

**PREDICTION OF PROTEIN-LIGAND BINDING AFFINITY USING
NEURAL NETWORKS**

PAVANDIP SINGH WASAN

NATIONAL UNIVERSITY OF SINGAPORE

2004

Name: Pavandip Singh Wasan
Degree: Master of Science
Dept: Information Systems
Thesis Title: Prediction Of Protein-Ligand Binding Affinity Using Neural Networks

Abstract

A big problem in the life-sciences is the ability to calculate, *in-silico*, the binding affinity between a protein active site and a lead-ligand. This thesis introduces a new method to predict the binding affinity of a given drug ligand and active site, using backpropagation neural networks of 128 protein ligand complexes, with electrostatic, hydrogen bonding and molecular weight parameters. The parameters are given space and magnitude consideration, through the use of physico-chemical autocorrelation for the preparation of the input parameters. Self-Organizing Maps(SOM) are used as well to visualize the distribution of the input cases in similarity space. The results showed an improvement in accuracy over multiple regressive and the BLEEP method for calculation of binding affinity, using Root Mean Square, Relative Root Mean Square, Mean Absolute and Relative Mean Absolute Error calculations. The SOM additionally showed positive clustering of protein-ligand complexes, from similar families spread through the input space.

Keywords: Binding Affinity, Neural Networks, Backpropagation, Physico-Chemical Autocorrelation, Self Organizing Maps, Drug Design

**PREDICTION OF PROTEIN-LIGAND BINDING AFFINITY USING
NEURAL NETWORKS**

PAVANDIP SINGH WASAN

**(B. Sc. (Hons.) in Computing and Information Systems,
University of London)**

**A THESIS SUBMITTED
FOR THE DEGREE OF MASTER OF SCIENCE
DEPARTMENT OF INFORMATION SYSTEMS
SCHOOL OF COMPUTING
NATIONAL UNIVERSITY OF SINGAPORE**

2004

Acknowledgment

I am firstly most grateful to my supervisor, A/Prof Rudy Setiono for all his patience and support throughout the duration of this project. I am very thankful for his time, always ready to help me, even on no notice. Thank you Rudy.

I must as well express my most heartfelt gratitude towards all the scientific researchers at Lynk Biotechnologies, who were always willing to help me better understand the science behind this thesis, and for helping me plan my experiments.

Lastly, I would like to thank the people closest to me, my family and dearest friends for all their moral support and tolerance throughout this study period. It will not go forgotten.

Contents

Acknowledgements	I
Table of Contents	II
List of Figures	IV
List of Tables	V
Summary	VI
1 Introduction	1
1.1 Motivation	1
2 Literature Review and Related Work	4
2.1 Proteomics and Drug Design	4
2.2 Currently Used Computational Docking Methodologies	10
2.3 Neural Network Review	16
2.4 Applicability of Neural Networks to Drug Design	20
2.5 Coding Chemical Structures	21
2.6 Neural Networks in Structure Activity Relationships and Drug Design	28
3 Methodologies	38
3.1 Preparation of Interacting Molecules	39
3.2 Physico-chemical Autocorrelation	42
3.3 Preprocessing, Postprocessing and Normalization	45
3.4 Self-Organizing Maps	47
3.5 Feed-forward Backpropagation	50
3.5.1 Introduction to Backpropagation.....	50
3.6 Multiple Linear Regression Analysis	57
3.7 Error Calculation	58
4 Data Used	60
4.1 Data Relevance and Requirements	60
4.2 Data Selection	62

4.3	Input and Output Parameters	69
5	Results and Analysis	73
5.1	Self-Organizing Maps	73
5.1.1	Clustering Through Protein Active Site Similarity.....	75
5.1.2	Influence of Ligand Similarity on Clustering.....	77
5.1.3	Clustering Through Structural and Physico-chemical Similarity.....	83
5.2	Backpropagation Neural Networks	85
6	Conclusion and Future Work	92
6.1	Conclusion	92
6.2	Future Work	94
	References	i
	Appendix A	ix
	Appendix B	xv
	Appendix C	xxi
	Appendix D	xxi
	Appendix E	xxii

List Of Figures

Figure 2.1.1 : Haemoglobin molecule with oxygen bound at all four haems.....	Pg. 5
Figure 2.1.2 : A ligand bound within the active site of a macromolecule.....	Pg. 7
Figure 2.2.1 : The Glide Funnel.....	Pg. 13
Figure 2.3.1 : Kohonen Contour Map.....	Pg. 19
Figure 2.5.1 : Contribution of atoms No.1 and No. 2(at (r_1, φ_1) and (r_2, φ_2)) to the intensity s_i at interval i on the circle with radius R, shown as shaded areas of the corresponding Lorentzian bell-shape function).....	Pg. 26
Figure 2.6.1 : Reaction scheme on the production of xylene isomers(including relative distribution).....	Pg. 31
Figure 3.2.1 : 2D Structure of Water Molecule.....	Pg. 43
Figure 3.5.1 : Structure of a Feedforward backpropagation Neural Network.....	Pg. 51
Figure 3.5.2 : Schematic presentation of weight correction with backpropagation. W_x represents weights in layer x.....	Pg. 52
Figure 3.5.3 : Illustration of a general neuron within a backpropagation neural network.....	Pg. 53
Figure 3.5.4 : Tangent-Sigmoid(left) and Linear(right) Functions Used In the experiments..	Pg. 53
Figure 4.1.1 : 2-D Structure of benzamidine.....	Pg. 61
Figure 4.2.1 : An extract of the MOL2 file format used.....	Pg. 69
Figure 4.3.1 : An example of a U-matrix.....	Pg. 72
Figure 5.1.1 : Stabilized U-matrix of SOM at 3500 iterations.....	Pg. 74
Figure 5.1.2 : Common substructure found in group (k) ligands.....	Pg. 82
Figure 5.2.1 : Root Mean Squared Error and Mean Absolute Error vs No. of Hidden Neurons.....	Pg. 86
Figure 5.2.2 : Relative Root Mean Squared Error and Relative Mean Absolute Error vs No. of Hidden Neurons.....	Pg. 86

