-130
1nxj:A	1ijy:A	1oht:A	1gvo:A
1oe1:A:1-159	1ix9:A:1-90	1or7:A:-1-111	1h05:A

Training Dataset (cont'd)			
1p5v:A:7-147	1iz5:A:2-120	1ox0:A:-5-251	1h3n:A:226-417
1qhv:A	1jos:A	1qcz:A	1hb6:A
1qqf:A	1k0r:A:184-262	1qh4:A:103-381	1i2t:A
1qsa:A:1-450	1khb:A:10-259	1qnx:A	1i4j:A
1rss	1khd:A:12-80	1d0c:A	1ikt:A
1tig	1kp8:A:2-136	1dd3:A:1-57	1ixb:A:91-205
1yge:150-839	1krh:A:106-205	1dj0:A:7-114	1ixh
2tps:A	1kwm:A:1A-95A	1dow:A	1j6z:A:4-146
7odc:A:44-283	1kyp:A	1dqi:A	1jke:A
1awq:A	1l6p:A	1e6i:A	1k4i:A
1bd0:A:2-11	1mix:A:195-308	1ei5:A:336-417	1k6d:A
1bxy:A	1mla:198-307	1f86:A	1knl:A
1byi	1n5u:A:2-196	1f9y:A	1l2h:A
1chd	1np7:A:1-204	1fhu:A:1-99	1l5o:A
1cxq:A	1o8b:A:199-218	1fjj:A	1lm4:A
1d5t:A:292-388	1oew:A	1fmt:A:207-314	1m6y:A:115-215
1egw:A	1p5u:A:148-234	1g7s:A:329-459	1m9n:A:201-593
1erz:A	1qop:A	1gkp:A:2-54	1mgt:A:89-169
1ftr:A:1-148	1whi	1gxr:A	1nkp:A
1gpr	1xxa:A	1hxk:A:31-411	1npk
1gxu:A	2ilk	1i9c:A	1pcf:A
1h16:A	2sns	1iu7:A:212-628	1qqq:A
1h4x:A	2spc:A	1j8b:A	1sei:A
1hqs:A	3nul	1jp3:A	1uro:A
1hs6:A:1-208	8ruc:A:9-147	1jsd:A	2aop:149-345
1hz4:A	1a77:209-316	1jw9:B	4uag:A:1-93
1jid:A	1a8o	1k8y:B	1a8d:248-452
1k4g:A	1c1d:A:1-148	1kek:A:416-668	1b8z:A
1kj9:A:113-318	1crz:A:141-409	1kpf	1bkf

Training Dataset (cont'd)			
1kq1:A	1czp:A	1ku1:A	1chm:A:2-156
1kqf:A:34-850	1dci:A	1mc2:A	1kwf:A
1dfu:P	1mfm:A	1lb3:A	1dl5:A:214-317
1mwx:A:139-327			

Table D.1: Training Dataset of 285 proteins

Validation Dataset : Protein{:Chain}{:Domain Boundaries}			
1d8h:A	1jfl:A:1-115	1bgf	1nxu:A
1dce:A:242-350	1jhg:A	1eu1:A:626-780	1o04:A
1di2:A	1ji7:A	1euw:A	1o6v:A:33-416
1dtj:A	1k2y:X:5-154	1f4l:A:389-548	1obo:A
1e0t:A:70-167	1klx:A	1f5n:A:284-583	1on2:A:63-136
1e85:A	1l3k:A:8-91	1f7l:A	1opd
1e8c:A:3-103	1lb6:A	1goi:A:447-498	1qh5:A
1eaq:A	1lc5:A	1gwy:A	1qlm:A
1ef1:C	1mrj	1hw1:A:79-230	1sox:A:94-343
1ewf:A:1-217	1o0w:A:-1-167	1hw5:A:1-137	1zfj:A:95-158
1f0j:A	1o1z:A	1i40:A	2mhr
1feh:A:210-574	1o6s:B	1i4m:A	2pvb:A
1fyf:A:242-532	1ogw:A	1j09:A:306-468	3sil
1g6s:A	1oi1:A:33-135	1jz8:A:731-1023	1aol
1g8t:A	1qcs:A:86-201	1k7i:A:259-479	1b6a:110-374
1gz8:A	1qnt:A:6-91	1kmv:A	1by2
1ifr:A	1t1d:A	1ku3:A	1c96:A:529-754
1ijq:A:377-642	1wpo:A	1l3p:A	1dhn
1iq4:A	1a12:A	1lfw:A:187-382	1dlj:A:295-402
1iqy:A:9-96	1a3a:A	1lsh:A:285-620	1dqa:A:587-703
1itx:A:338-409	1aop:81-145	1m1h:A:51-131	1dzf:A:5-143
1iu8:A	1b8o:A	1nm2:A:134-195	1e2w:A:1-168
1iwl:A	1bdo	1nte:A	1ekj:A
1jcl:A	1bfd:2-181	1nwa:A	

Table D.2: Validation Dataset of 95 proteins

Preliminary Test Dataset : Protein{:Chain}{:Domain Boundaries}			
1eye:A	1ks2:A:127-198	1bm8	1kjq:A:319-392
1fcy:A	1lm5:A	1d3v:A	1kwn:A
1fkm:A:249-442	1lpl:A	1dk8:A	1l0i:A
1g8e:A	1ls1:A:1-88	1dmg:A	1l3l:A:2-169
1gci	1m5w:A	1doz:A	1luc:A
1gk8:A:150-475	1m9x:C	1fma:E	1lyv:A
1gwu:A	1mmg:34-79	1fpo:A:1-76	1m22:A
1h8e:A:380-510	1moq	1g87:A:457-614	1m4j:A
1hdh:A	1mwp:A	1gte:A:2-183	1mky:A:359-439
1hxn	1n08:A	1gxj:A	1mro:A:270-549
1i1q:A	1n63:C:178-287	1h2w:A:1-430	1mzg:A
1j98:A	1nox	1h7m:A	1o7n:A:155-448
1jbe:A	1nz0:A	1hty:A:412-522	1oac:A:5-90
1jfb:A	1o7j:A	1hzt:A	1osp:O
1jg1:A	1obd:A	1io0:A	1qdd:A
1jhd:A:1-173	1pin:A:6-39	1iv3:A	1qhd:A:1-148
1jz7:A:220-333	1qjb:A	1iw0:A	1vhh
1k20:A	1slu:A	1j96:A	1vps:A
1k5n:A:182-276	1uaq:A	1j9j:A	1wer
1k92:A:189-444	1uca:A	1jf8:A	2bop:A
1kg2:A	1uxy:201-342	1jl0:A	2nac:A:1-147
1kgs:A:124-225	2sic:I	1k3w:A:1-124	3lzt
1ko7:A:1-129	1a9x:A:403-555	1k5c:A	3seb:122-238
1kr4:A	1axn	1kid	

Table D.3: Preliminary Test Dataset of 95 proteins

Appendix E

Histograms of mean k=4 neural network scores of the proteins in the Baker Decoy dataset

This appendix, as mentioned in Section 2.4.2.1, gives the histograms of the mean neural network scores at separation k=4 for the Baker decoys as listed in Table 2.5.

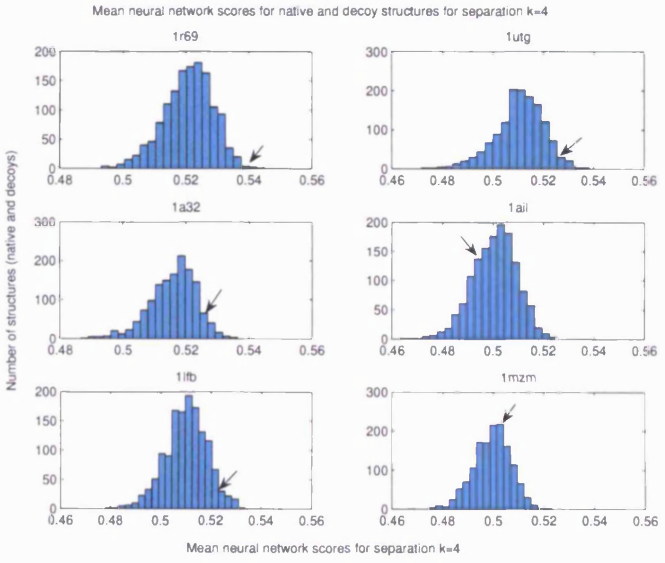


Figure E.1: Mean neural network scores for separation k=4 for the Baker decoy dataset (A)

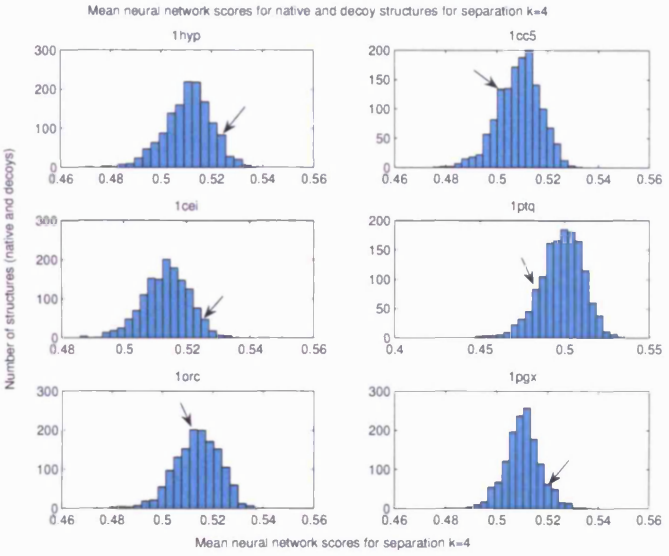


Figure E.2: Mean neural network scores for separation k=4 for the Baker decoy dataset (B)

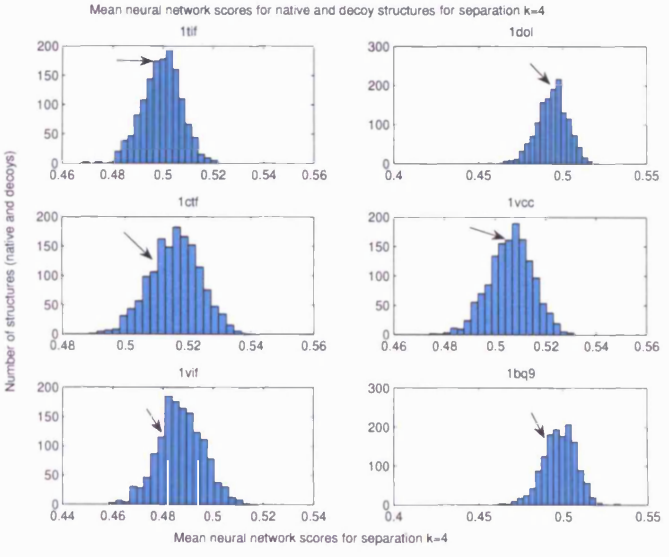


Figure E.3: Mean neural network scores for separation k=4 for the Baker decoy dataset (C)

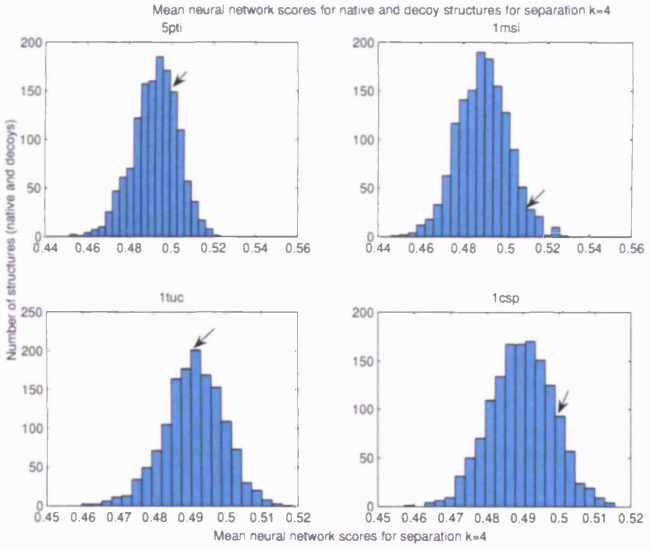


Figure E.4: Mean neural network scores for separation k=4 for the Baker decoy dataset (D)

Appendix F

Histograms of mean neural network scores for all separations *k* for protein *1r69* in the Baker decoy dataset

This appendix chapter, as mentioned in Section 2.4.2.1, gives the histograms of the mean neural network scores for all sequence separations k=4 to 22, and $k > 22$, for the *1r69* protein in the Baker decoy dataset.