List of Tables

3.1.1 :	Atom Types Selected For Characterization with Hydrogen Bonding Characteristics.....	Pg. 40
3.2.1 :	Atom coordinates and charge for water molecule.....	Pg. 44
3.2.2 :	Parameters and respective ranges used for autocorrelation.....	Pg. 45
4.2.1 :	List of 128 Protein-Ligand Complexes Used in Experiments.....	Pg. 63
4.3.1 :	Parameters for characterization of ligands and active sites.....	Pg. 70
5.1.1 :	Breakdown of Protein-Ligand Complexes by cluster and majority protein.....	Pg. 75
5.1.2 :	Protein Families and common amino acid residues present.....	Pg. 76
5.1.3 :	Ligands for group (g).....	Pg. 78
5.1.4 :	Ligands for group (d).....	Pg. 79
5.1.5 :	Ligands for group (k).....	Pg. 80
5.1.6 :	Ligands for group (m).....	Pg. 81
5.1.7 :	Ligands for group (h).....	Pg. 83
5.2.1 :	Error calculated from backpropagation neural network with varying hidden neurons.....	Pg. 85

Thesis Summary

This thesis presents a new methodology to be used for predicting the binding affinity of ligands (drug leads) to protein active sites, using neural networks. A large part of healthcare is derived from the suitability of medication and as well its affordability. Medicines today do not come cheap due to the laborious process through which they are developed. These more traditional methods of drug development involve a large amount of potential drug leads being screened against the active sites (functional regions) within proteins which are believed to either excite or inhibit a particular physiological activity within our complex systems. This brute force mass screening not only introduces great waste in time and resources, but is not able to guarantee the successful outcome of a drug with suitable efficacy. As such, a more informed approach has been taken to design these drug leads – rational drug design.

Rational drug design involves the development of drugs based on the structural and physico-chemical characteristics held by these bioactive molecules, with the aim of identifying the pharmacophore, or set of complementary characteristics within the ligand and the active site, to produce a bind with high affinity and specificity. Two characteristics known to be vital to this interaction are electrostatic charge, and hydrogen bonding capacity. These very two factors are modeled in this thesis, with the aim of finding a good correlation with binding affinity. The adaptation of the physico-chemical information to its computable

representation is carried out by autocorrelation, a method that enables multiple properties of varied molecules, in terms of size, structure and chemical composition, to be represented by a fixed number of parameters, making it ideal for any statistical or machine learning approach.

Neural networks have chosen to be trained by a set of 128 protein-ligand complexes with known binding affinity. Before supervised training is carried out, the protein-ligand complexes are clustered, based on their modeled characteristics, by Kohonen Self Organizing Maps(SOM). SOMs make visible the spread of physico-chemical and structural diversity allowing any bias to be identified before the supervised training is started, and as well complements analysis of supervised training results. Once seen to be fairly spread out across the input space of the SOM, the backpropagation algorithm is used to train the network towards increasing its predictability of binding affinity being given a protein active site-ligand complex as input.

A range of tests were carried out to identify the best possible training topology of the neural network and once secured, comparisons were made to understand the relative strength of the method developed. Comparisons with Multiple Linear Regression and a previously published method of Biomolecular Ligand Energy Evaluation Protocol (BLEEP) were made and the developed method produced higher binding prediction accuracies than both methods. Further analysis into the clustering of the complexes alongside the supervised training highlighted

additional factors that could potentially improve binding results even more. Further research into these improvements is thus highly anticipated and expected to bring new light into the field of drug design.

Chapter 1. Introduction

This chapter is intended to introduce the reader to the arena of drug design and development, and the importance of the research undertaken in a wider perspective. The motivation behind the research undertaken will as such be described and followed by the organization and scope of this report.

1.1 Motivation

Drugs work with our biological systems through their interaction with receptors causing alterations in their activities to bring about biochemical changes within our bodies. These interactions can be agonistic, where the activity of the receptor is stimulated or antagonistic, where the activity is retarded.

Discovering and developing an effective drug is by no means an easy task. Many drugs we use today have been discovered by chance observation, second-hand analysis of traditional remedies or by taking note of the side effects of already developed drugs, and manipulating them to elevate the desired effects. A more systematic means of discovering drugs has been developed, that is, through combinatorial chemistry. Combinatorial chemistry involves large libraries of test compounds being screened against potential drug targets and their interactions studied. This trial and error methodology is understood to be a time-consuming and expensive method, requiring an inefficiently large amount of time and chemical resources.

A more organized approach to discovering drugs is known as rational drug design. As its name implies, it is a more systematic method of designing drugs

which rather than through brute force methods, uses information (such as the three dimensional structure and physico-chemical properties) inherent within the target receptor and that of potential ligands (drug compounds) that might bind with these receptors, to identify feasible candidate drug compounds. This more informed methodology was developed to reduce the number of candidates eventually being tested *in vivo* and *in vitro* (in the wet-lab) and reduce waste in the process. Several drugs have already been developed through rational drug design. Among them are Relenza for influenza, Ritonivir and Indinavir for HIV infections and as well as Viagra for the treatment of sexual dysfunction.

Many computational techniques have been developed to support this methodology, from the analysis and comparison of the protein sequences (to find homologous regions that could potentially reflect the actual active-sites within the protein structures that the candidate compound will interact with), through the prediction of protein structures from their sequential information, all the way to the predictive calculation of the binding affinity between the candidate ligands and the target protein receptors (active sites).

The motivation of this paper is to develop a methodology for predicting the binding affinity between candidate ligands and the active sites of the target protein molecules using neural networks. This will enable bench scientists to actually run *dry* experiments before taking on the much larger and longer task of synthesizing the drug compounds. A neural network is a computational model that uses concepts from that of the central nervous system to solve

computational problems involving association, classification, transformation and modeling [Zupan et al., 1999].

By using neural networks, this research therefore aims to develop a method to calculate how well a given drug compound binds to a target receptor *in silico* (computationally). This will enable better predictions and ranking of feasible ligands to be performed, reducing waste of biochemical compounds in the wet-lab. Current computational methods of predicting binding affinity have not yet been able to provide sure-fire results due to the multidimensional complexity of the molecular interactions involving a large number of parameters, some of which are not yet even be known. Thus, to deal with this complexity, the neural network designed will be based on results obtained directly from wet lab experiments. This will not only provide a means of predicting binding affinity based on real-life interactions but will as well better provide insight into the discovery of parameters that contribute more to the binding.

Chapter 2. Literature Review and Related Work

In this chapter, the field of proteomics and drug design will first be introduced including methods, technologies and the terminology used in the field. The transition of wet-lab to dry-lab will then be discussed with reference to the current computational methods used to approach these drug design challenges. The technology of neural networks will then be described along with how it has been used in various related life science problems, and the challenges its application poses. Finally, special focus will be made on the use of these neural network technologies to Quantitative Structure Activity Relationships (QSAR) problems, such as the one approached by this thesis.

2.1 Proteomics and Drug Design

The post-genomic era has brought about a whole new set of challenges, increasingly and especially so in the field of proteomics. The design of pharmaceutical leads as is an extremely complex process and is up to this day, not yet completely understood [Balbes et *al.* (1994)]. A great amount of computational effort has been put into the study of protein sequences, their relative homology, the prediction of their three dimensional conformations, the identification of sites of interaction within these complex 3D structures, and the design of suitable small molecular structures as therapeutic drugs with appropriate structural and physico-chemical constitution to bind to these macromolecules with high specificity so as to provide high efficacy with little, or ideally, no side effect.

The currently more traditional method of drug development, using mass screening of large combinatorial libraries against target protein structures, even with their evolutionary increase in speed, incur high costs and waste through their brute force methodology of blindly ‘attacking’ proteins with millions of ligand analogues. The basis of rational drug design is that drug activity arises through the molecular binding of a small molecule, or ligand, to a receptor or active-site of a larger molecule, which is usually a protein [Finn et al. (1999)]. In their bound state, the protein-ligand complex exhibits some biological activity, activated through their structural and chemical complementarity, both of which are vital for drug activity [Lengauer (1993), Finn et al. (1999)]. By binding to the active-site of the macromolecule, the designed ligand can either play an inhibitory (antagonist) or excitatory (agonist) role by replacing the activity of the macromolecules complex with its natural substrate with that of one created. Figure 2.1.1 below shows an example of a such a complex, of haemoglobin with oxygen bound at all four haems [Bernstein et al. (1977)].

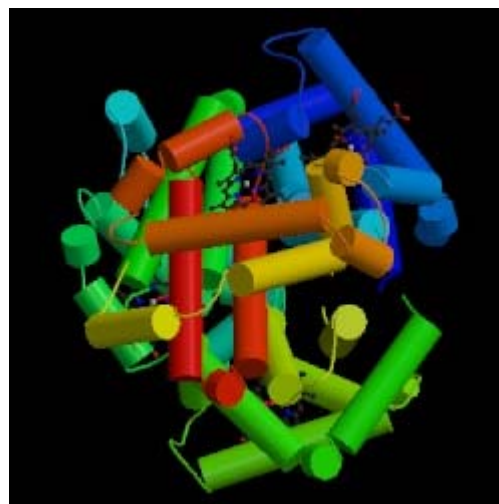


Figure 2.1.1 – Haemoglobin molecule with oxygen bound at all four haems.

Due to the non-static, constantly changing conformation of molecules, their modeling becomes a complex task. This not only involves the varying degrees

of movement of the small ligand alone, but that of the macromolecule as well, with particular focus on its active site. To increase the complexity further, the properties and composition of the molecules' solvent environment also need to be taken into consideration. This makes the exact simulation or modeling of the whole binding process, that of determining the molecular complex with the lowest energy, or most stable state, a huge feat. Biochemists, medicinal chemists and physicists together all work at increasing the accuracy of these molecular models through their energetic studies involving quantum physics and chemistry, and using technologies such as Nuclear Magnetic Resonance (NMR) Spectroscopy and X-Ray Crystallography, they are even able to attain the three dimensional structure of the molecules. The structures obtained from such processes are however just a snapshot of these molecules in motion and therefore are still not able to tackle the complexity of intermolecular binding. To deal with such complexity, studies are therefore made within certain limits of assumption determined by the complexity and flexibility of the molecules themselves.

A common term used to describe the computational binding of a ligand to its best matched active site within a macromolecule is 'docking' [Halperin et al. (2002)]. Halperin et al. (2002) discussed the two main challenges in docking, namely unbound and bound docking. Unbound docking is generally the greater challenge of the two types as it involves fitting optimally an unbound ligand to a receptor macromolecule's active site(s) to form a complex in its lowest composite energy level, or most stable state. This challenge is better known as the 'docking problem' [Finn et al. (1999)]. Algorithms written to solve this problem try to achieve one or both of two goals. The first one is that of enabling

the researcher to study the detailed interaction of the ligand with the microstructure of the active site. The second goal is that of predicting the 'wellness of fit', of the ligand to the active site enabling the researcher to rank a library of ligands according to how well they fit, and help chemists filter out the less likely leads to save resources on less informed biochemical synthesis. The docking problem presents yet another challenge, that of the identification of the active site, in terms of where on the macromolecule it resides, and what amino acid composition it has, in the case of proteins for example.

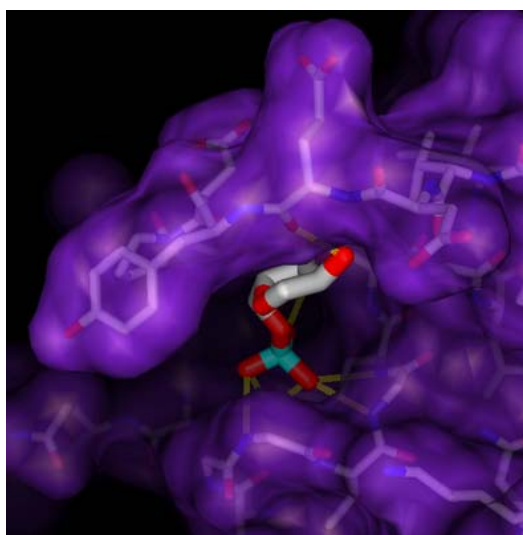


Figure 2.1.2 : A ligand bound within the active site of a macromolecule

To begin, researchers have used 'bound docking', mentioned earlier to gain more knowledge on these molecular interactions. Bound docking on its own is a much simpler problem as in this case, the location of the active site is known [Halperin et *al.* (2002)]. In bound docking, the location of the active site is made directly visible through wet-lab experimental means where the three dimensional structure of a protein-ligand complex is obtained through either NMR Spectroscopy or X-ray crystallography. The goal of this method is to extract these ligands from the complex *in-silico* and study the characteristics

within both interacting members of the complex individually and to establish their geometric and physico-chemical complementarity which will then help determine the vital factors involved in that particular binding. It is therefore valuable to find complexes of the same particular specific active site, with that of many different experimentally bound ligands. This enables the researcher to study the 'pharmacophore' within the ligands [Finn et al. (1999)]. Finn et al. (1999) describe the pharmacophore as the set of features present in a specific three dimensional configuration, regardless of its conformation. The pharmacophore is intended to reveal a template present in all reacting ligands to a specific site, for that specific site, that present the essential constituents that are required for a reaction to take place. The remainder of the molecule, not part of the pharmacophore thus acts merely as its scaffold, holding it in place. To exemplify this, Glen et al. (1995) used these principles to help in the discovery of a drug for migraine, 311C90(6), by identifying the interacting pharmacophore to comprise a protonated amine site, an aromatic site, a hydrophobic pocket, and two hydrogen bonding sites. It can be inferred from this that the geometry of a potentially binding ligand to be bound is of utmost importance, and has been the grounds for the development of several docking algorithms [Kuntz et al. (1982), Connolly (1983), Lee et al. (1985), DesJarlais et al. (1986)].