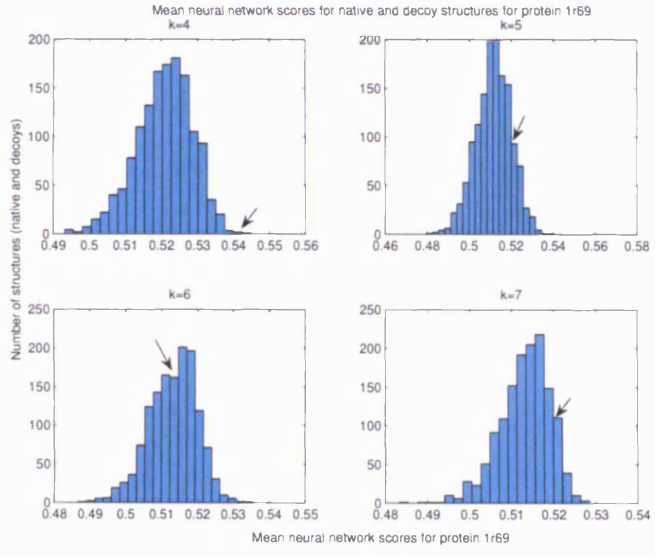


Figure F.1: Mean neural network scores for separation k=4 to 7 for structures of protein *1r69*

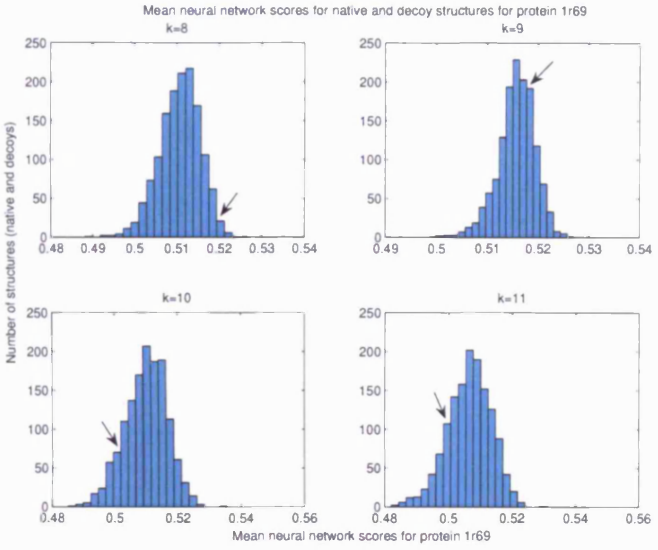


Figure F.2: Mean neural network scores for separation k=8 to 11 for structures of protein *1r69*

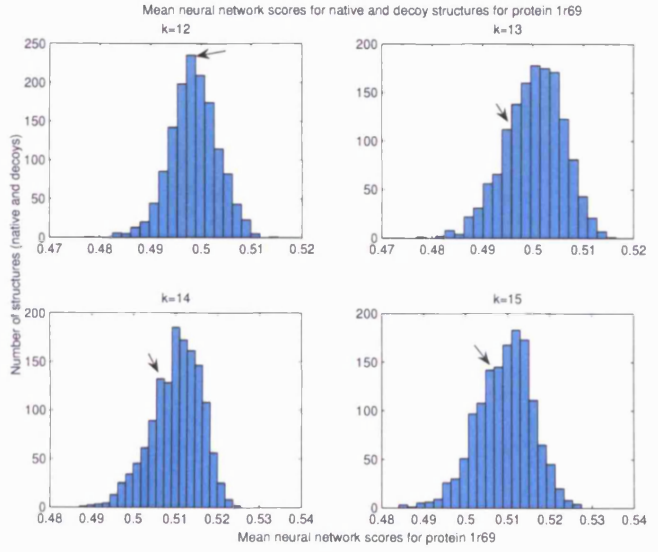


Figure F.3: Mean neural network scores for separation k=12 to 15 for structures of protein *1r69*

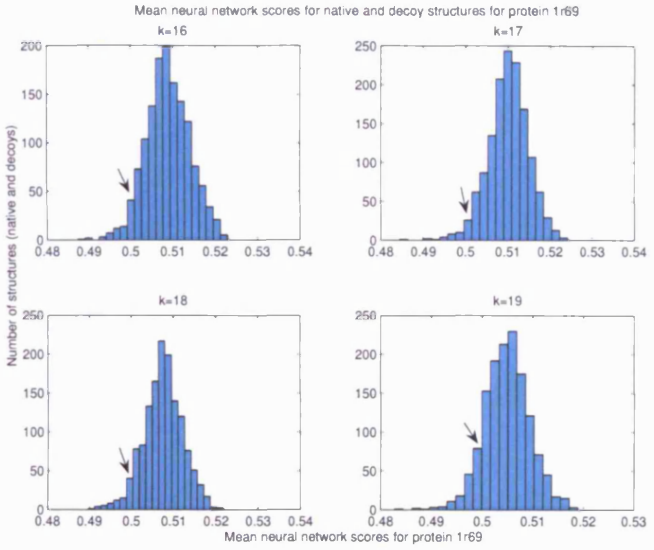


Figure F.4: Mean neural network scores for separation k=16 to 19 for structures of protein *1r69*

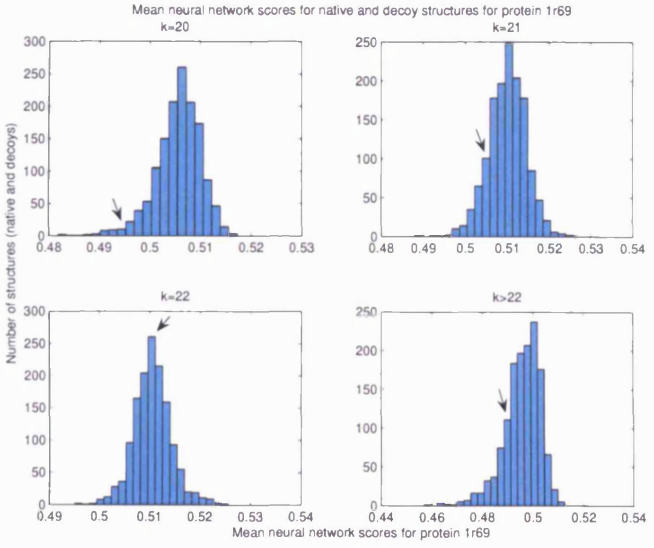


Figure F.5: Mean neural network scores for separation k=20 to 22, and $k > 22$, for structures of protein *1r69*

Appendix G

3D Scatter plots of native and simulated decoy training instances with additional solvent accessibility values

This appendix, as mentioned in Section 2.5.2, gives the 3D scatter plots of the native and simulate decoy training instances of various types of residue pairs at separation k=6. Figure G.1 shows the scatter plots of hydrophobic ALA-ALA residue pairs.