How is this geometrical and spatial data then made useful? A correlative study is required to link up the structure of the ligand/active-site complex to its function. These studies are known as Structure Activity Relationship (SAR) or Structure Property Relationship (SPR) studies. The functions proteins hold are often uncovered through the study of their evolutionary history, visible through

sequence similarity, or homology [Lichtarge et al. (1996), Marcotte et al. (1999)]. Predictive methods, as well, have been developed for the identification of potential active sites within macromolecules. These methods either use comparative similarities amongst proteins with similar function, surface searches for geometric cleavages in the macromolecular structure [Laskowski et al. (1996)], searches through simulated docking [Oshiro et al. (1998)] of ligand libraries [Chen et al. (2001)], or through the study of chemical and electrostatic properties throughout the protein [Shehadi (2003)].

Due to the mobility of proteins and therefore their active sites *in vivo*, it is difficult to 'capture' the actual molecular conformation of either the active site required for a successful bind or predict the best conformation of a ligand for it to bind to a particular active site. This is due to the varying degrees of freedom each chemical bond holds. Therefore, to simulate a docking between the active site and a ligand, the varying flexibilities tend to determine the algorithms used [Fraga et al. (1995)]. Fraga et al. (1995) has classified docking into three categories, according to their degrees of flexibility, rigid body docking, semi-flexible docking, and flexible docking. In rigid body docking, both the molecules are considered to have a fixed conformation, while in semi-flexible docking, one of the molecules, more often than not the smaller, is considered flexible while the active site is taken to be rigid. In flexible docking, both molecules are considered flexible but only to a pre-defined extent to simplify the complexity of the problem. Among these three methods, the first is the most simplistic and may not provide accurate predictions on the wellness of fit amongst the ligand and macromolecule. Therefore, it is desirable to have at least one of the two molecules, usually the ligand as a flexible molecule to allow the study of the fit

of the same molecule in its various conformations. This wellness of fit is usually measured in terms of energy, and the goal of a good fit would be one which delivers the lowest energy level [Lengauer et al. (1996)].

2.2 Currently Used Computational Docking Methodologies

Many docking algorithms can be thought of to act as search functions, searching for the optimum conformation (the actually docked state), of both the ligand and the active site, within the limitations of their conformational flexibility. Such a search algorithm, may however produce an impractically large number of solutions. In theory, zeroing down on the best solution using free-energy simulations is reliable [Pearlman et al. (2001)], but impractical due to the computational time involved. As such, the use of structure, and not energy, based methods have been vastly used in drug design enabling the prediction of suitably binding compounds. The six well known docking programs that shall be discussed here are FlexX [Rarey et al., (1996)], DOCK [Kuntz et al., (1982)], GOLD [Jones et al., (1997)], Glide [Eldridge et al., (1997)], Ligand-Fit [Kontoyianni et al. (2004)], and BLEEP [Nobeli et al., (2001)].

FlexX [Rarey et al. (1996)] uses an incremental construction algorithm, combining physico-chemical interactions as well as geometric conformational sampling to find the optimum binding conformation of the protein-ligand complex, and predict the binding affinity. It is used when the three dimensional structures of the proteins are known and a single or a library of ligands is available for docking. FlexX attempts to predict the complex conformation, which is useful when the protein-ligand complex has not been found through

experimental means. Its mass virtual screening abilities comes in useful when ranking of a ligand library is needed before proceeding to further wet-lab experimental synthesis. FlexX works by first placing a fragment into a pre-defined active site of the protein followed by a tree search algorithm based on a greedy strategy to incrementally grow the first fragment to its final optimal conformation. This is similar to the algorithm used by Leach and Kuntz (1992). Adaptations from LUDI [Bohm (1992a/b)] are then used to model the protein-ligand interactions, using geometric pairwise assignments based on physico-chemical complementarity. To deal with conformational flexibility (of the ligand, as the active site is considered rigid in FlexX), the same method as used in MIMUMBA [Klebe et al. (1994)], a conformational search program, is used. Pose clustering [Linnainmaa et al. (1988)], a pattern recognition technique is first used for the placement of the first (base) fragment. Once the first fragment has been placed, fragments are added to it in all possible conformations, and the k best choices are then taken to the next similar iteration, to build a ligand to its eventual full structure.

DOCK [Kuntz et al. (1982)] uses shape based algorithms to run its protein-ligand binding. In DOCK, as in FlexX, a three dimensional macromolecular structure is required with its active site defined and a single or library of ligands to be either bound optimally (predictively), or ranked according to their wellness of fit. In DOCK, a Connolly (1983) molecular surface of the active site is first generated. The cleavage shape presented by the Connolly surface is then used to define spheres within the 'pocket'. The centre of each of these spheres is now taken as potential locations for atoms of the ligand to be docked. The ligands presented to the program are then geometrically manipulated for their

atomic positions to match the centre of these spheres, determining all possible conformations of the ligand within the active site. Each conformation is then scored, using one of three scoring strategies, namely shape scoring, which uses a Lennard-Jones (1932) potential approximation, electrostatic scoring using DELPHI [Rocchia et al. (2001)] to calculate the electrostatic potential, and Force-field scoring which uses AMBER [Pearlman et al. (1995)] force fields.

GOLD [Jones et al. (1997)] (Genetic Optimization for Ligand Docking) is yet another automated ligand docking software based on an algorithm by Jones et al. (1995) which this time uses genetic algorithms to explore the full range of ligand conformations, and flexibility of the molecules comprising the active site. This flexibility within the active site is however limited to that of the side chains of amino acids within it. Scoring then ranks the ligands in their respective conformations taking into consideration hydrogen bonding, a pairwise dispersion potential to describe hydrophobicity contributions, and molecular mechanics for the internal energetic representation of the ligand. Good results from the genetic algorithm are therefore likely to produce protein-ligand complexes with maximal interactions at hydrogen bonding sites between the respective hydrogen donors, acceptors and acceptor/donors, as well as burial of hydrophobic surfaces.

Glide [Eldridge et al., (1997)] uses its own algorithm for conformational generation allowing efficient systematic searches within the ligand conformational space by hierarchically filtering out undesirable conformations leaving fewer combinations to compute. This is illustrated in Figure 2.2.1 below

from <http://www.schrodinger.com/Products/ glide.html>. It clusters the core regions of the generated ligand conformations, treating the end groups independently. Optimal binding conformations are then identified using a combination of Monte-Carlo sampling and minimization of the ligand within the active site.

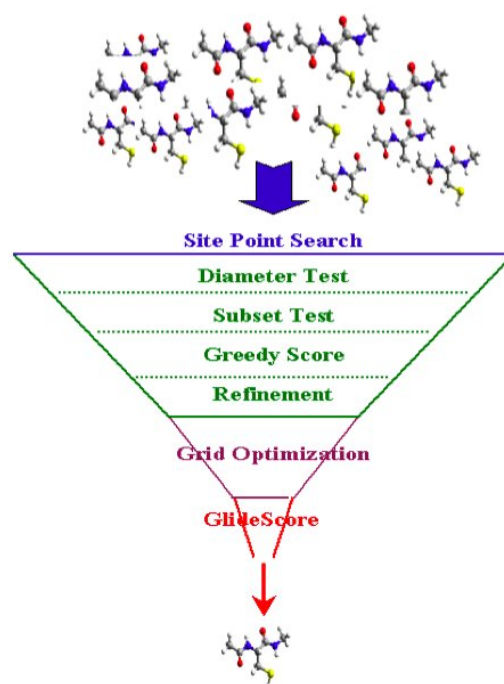


Figure 2.2.1 – The Glide Funnel

This system, similar to DOCK and FlexX above performs binding between a rigid active site and flexible ligands allowing both the identification of a ligand in its predicted optimal complex conformation, as well as a ranking of a ligand library according to binding affinity, in this case using a scoring strategy involving grid-based energy minimization, Monte-Carlo sampling, and a modified ChemScore [Eldridge et al. (1997)], known to the application as GlideScore.

The next docking system that will be discussed in this chapter is that of LigandFit [Venkatachalam et al. (2003)], a shape based methodology used to dock ligands into protein active sites. In LigandFit, the active sites of the protein need not be known before the docking is carried out. It is able to detect invaginations within the protein structure surface using a flood-fill algorithm [Foley et al. (1982), Rogers (1985)] representing possible sites of interaction. LigandFit as well allows the determination and extraction of a site from a pre-bound three dimensional complex for manipulation of the site to more accurately simulate its dynamics *in vivo* or as well to dock alternative ligands to those from the complex. LigandFit's docking procedure then employs stochastic selection of the ligands variable torsional angles as a means of conformational searching, selection of a particular conformation based on shape matching with the active site, and a predictive binding affinity calculation using a grid-based energy calculation to estimate interaction energies within the docked complex.

The Biomolecular Ligand Energy Evaluation Protocol (BLEEP) [Nobeli et al., (2001)], is yet another methodology that has been developed to predict protein-ligand interactions, in this case through potentials of mean force (PMF) [Muegge et al., 1999]. BLEEP does this by considering atom distances between 2.5Å and 8 Å between proteins and ligands and converting them into pair potential functions. The atoms of the protein and ligand are respectively first typed, using the Simple Atom Type Information System (SATIS) [Mitchell et al., (1999)]. Each atom within the protein-ligand system is then assigned a ten digit code, the first two digits representing the atoms atomic number, and the remaining eight consisting of the two digit atomic numbers of the atoms

covalently bonded to it. To take account of the important hydrogen bonding parameter [Jones et al.,1997], polar hydrogens were added as interaction sites, their coordinates calculated by HBPLUS [McDonald, 1994]. In addition to that, to account for interactions with water particles, missing water particles were added using AQUARIUS2 [Pitt, et al., 1993, Goodfellow et al., 1995]. Once done, BLEEP then uses thermodynamics to convert these typed distance distributed atoms into pair potentials. This thermodynamic equation used is as follows,

$$\Delta E^{ab}(s) = kT \ln[1 + m^{ab} \sigma] - kT \ln \left[1 + m^{ab} \sigma \left\{ \frac{f^{ab}(s)}{f(s)} \right\} \right] \quad [\text{Eq.2.2.1}]$$

where $\Delta E^{ab}(s)$ is the net potential within a pair comprising atom types a and b at distance s, k is the Boltzman constant, T is the absolute temperature, m^{ab} is the total number of contacts between atom types a and b, $f^{ab}(s)$ is the distance distribution between atom types a and b at distance s, σ is a weighting function and $f(s)$ is the reference potential, derived from the average of the atom-atom distances for the entire dataset. The overall interaction is then calculated by summing up all PMF scores between all the atom pairs within the protein-ligand complex.

The six methods described above have exemplified how current popular applications have adopted a combination of geometric, energetic and physico-chemical complementarity to find optimal binding conformations and to predict binding affinity. The following section will discuss how machine learning methodologies, in particular that of neural networks, work and further how they

have been used in chemistry and drug design, and the potential for such techniques to be further used in the prediction of binding affinity and conformation.

2.3 Neural Network Review

The motivation towards the development of neural networks has been to mimic the information processing capabilities of the brain, a completely different means of processing when compared to that of the traditional von Neumann architecture. Neural networks are used today mostly where complex data needs to be processed for the sieving of useful information from it, with applications ranging from stock market analysis and predictions, to biometric fingerprint pattern recognition and medical diagnoses. Neural networks are commonly used to approach challenges of the following types [Gasteiger et *al.*, 1993]:

- Classification: This is where an object with several characteristics or parameters, is assigned to one of many predetermined categories.
- Modeling: This is where an analytical function is derived from the correlation of a set of inputs to a set of outputs of the network. This is especially useful in cases where input and output data to a process is available but no mathematical function is available to correlate the two.
- Association: This type of problem can be divided into auto-association and hetero-associative categories. In auto-association, patterns learned by the

network can be reproduced given the incomplete patterns as inputs, a common application being character recognition in handwriting. Hetero-association involves the one-to-one correlation of two discrete sets of patterns that need not have any correlative similarity.

- Mapping: This is where a transformation of dimensionality from a higher to a lower level, or vice versa takes place, an example of this being the property mapping of a three dimensional object to a two dimensional plane.