Figure G.2 shows the scatter plots of oppositely charged ASP-GLU residue pairs.

Figure G.3 shows the scatter plots of similarly charged ASP-LYS residue pairs.

Figure G.4 shows the scatter plots of polar SER-SER residue pairs.

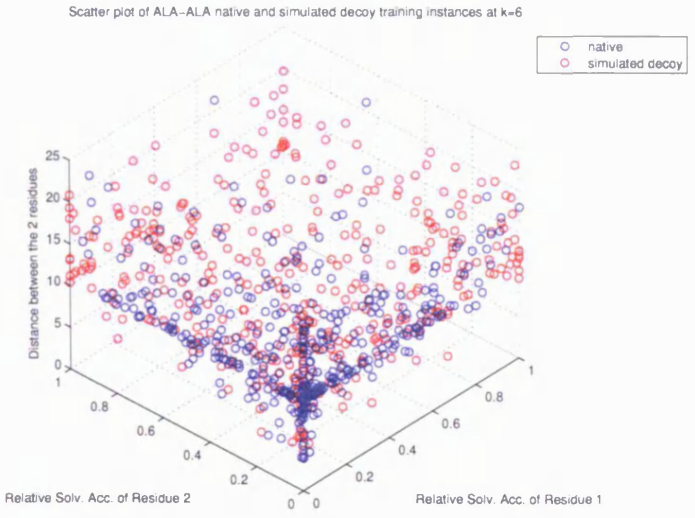


Figure G.1: Distribution of input training instances, with additional solvent accessibility information, of ALA-ALA at k=6

Figure G.2: Distribution of input training instances, with additional solvent accessibility information, of ASP-GLU at k=6

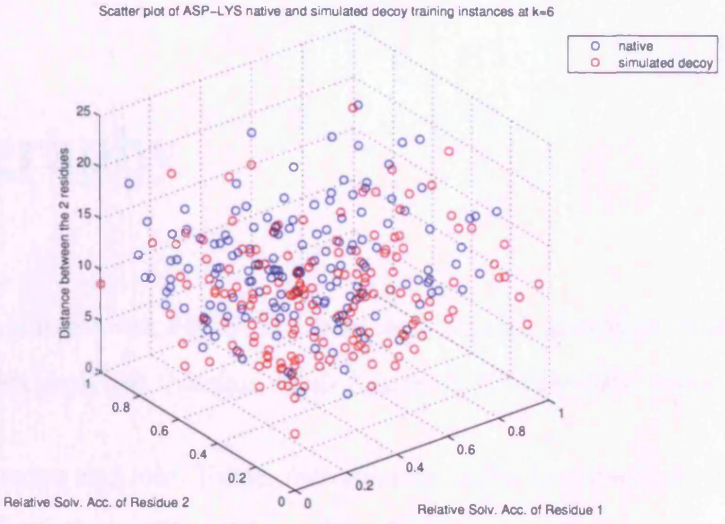


Figure G.3: Distribution of input training instances, with additional solvent accessibility information, of ASP-LYS at k=6

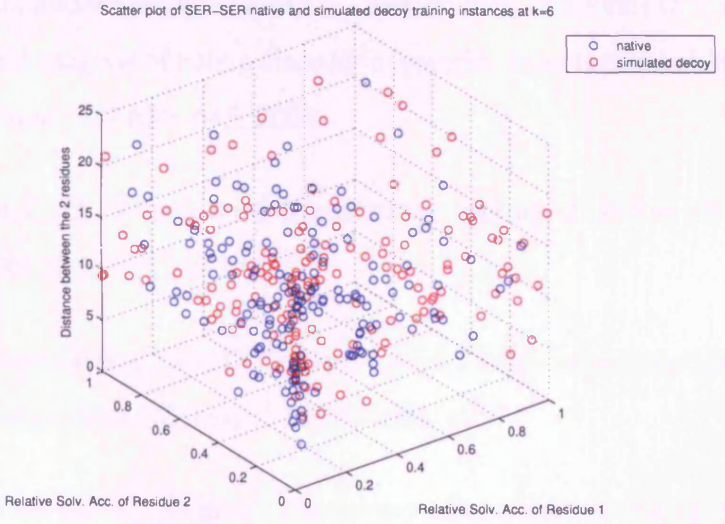


Figure G.4: Distribution of input training instances, with additional solvent accessibility information, of SER-SER at k=6

Bibliography

[1] Shuichi Kawashima, Hiroyuki Ogata, and Minoru Kanehisa. AAindex: Amino acid index database. *Nucleic Acids Research*, 27:368–369, 1999.

[2] Carl Branden and John Tooze. *Introduction to Protein Structure, Second Edition, Chapter 10.* Garland Publishing, Inc., 1999.

[3] Richard G. Brennan and Brian W. Matthews. The Helix-Turn-Helix DNA Binding Motif. *Journal of Biological Chemistry*, pages 1903–1906, 1989.

[4] Chin-Hsien Tai, Woei-Jyh Lee, James J. Vincent, and Byungkook Lee. Evaluation of Domain Prediction in CASP6. *Proteins*, 61, Suppl 7:183–192, 2005.

[5] Ward J.J., Sodhi J.S., McGuffin L.J., Buxton B.F., and Jones D.T. Prediction and functional analysis of native disorder in proteins from the three kingdoms of life. *J. Mol. Biol.*, 337:635–645, 2004.

[6] Anfinsen C. B. Principles that govern the folding of protein chains. *Science*, 181(4096):223–230, 1973.