The main neural network strategies adopted to tackle such problems are that of back-propagation, counter-propagation, and Kohonen networks. The back propagation [Werbos (1982), Rumelhart et al. (1986)] strategy of neural networks is one which involves at least three layers of nodes (neurons), an input layer, one or more hidden layers and an output layer.

Back propagation networks use supervised learning methods. This means that the output must be known for each set of input data. The network first has its edges initialized with weights. The data is then passed through the network and the transfer functions within the nodes, and the output layer calculates the error, which is the difference between the output of the network and its intended response. The error is then propagated backwards through the network, and its weights adjusted using the Widrow-Hoff [Widrow et al. (1960)] delta learning rule to decrease the error the next time the same inputs are presented to the network. The correction of weights can either be done immediately after each individual input, after the error is detected (interactive), or as a batch using the accumulated errors from each training iteration.

Kohonen networks or maps [Kohonen (1982)], are yet another neural network strategy, this time aimed at preserving the topology of a multidimensional representation within a one or two-dimensional array of neurons. Kohonen networks are a means of unsupervised learning in which the algorithms involved identify clusters in the data they are subjected to. Such an unsupervised learning methodology enable the grouping of data according to the closeness of their parameters relative to one another in an n-dimensional space (where n is the number of parameters or variables imposed onto the data). Each neuron in a Kohonen network has a set of weights with which it is associated, each one corresponding to one of the data inputs. Applying a set of data to a Kohonen network thus involves the calculation of an activation level at each neuron. This activation level is represented by the Euclidean distance between the input vector and the weight vector at each neuron or mathematically represented as:

$$\text{Activation Level} = \sqrt{\sum_{i=0}^n (\text{weight}_i - \text{input}_i)^2} \quad [\text{Eq.2.2.2}]$$

A neuron whose weight vector is thus ‘close’ to that of an input vector would have a low activation level and conversely, vector pairs with higher Euclidean distance will have a higher activation level. For each input vector presentation, the neuron with the smallest activation level takes the title “winner” of that iteration. During the training process, input vectors are introduced to the network and at each cycle as a winner arises, the winner along with a predefined group of neurons around it (in its neighborhood, which may change throughout the training), have their weight vectors adjusted to more closely

match the input vector presented. The size of the neighborhood is usually decreased linearly as the training proceeds till eventually, the only neuron having its weights adjusted is the winner. The weight vector alteration depends on a factor known as the learning rate, each weight in the weight vector is adjusted according to the following equation:

$$\delta w_i = -\alpha(w_i - i_i) \quad [\text{Eq.2.2.3}]$$

where α is the learning rate and δw_i is the weight change. This learning rule is meant to distribute the neurons evenly throughout the n-dimensional space [Hecht-Nielsen, 1990; Hertz *et al.*, 1991; Kohonen, 1989]. With iteration of this learning algorithm, the input patterns which ‘trigger’ the same winning node are therefore said to belong to the same cluster or group. Lines can then be drawn to enclose the different groups to attain a contour like map, similar to the one below in Figure 2.1.2 [Zupan *et al.*, (1999)].

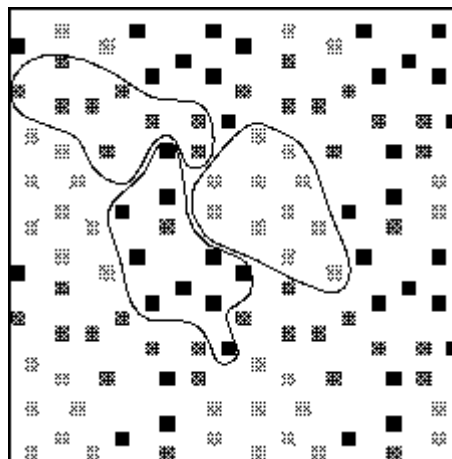


Figure 2.3.1 : Kohonen Contour Map

2.4 Applicability of Neural Networks to Drug Design

Drug design is vastly considered as a challenge involving the linking of structural and physico-chemical as well as energetic characteristics of molecules to their complementary reactivity. Neural networks have to date, been used vastly in the field of quantitative structure activity relationships (QSAR), a field introduced in the 1960's by Hansch and his co-workers [Hansch (1969), Martin (1978)]. These researchers were able to demonstrate that the biological activity taken on by chemical compounds has a direct mathematical correlation to their physico-chemical characteristics such as molecular weight, lipophilic potential, as well its electrostatic properties [Andrea *et al.* (1991)]. The modeling of such ideas is carried out through the mapping of a biological activity, A , to linear or parabolic functions of its physico-chemical properties (X, Y, \dots) [Andrea *et al.* (1991)] in the form

$$A = C_0 + C_1X + C_2X^2 + C_3Y + C_4Y^2 + \dots \quad [\text{Eq.2.4.1}]$$

and by using multiple linear regression to determine the values of C_0, C_1, \dots , which then helps to minimize the variance between the data and the model. Due to the non-linear feature extraction capability of neural networks, it has become a potential candidate to help solve QSAR problems. Amongst the different types of neural networks that exist, the one used most commonly is the back-propagation network [Zupan *et al.* (1991)]. Andrea *et al.* (1991) for instance have used back propagation neural networks to link the inhibitory activity of 256 2,4-diamino-6,6-dimethyl-5-phenyldihydrotriazines to dihydrofolate reductase (DFHR), by modeling the physico-chemical properties

of its ortho, meta and para positions of its phenyl rings, particularly their free energy and hydrophobicity. Similar functional group substitution methods were carried out by Aoyama et al. (1990) to study SAR in mitomycins and arylacryloylpiperazines with favorable results when compared to those obtained via the Adaptive Least Square (ALS) method [Moriguchi (1986)]. Before delving further into the applicability of Neural Networks in the various fields of drug design and QSAR, the representation and coding of input data for these neural networks will be described.