[7] A. G. Szent-Gyorgyi and C. Cohen. Role of proline in polypeptide chain configuration of proteins. *Science*, 126:697–698, 1957.

[8] W. Kabsch and C. Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 12:2577–2637, 1983.

[9] Frishman D. and Argos P. Knowledge-based protein secondary structure assignment. *Proteins*, 23:566–579, 1995.

[10] Frederic M. Richards and Craig E. Kundrot. Identification of Structural Motifs From Protein Coordinate Data: Secondary Structure and First-Level Supersecondary Structure. *Proteins*, 3:71–84, 1988.

[11] Sonya M. King and W. Curtis Johnson. Assigning Secondary Structure From Protein Coordinate Data. *Proteins*, 35:313–320, 1999.

[12] Juliette Martin, Guillaume Letellier, Antoine Marin, Jean-Francois Taly, Alexandre G de Brevern, and Jean-Francois Gibrat. Protein secondary structure assignment revisited: a detailed analysis of different assignment methods. *BMC Structural Biology*, pages 5–17, 2005.

[13] M.N. Fodje and S.Al-Karadaghi. Occurrence, conformational features and amino acid propensities for the π-helix. *Protein Engineering*, 15(5):353–358, 2002.

[14] G. Pollastri, D. Przybylski, B. Rost, and P. Baldi. Improving the Prediction of Protein Secondary STructure in Three and Eight Classes Using Recurrent Neural Networks and Profiles. *Proteins*, 47:228–235, 2002.

[15] David T. Jones. Protein Secondary Structure Prediction Based on Position-specific Scoring Matrices. *J. Mol. Biol.*, 292:195–202, 1999.

[16] Burkhard Rost. Review: Protein Secondary Structure Prediction Continues to Rise. *Journal of Structural Biology*, 134:204–218, 2001.

[17] Cuff J.A. and Barton G.J. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins*, 34:508–519, 1999.

[18] Rost B., Sander C., and Schneider R. Redefining the goals of protein secondary structure prediction. *J. Mol. Biol.*, 235:13–28, 1994.

[19] Zemla A., Venclovas C., Fidelis K., and Rost B. A modified definition of SOV, a segment-based measure for protein secondary structure prediction assessment. *Proteins*, 34:220–223, 1999.

[20] Chou P.Y. and Fasman G.D. Conformational parameters for amino acids in helical, β-sheet, and random coil regions calculated from proteins. *Biochemistry*, 13:211–222, 1974.

[21] Garnier J., Osguthorpe D.J, and Robson B. Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. *J. Mol. Biol.*, 120:97–120, 1978.

[22] Garnier J., Gibrat J-F, and Robson B. GOR secondary structure prediction method version IV. *Meth. Enzymol*, 266:540–553, 1996.

[23] Zvelebil M. J., Barton G. J., Taylor W. R., and Sternberg M. J. E. Prediction of protein secondary structure and active sites using alignment of homologous sequences. *J. Mol. Biol.*, 195:957–961, 1987.

[24] B. Rost and C. Sander. Prediction of protein secondary structure at better than 70% accuracy. *J. Mol. Biol.*, 232:584–599, 1993.

[25] Altschul S.F., Madden T.L., Schaffer A.A., Zhang J., Zhang Z., Miller W., and Lipman D.J. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25:3389–3402, 1997.

[26] C. A. Orengo, J. E. Bray, T. Hubbard, L. LoConte, and I. Sillitoe. Analysis and assessment of ab initio three-dimensional prediction, secondary structure, and contacts prediction. *Proteins*, 37, Suppl 3:149–170, 1999.

[27] Kevin Karplus, Christian Barrett, Melissa Cline, Mark Diekhans, Leslie Grate, and Richard Hughey. Predicting Protein Structure Using Only Sequence Information. *Proteins*, 37, Suppl 3:121–125, 1999.

[28] Bystroff C., Thorsson V., and Baker D. HMMSTR: A hidden Markov model for local sequence-structure correlations in proteins. *J. Mol. Biol.*, 301:173–190, 2000.

[29] Arthur M. Lesk, Loredana Lo Conte, and Tim J. P. Hubbard. Assessment of Novel Fold Targets in CASP4: Predictions of Three-Dimensional Structures,

Secondary Structures, and Interresidue Contacts. *Proteins*, 45, Suppl 5:98–118, 2001.

[30] Solovyev V. and Salamov A.A. Predicting α-helix and β-strand segments of globular proteins. *Comput. Appl. Biosci.*, 10:661–669, 1994.

[31] Salamov A. A. and Solovyev V. V. Prediction of protein secondary structure by combining nearest-neighbor algorithms and multiple sequence alignments. *J. Mol. Biol.*, 247:11–15, 1995.

[32] King R. D. and Sternberg M. J. E. Identification and application of the concepts important for accurate and reliable protein secondary structure prediction. *Protein Science*, pages 2298–2310, 1996.

[33] David T. Jones. Learning to Speak the Language of Proteins. *Science*, 302:1347–1348, 2003.

[34] Dariusz Przybylski and Burkhard Rost. Alignments Grow, Secondary Structure Prediction Improves. *Proteins*, 46:197–205, 2002.

[35] Rost B. and Eyrich V.A. EVA: Large-scale analysis of secondary structure prediction. *Proteins*, 45, Suppl 5:192–199, 2001.

[36] David T. Jones. GenTHREADER: An Efficient and Reliable Protein Fold Recognition Method for Genomic Sequences. *J. Mol. Biol.*, 287:797–815, 1999.

[37] A. J. Shepherd, D. Gorse, and J. M. Thornton. Prediction of the location and type of beta-turns in proteins using neural networks. *Protein Science*, 8:1045–1055, 1999.

[38] Harpreet Kaur and Gajendra Pal Singh Raghava. Prediction of β-turns in proteins from multiple alignment using neural network. *Protein Science*, 12:627–634, 2002.

[39] Steven Mosimann, Ron Meleshko, and Michael N. G. James. A critical assessment of comparative molecular modeling of tertiary structures of proteins. *Proteins*, 23:301–317, 1995.

[40] Andrew C. R. Martin, Malcolm W. MacArthur, and Janet M. Thornton. Assessment of comparative modeling in CASP2. *Proteins*, 29, Suppl 1:14–28, 1997.

[41] T. Alwyn Jones and Gerard J. Kleywegt. CASP3 comparative modeling evaluation. *Proteins*, 37, Suppl 3:30–46, 1999.

[42] Veronica Morea Anna Tramontano, Raphael Leplae. Analysis and assessment of comparative modeling predictions in CASP4. *Proteins*, 45, Suppl 5:22–38, 2001.

[43] Tramontano A. and Morea V. Assessment of homology-based predictions in CASP5. *Proteins*, 53, Suppl 6:352–368, 2003.

[44] Michael Tress, Iakes Ezkurdia, Osvaldo Grana, Gonzalo Lopez, and Alfonso Valencia. Assessment of predictions submitted for the CASP6 comparative modeling category. *Proteins*, 61, Suppl 7:27–45, 2005.

[45] John Moult. A decade of CASP: progress, bottlenecks and prognosis in protein structure prediction. *Current Opinion in Structural Biology*, 15:285–289, 2005.

[46] Bowie J.U., Luthy R., and Eisenberg D. A method to identify protein sequences that fold into a known three-dimensional structure. *Science*, 253:164–170, 1991.

[47] Alexei V. Finkelstein and Boris A. Reva. A search for the most stable folds of protein chains. *Nature*, 351:497–499, 1991.

[48] John Overington, Mark S. Johnson, Andrej Sali, and Tom L. Blundell. Tertiary Structural Constraints on Protein Evolutionary Diversity: Templates, Key Residues and Structure Prediction. *Proceedings: Biological Sciences*, 241:132–145, 1990.

[49] D. T. Jones, W. R. Taylor, and J. M. Thornton. A new approach to protein fold recognition. *Nature*, 358:86–89, 1992.

[50] Jeffrey Skolnick and Andrzej Kolinski. Dynamic Monte Carlo simulations of a new lattice model of globular protein folding, structure and dynamics. *J. Mol. Biol.*, 221:499–531, 1991.

[51] A. A. Rabow and H. A. Scheraga. Lattice Neural Network Minimization: Application of Neural Network Optimization for Locating the Global-minimum Conformations of Proteins. *J. Mol. Biol.*, 232:1157–1168, 1993.

[52] David A. Hinds and Michael Levitt. Exploring Conformational Space with a Simple Lattice Model for Protein Structure. *J. Mol. Biol.*, 243:668–682, 1994.

[53] Jones D. T. Successful ab initio prediction of the tertiary structure of NK-Lysin using multiple sequences and recognized supersecondary structural motifs. *Proteins*, 29, Suppl 1:185–191, 1997.

[54] K. T. Simons, C. Kooperberg, E. Huang, and D. Baker. Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and bayesian scoring functions. *J. Mol. Biol.*, 268:209–225, 1997.

[55] Yang Zhang and Jeffrey Skolnick. Automated structure prediction of weakly homologous proteins on a genomic scale. *Proc. Natl. Acad. Sci USA*, 101:7594–7599, 2004.

[56] Lazaridis T. and Karplus M. Effective energy functions for protein structure prediction. *Current Opinion in Structural Biology*, 10:139–145, 2000.

[57] Anthony K. Felts, Emilio Gallicchio, Anders Wallqvist, and Ronald M. Levy. Distinguishing Native Conformations of Proteins From Decoys With an Effective Free Energy Estimator Based on the OPLS All-Atom Force Field and the Surface Generalized Born Solvent Model. *Proteins*, 48:404–422, 2002.

[58] Meng-Juei Hsieh and Ray Luo. Physical Scoring Function Based on AMBER Force Field and Poisson-Boltzmann Implicit Solvent for Protein Structure Prediction. *Proteins*, 56:475–486, 2004.

[59] M. J. Sippl. Calculation of conformational ensembles from potentials of mean force. an approach to the knowledge-based prediction of local structures in globular proteins. *J. Mol. Biol.*, 213:859–883, 1990.

[60] C. M. Summa, M. Levitt, and W. F. Degrado. An atomic environment potential for use in protein structure prediction. *J. Mol. Biol.*, 352:986–1001, 2005.

[61] Ortiz A. R., Hu W. P., Kolinkski A., and Skolnick J. Method for low resolution prediction of small protein tertiary structure. *Pac. Symp. Biocomput.*, pages 316–327, 1997.

[62] Yang Zhang, Adrian K. Arakaki, and Jeffrey Skolnick. TASSER: An Automated Method for the Prediction of Protein Tertiary Structures in CASP6. *Proteins*, 61, Suppl 7:91–98, 2005.

[63] Moult J., Pedersen J.T., Judson R., and Fidelis K. A large-scale experiment to assess protein structure prediction methods. *Proteins*, 23:ii–v, 1995.

[64] Moult J., Hubbard T., Bryant S.H., Fidelis K., and Pedersen J.T. Critical assessment of methods of protein structure prediction (CASP): Round II. *Proteins*, 29, Suppl 1:2–6, 1997.

[65] Moult J., Hubbard T., Fidelis K., and Pedersen J.T. Critical assessment of methods of protein structure prediction (CASP): Round III. *Proteins*, 37, Suppl 3:2–6, 1999.

[66] John Moult, Krzysztof Fidelis, Adam Zemla, and Tim Hubbard. Critical Assessment of Methods of Protein Structure Prediction (CASP): Round IV. *Proteins*, 45, Suppl 5:2–7, 2001.

[67] John Moult, Krzysztof Fidelis, Adam Zemla, and Tim Hubbard. Critical Assessment of Methods of Protein Structure Prediction (CASP): Round V. *Proteins*, 53, Suppl 6:334–339, 2003.

[68] John Moult, Krzysztof Fidelis, Burkhard Rost, Tim Hubbard, and Anna Tramontano. Critical assessment of methods of protein structure prediction (CASP) - Round 6. *Proteins*, 61, Issue S7:3–7, 2005.

[69] `http://predictioncenter.org`.

[70] Adam Zemla. LGA: a method for finding 3D similarities in protein structures. *Nucleic Acids Research*, 31:3370–3374, 2003.

[71] Hubbard T.J.P. RMS/Coverage graphs: A qualitative method for comparing three-dimensional protein structure predictions. *Proteins*, 37, Suppl 3:15–21, 1999.