2.5 Coding chemical structures

The most important factor to the successful implementation of a neural network is the proper representation of the data used in it. Using neural networks for QSAR and drug design purposes, we thus need to represent the data suitably to allow correlations to be made between the structural, chemical and biological properties [Zupan et al., (1999)]. Of utmost importance is the representation of molecular data. Many representations of such data exist, such as 2-D, and 3-D representation of molecules in various formats, such as PDB, MOL, and mmCIF to name a few [Baxevanis et al., (1998)]. These formats further allow manipulation of the visual representation of molecular data in ball-and-stick forms, anti-aliased forms and spaced filled forms. At its simplest, the representation of a molecule takes on a graphs format with nodes (atoms), and edges (bonds). Such representations provide the user with topographical information of the constitution of a molecule. Most applications of such chemical data usually require more substantial information from the molecular

structure. This is usually done through the coding of the atomic coordinates, and in some cases, the bond data as well. In such cases, the larger the molecules, the larger the representations. However, in order to use molecular data for statistical, pattern recognition and machine learning methods such as neural networks, the molecules need to be represented by a fixed number of parameters, irrespective of their size [Zupan et al, (1999)]. Zupan et al. further mention that such a structure representation should satisfy four conditions,

- i) Uniqueness – Each compound should have only one code to uniquely distinguish it from other molecules
- ii) Uniformity – Each compound should be represented by the same number and type of parameters
- iii) Reversibility – The molecular structure should be able to be retrieved from the representation
- iv) Translational and Rotational Invariance – The representation should remain unchanged for translated and rotated structures

Three methods that aim to meet this goal of representative uniformity will now be discussed, one using an autocorrelation descriptor [Moreau et al., 1980], another using 3d-MoRSE (3D Molecule Representation of Structures based on Electron diffraction) [Schuur et al., 1996], and the final one using an infra spectral representation [Hemmer et al., 1999].

The autocorrelation descriptor represents a molecular structure as a graph and the physico-chemical properties, p_x , its atoms hold, for example, electrostatic charge, as real values assigned to the vertices of the graph. The descriptor is

then used by correlating this property of a particular atom i , $p_x(i)$, with the same property of another atom j , $p_x(j)$. These two values are then multiplied and summed up over all atom pairs within a predefined topological distance, d . This gives us the following function adapted from [Zupan et al., (1999)],

$$A(d) = \sum_{j=i+1}^n \sum_{i=1}^{n-1} \delta_{ij} p_x(i) p_x(j) \quad [\text{Eq.2.5.1}]$$

where δ_{ij} if $d_{ij}=d$, otherwise $\delta_{ij} = 0$. This distance is usually calculated in bond terms, that is, if $d=3$, the atoms under consideration within the molecular structure will have three bonds between them. As molecular graphs are likely to have different maximum distances, the value $A(d)$ is usually calculated for a range of values, $d \leq d^*$ to obtain a vector representation such as $(A(1), A(2), \dots, A(d^*))$, where typical values of d^* are 8 or 10 [Hollas, 2002]. Such autocorrelation descriptors have been used successfully with neural networks to predict the biodegradability of organic chemicals [Devillers et al., 1996]. In turn, Bauknecht et al. [1996] have used such autocorrelation descriptors to code partial atomic charges, electronegativity and polarizability from molecules. These representative vectors were then used with self-organizing neural network maps to distinguish dopamine agonists from benzodiazepine agonists, and thus enabling biological characterization of new potential leads to be carried out.

Another method of characterizing molecular structures of varying size by a fixed number of values has been introduced by Schuur et al. (1996), using a molecular transform derived from electron diffraction studies, called 3D-MoRSE

(Molecule Representation of Structures based on Electron diffraction). This work was based mainly on earlier electron diffraction study by Soltzberg and Wilkins (1977) for transforming three-dimensional atomic coordinates into a molecular code through the modification of an equation used in electron diffraction studies as follows,

$$G(\vec{S}) = \sum_{i=1}^N f_i \exp(2\pi\vec{r}_i \cdot \vec{S}) \quad [\text{Eq.2.5.2}]$$

where \vec{S} = the scattering in various directions, by N atoms at points \vec{r}_i , and f_i represents the form factors. This equation [2.5.2], is usually used in diffraction studies in the form in Eq. 2.5.3 [Wierl, 1931]

$$I(s) = K \sum_{i=2}^N \sum_{j=1}^{i-1} f_i f_j \int_0^{\infty} P_{ij}(r) \frac{\sin sr}{sr} dr \quad [\text{Eq.2.5.3}]$$

where $I(s)$ is the intensity of scattered radiation, r represents the interatomic distances, $P_{ij}(r)$ is the probability distribution of the vibrational variation between atoms i and j with f_i and f_j being their respective form factors, and K contains the instrument dependent constants. s here represents the scattering angle through the formula in Eq. 2.4.4 [Wierl, 1931],

$$s = 4\pi \sin(\vartheta / 2) / \lambda \quad [\text{Eq.2.5.4}]$$

with λ being the wavelength and ϑ being the scattering angle. Schuur et al. (1996) further made the assumptions that all molecules were rigid and atoms within them were point scatterers. Additionally, atomic parameters were used in place of the form factors, represented by A_i , leading to the final equation,

$$I(s) = \sum_{i=2}^N \sum_{j=1}^{i-1} A_i A_j \frac{\sin sr_{ij}}{sr_{ij}} \quad [\text{Eq.2.5.5}]$$

By now calculating $I(s)$ over a range of values of s and taking them as a vector, this vector can now be used to represent the molecular structure. Schuur et al. (1996) in their experiments took 32 values, ranging s from 0 to 31.0 \AA^{-1} , and used these vectors with counterpropagation neural networks to distinguish D1 dopamine agonists from D2 dopamine agonists, and as well to rank steroids binding to the corticosteroid binding globulin receptor into 3 categories according to their activity.

The final method of molecular representation that will be discussed is based on a representation of a 3-dimensional molecular structure by a unique vector with n elements despite the size of the structure, by projecting the molecules constituent atoms onto three perpendicular equatorial trajectories on an imaginary sphere large enough to accommodate the molecule [Zupan et al., 1997]. In order to convert the representation of the 3-dimensional molecule of N atoms, each represented by a $[x_j, y_j, z_j]$ triplet, the z_j , the y_j and x_j coordinates are set to 0, in turn. This enables a projection of (x, y) , (x, z) , and (y, z) -planar molecules to be made on the respective circles defined by the cross section of the sphere and the respective planes. A molecule with N atoms would thus give three sets of N pairs, $(x_1, y_1, x_2, y_2, \dots, x_n, y_n)$, $(x_1, z_1, x_2, z_2, \dots, x_n, z_n)$ and $(y_1, z_1, y_2, z_2, \dots, y_n, z_n)$. The problem of translation invariance is solved by adjusting the coordinates such that the origin of the coordinate system is set to the centre of mass of the molecule. The radius of the sphere chosen is arbitrary as long as it is larger or equal to the distance between the atoms centre of mass and

furthest atom. The representation is then converted to one independent of the number of atoms in the molecule, $S = (s_1, s_2, \dots, s_n)$ containing $3n$ variables, n for each plane. Each element of the vector S is defined as the cumulative intensity, s_i , at a predefined finite interval i , on the circle with arbitrary radius R , as illustrated in Figure 2.4.1, and is calculated as follows,

$$s_i = \sum_{j=1}^N I(i, r_j, \varphi_j, \sigma_j) = \sum_{j=1}^N \frac{r_j}{(\varphi_i - \varphi_j)^2 + \sigma_j^2} \quad [\text{Eq.2.5.6}]$$

for $i=1, \dots, n$. Any bell shaped function can be used to measure the intensity, $I(i, r_j, \varphi_j, \sigma_j)$. Figure 2.5.1 for instance describes the projection using a Lorentzian shape. The cumulative intensity s_i is a sum of N contributions of $I(i, r_j, \varphi_j)$ from each atom j in the molecule.