[72] `http://www.cs.bgu.ac.il/~dfischer/CAFASP5`.

[73] Krzysztof Ginalski, Arne Elofsson, Daniel Fischer, and Leszek Rychlweski. 3D-Jury: a simple approach to improve protein structure predictions. *Bioinformatics*, 19(8):1015–1018, 2003.

[74] Andriy Kryshtafovych, Ceslovas Venclovas, Krzysztof Fidelis, and John Moult. Progress Over the First Decade of CASP Experiments. *Proteins*, 61, Suppl 7:225–236, 2005.

[75] Reva B. A., Finkelstein A. V., and Skolnick J. What is the probability of a chance prediction of a protein structure with an rmsd of 6 Å? *Fold Des.*, 3:141–147, 1998.

[76] Yang Zhang and Jeffrey Skolnick. Scoring Function for Automated Assessment of Protein Structure Template Quality. *Proteins*, 57:702–710, 2004.

[77] Siew N., Elofsson A., Rychlewski L., and Fischer D. MaxSub: An automated measure for the assessment of protein structure prediction quality. *Bioinformatics*, 16:776–785, 2000.

[78] Shindyalov I. N. and Bourne P. E. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Engineering*, 11:739–747, 1998.

[79] Taylor W. R. and Orengo C. A. Protein structure alignment. *J. Mol. Biol.*, 208:1–22, 1989.

[80] Gibrat J. F., Madej T., and Bryant S. H. Surprising similarities in structure comparison. *Curr. Opin. Struct. Biol.*, 6:377–385, 1996.

[81] Holm L. and Sander C. Protein structure comparison by alignment of distance matrices. *J. Mol. Biol.*, 233:123–138, 1993.

[82] D. T. Jones, K. Bryson, A. Coleman, L. J. McGuffin, M. I. Sadowski, J. S. Sodhi, and J. J. Ward. Prediction of Novel and Analogous Folds Using Fragment Assembly and Fold Recognition. *Proteins*, 61, Suppl 7:143–151, 2005.

[83] Ceslovas Venclovas and Mindaugas Margelevicius. Comparative Modeling in CASP6 Using Consensus Approach To Template Selection, Sequence-Structure Alignment, and Structure Assessment. *Proteins*, 61, Suppl 7:99–105, 2005.

[84] A. Sali and T. L. Blundell. Comparative protein modelling by satisfaction of spatial restraints. *J. Mol. Biol.*, 234:779–815, 1993.

[85] `http://trantor.bioc.columbia.edu/programs/jackal`.

[86] Canutescu A. A., Shelenkov A. A., and Dunbrack R. L. Jr. A graph theory algorithm for protein side-chain prediction. *Protein Science*, 12:2001–2014, 2003.

[87] McGuffin L. J. and Jones D. T. Improvement of the genTHREADER method for genomic fold recognition. *BioInformatics*, 19:874–881, 2003.

[88] Kelley L. A., MacCallum R. M., and Sternberg M. J. E. Enhanced Genome Annotation using Structural Profiles in the Program 3D-PSSM. *J. Mol. Biol.*, pages 499–520, 2000.

[89] Lukasz Jaroszewski, Leszek Rychlewski, Zhanwen Li, Weizhong Li, and Adam Godzik. FFAS03: a server for profile-profile sequence alignments. *Nucleic Acids Research*, 33:284–288, 2005.

[90] Leszek Rychlewski and Daniel Fischer. LiveBench-8: The large-scale, continuous assessment of automated protein structure prediction. *Protein Science*, 14:240–245, 2005.

[91] Chris S. Pettitt, Liam J. McGuffin, and David T. Jones. Improving sequence-based fold recognition by using 3D model quality assessment. *Bioinformatics*, 21:3509–3515, 2005.

[92] B. Wallner and A. Elofsson. Can correct protein models be identified? *Protein Science*, 12:1073–1086, 2003.

[93] Silvio C. E. Tosatto. The Victor/FRST Function for Model Quality Estimation. *Journal of Computational Biology*, 12:1316–1327, 2005.

[94] L. Holm and C. Sander. Evaluation of protein models by atomic solvation preferences. *J. Mol. Biol.*, 225:93–105, 1992.

[95] Philip Bradley, Lars Malmstom, Bin Qian, Jack Schonbrun, Dylan Chivian, David E. Kim, and David Baker. Free modeling with Rosetta in CASP6. *Proteins*, 61, Suppl 7:128–134, 2005.

[96] David Shortle, Kim T. Simons, and David Baker. Clustering of low-energy conformations near the native structures of small proteins. *Proc. Natl. Acad. Sci. USA*, 95:11158–11162, 1998.

[97] Yang Zhang and Jeffrey Skolnick. SPICKER : A Clustering Approach to Identify Near-Native Protein Folds. *Journal of Computational Chemistry*, 25:865–871, 2004.

[98] Ora Schueler-Furman, Chu Wang, Phil Bradley, Kira Misura, and David Baker. Progress in modeling of protein structures and interactions. *Science*, 310:638–642, 2005.

[99] R. D. King, D. A. Clark, J. Shirazi, and M. J. E. Sternberg. Inductive logic programming used to discover topological constraints in protein structures. *Second International Conference on Intelligent Systems for Molecular Biology*, pages 219–226, 1994.

[100] Ward J.J., McGuffin L.J., Buxton B.F., and Jones D.T. Secondary structure prediction with support vector machines. *BioInformatics*, 19:1650–1653, 2003.

[101] Fariselli P. and Casadio R. A neural network based predictor of residue contacts in proteins. *Protein Engineering*, 12:15–21, 1999.

[102] Fariselli P., Olmea O., Valencia A., and Casadio R. Prediction of contact maps with neural networks and correlated mutations. *Protein Engineering*, 14:835–843, 2001.

[103] Rafal Adamczak, Aleksey Porollo, and Jaroslaw Meller. Accurate Prediction of Solvent Accessibility Using Neural Networks-Based Regression. *Proteins*, 56:753–767, 2004.

[104] Jinfeng Liu and Burkhard Rost. Sequence-based prediction of protein domains. *Nucleic Acids Research*, 32(12):3522–3530, 2004.

[105] Jaehyun Sim, Seung-Yeon Kim, and Jooyoung Lee. PPRODO: Prediction of protein domain boundaries using neural networks. *Proteins*, 59:627–632, 2005.

[106] Rumelhart D. E. and McClelland J. L. *Parallel distributed processing: exploration in the microstructure of cognition.* Cambridge, MA: MIT Press, 1986.

[107] Christopher Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[108] Ning Qian and Terrence J. Sejnowski. Predicting the secondary structure of globular proteins using neural network models. *J. Mol. Biol.*, 202:865–884, 1988.

[109] B. Rost and C. Sander. Combining evolutionary information and neural networks to predict protein secondary structure. *Proteins*, 19:55–72, 1994.

[110] Marco Punta and Burkhard Rost. PROFcon: novel prediction of long-range contacts. *Bioinformatics*, 21(13):2960–1968, 2005.

[111] Robert Farber, Alan Lapedes, and Karl Sirotkin. Determination of eukaryotic protein coding regions using neural networks and information theory. *J. Mol. Biol.*, 226:471–479, 1992.

[112] Faramarz Valafar. Pattern recognition techniques in microarray data analysis: A survey. *Annals of the New York Academy of Sciences*, 980:41–64, 2002.

[113] Hu X., Maglia A., and Wunsch D. A general recurrent neural network approach to model genetic regulatory networks. *Conf Proc IEEE Eng Med Biol Soc*, 5:4735–4738, 2005.

[114] `http://www.mathworks.com`.

[115] Qiwen Dong, Xiaolong Wang, and Lei Lin. Novel knowledge-based mean force potential at the profile level. *BMC Bioinformatics*, 7:324, 2006.

[116] David T. Jones. Predicting novel protein folds by using FRAGFOLD. *Proteins*, 45, Suppl 5:127–132, 2001.

[117] Jerry Tsai, Richard Bonneau, Alexandre V. Morozov, Brian Kuhlman, Carol A. Rohl, and David Baker. An Improved Protein Decoy Set for Testing Energy Functions for Protein Structure Prediction. *Proteins*, 52:76–87, 2003.

[118] Jones D. T. and McGuffin L. J. Assembling novel protein folds from super-secondary structural fragments. *Proteins*, 53 Suppl 6:480–485, 2003.

[119] G. M. Clarke and D. Cooke. *A basic course in Statistics, Third Edition, Chapter 12*. Hodder Arnold, 1991.

[120] Samudrala R. and Levitt M. Decoys 'R' Us: a database of incorrect conformations to improve protein structure prediction. *Protein Science*, 7:1399–1401, 2000.

[121] `http://dd.compbio.washington.edu/`.

[122] Tanaka S. and Scheraga H. A. Medium- and long-range interaction parameters between amino acids for predicting three-dimensional structures of proteins. *Macromolecules*, 6:945–950, 1976.

[123] Miyazawa S. and Jernigan R. L. Estimation of effective interresidue contact energies from protein crystal structures: quasi-chemical approximation. *Macromolecules*, 18:534–552, 1985.

[124] Paul D. Thomas and Ken A. Dill. Statistical Potentials Extracted From Protein Structures: How Accurate Are They? *J. Mol. Biol.*, 257:457–469, 1996.

[125] Ram Samudrala and John Moult. An All-atom Distance-dependent Conditional Probability Discriminatory Function for Protein Structure Prediction. *J. Mol. Biol.*, 275:895–916, 1998.

[126] Hui Lu and Jeffrey Skolnick. A Distance-Dependent Atomic Knowledge-Based Potential for Improved Protein Structure Prediction. *Proteins*, 44:223–232, 2001.

[127] Buchete N-V, Straub J. E., and Thirumalai D. Anisotropic coarse-grained statistical potentials improve the ability to identify native-like protein structures. *J Chem Phys*, 118:7658–7671, 2003.

[128] Kam Y. J. Zhang and David Eisenberg. The three-dimensional profile method using residue preference as a continuous function of residue environment. *Protein Science*, 4:687–694, 1994.

[129] Marc Delarue and Patrice Koehl. Atomic Environment Energies in Proteins Defined from Statistics of Accessible and Contact Surface Areas. *J. Mol. Biol.*, 249:675–690, 1995.

[130] Gordon M. Crippen. Recognizing Protein Folds by Cluster Distance Geometry. *Proteins*, 60:82–89, 2005.

[131] Alexey G. Murzin, Steven E. Brenner, Tim Hubbard, and Cyrus Chothia. SCOP: A Structural Classification of Proteins Database for the Investigation of Sequences and Structures. *J. Mol. Biol.*, 247:536–540, 1995.

[132] Britt Park and Michael Levitt. Energy functions that Discriminate X-ray and Near-native Folds from Well-constructed Decoys. *J. Mol. Biol.*, 258:367–392, 1996.

[133] Samudrala R., Xia Y., Levitt M., and Huang E.S. A combined approach for ab initio construction of low resolution protein tertiary structures from sequence. *Proceedings of the Pacific Symposium on Biocomputing*, pages 505–516, 1999.

[134] Samudrala R. and Levitt M. A comprehensive analysis of 40 blind protein structure predictions. *BMC Structural Biology*, 2:3–18, 2002.

[135] Yu Xia, Enoch S. Huang, Michael Levitt, and Ram Samudrala. Ab initio construction of protein tertiary structures using a hierarchical approach. *J. Mol. Biol.*, 300:171–185, 2000.

[136] Chen Keasar and Michael Levitt. A novel approach to decoy set generation: Designing a physical energy function having local minima with native structure characteristics. *J. Mol. Biol.*, 329:159–174, 2003.

[137] Michael Levitt. Protein Folding by Restrained Energy Minimization and Molecular Dynamics. *J. Mol. Biol.*, 170:723–764, 1983.

[138] George D. Rose, Ari R. Geselowitz, Glenn J. Lesser, Richard H. Lee, and Micheal H. Zehfus. Hydrophobicity of Amino Acid Residues in Globular Proteins. *Science*, 229:834–838, 1985.

[139] B. Rost and C. Sander. Conservation and Prediction of Solvent Accessibility in Protein Families. *Proteins*, 20:216–226, 1994.

[140] Boris A. Reva, Jeffrey Skolnick, and Alexei V. Finkelstein. Averaging Interaction Energies Over Homologs Improves Protein Fold Recognition in Gapless Threading. *Proteins*, 35:353–359, 1999.