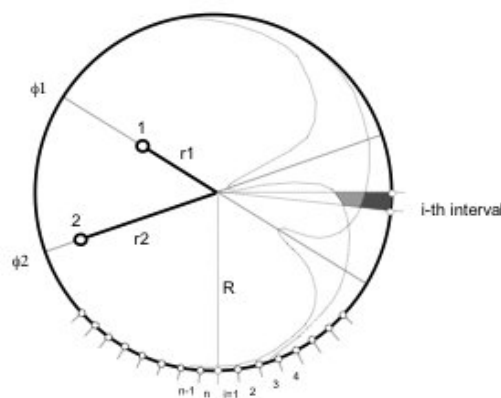


Figure 2.5.1 : Contribution of atoms No.1 and No. 2(at (r_1, φ_1) and (r_2, φ_2)) to the intensity s_i at interval i on the circle with radius R , shown as shaded areas of the corresponding Lorentzian bell-shape functions

Therefore, a Lorentzian curve peak represents for each atom j , a projection located at angle φ_j . The last parameter of the intensity function, σ_j represents the width of the curve associated with each atom, and therefore is the means for the equation to include any possible physico-chemical properties, such as the atom type, electrostatic charge or any other desired atomic property.

Comparing the three methods of molecular structural representation in a form with a fixed number of parameters, regardless of the size of the structure, we see that all the four desired characteristics of uniformity, uniqueness, reversibility and rotational/translational invariance have been achieved. Not a single one of the four however achieved all four. While the method using physico-chemical autocorrelation achieved uniqueness, uniformity and translational/rotational invariance, its converted representation is not reversible to the structure of the actual molecule. Moreover, by using bonds as distance parameters, the representation is restricted to two dimensional molecular representations. Finally, this method of representation takes similarities between pairs of atoms with similar characteristics. Therefore, if a molecule were to occur with only a single atom with a particular characteristic value, regardless of its importance, it will not be characterized by such a method. The second method of deriving representations through the 3D-MoRSE code method, was however able to cope with this limitation but was not reversible. While the final method using projection of structures onto imaginary spheres was able to produce reversibility, this is only in the case where the resolution of the intervals on the circular planes is very high, resulting in a larger number of parameters, or at the expense of its reversibility, on top of its lack of rotational and translational invariance. It is as well evident that all three methods are not totally independent of the size of the molecular structure. A smaller representation results in a loss of unique characterization, and as such, should be scaled to accommodate the largest molecules in the data set chosen. It should be noted therefore that the choice of representation is problem specific, and need not necessarily be one that is able to satisfy all the four requirements.

2.6 Neural Networks In Structure Activity Relationships (SAR) and Drug Design

This section will discuss related research that has been carried out using the various implementations of neural networks in the study of SAR and the various disciplines of drug design. The use of unsupervised neural networks will first be discussed followed by supervised training applications and their overall applicability discussed.

Unsupervised neural networks, in particular Kohonen [Kohonen, (1982)] networks, or Self-Organising Maps (SOMs), have been employed in various applications in drug design and SAR studies. Anzali et al. (1996) used Kohonen networks for the transformation of 3-D molecular surfaces into 2-D Kohonen maps. In this study, the molecular electrostatic potentials (MEP) for the van der Waals surface of cardiac glycosides and ryanodines were calculated and a Kohonen network trained using samples of coordinates from random points on this surface as inputs. Following in with the continuous 3-D structure of molecular surfaces, the mapping was done using a 2-D torus shaped Kohonen Network, with three inputs per neuron, one for each 3-D axis. This strictly structurally inspired network (not taking into consideration any electrostatic or physico-chemical properties) was then trained to bring points with similar coordinates closer to one another. The trained neurons were then colored according to the MEP values of the points represented by the respective coordinates. Anzali et al. (1996) further suggested that any molecular properties can be mapped onto this network including hydrogen-bonding potential and atom type.

Visual reference and comparison of such maps offers a means to look for similarities and differences between molecular structures, for instance ligands, binding to the same active site and relate these comparisons to their binding affinity. This was verified through tests with two different types of receptors, muscarinic and nicotinic. Not only did the analysis of the ligands binding to the same receptors show distinct similarities, characteristic differences were also visible between the two groups of ligands. Taking this study further, Kohonen networks were built using the same methodology, of 31 steroids of known binding affinities with corticosteroid binding globulin (CBG), and divided into three groups according to their binding affinity, one low, one intermediate and one high affinity group. The maps of the ligands within each group were then averaged via indexes assigned to the colours associated with their MEPs. Distinct patterns found in each of the three average maps then proved useful in identifying which group a new ligands might most likely belong to through comparison, with the average map which now represented a pharmacophore of the molecular interaction. Transforming the coloured maps further into vectors (of MEP or colour indexes), as well introduces an alternative in 3-D molecular structure represented earlier discussed in Section 2.3. The feasibility of this method has lead to Kohonen networks representing particular template molecules to be used as a benchmark for comparison with the maps of other molecules to study their degree of similarity of difference. For better comparison, any two molecules, one template and one test, can be superimposed and their respective positional similarity expressed through colouring, such that different degrees of similarity can be represented by different colours and lack of it by white space. Resultant superimpositions of

mainly white space would thus infer a large difference between the two molecules being compared. Template methods as such have been used to study binding between steroids and CBG and TBG [Anzali et al., 1996], [Polanski, 1996, 1997], ryanodine derivatives to membrane proteins [Anzali et al., 1996], and to detect correlations between histamine analogues and H2 activities [Barlow, 1995], nitro and cyanoanilines and arylsulfonylalkanoic acid to sweetness activity, and as well ethylcarboxylates to Taft's E_s constant [Anzali et al., 1998], based on mapping of MEP. Polanski (1996) came up with a similar system, this time using many templates instead of one, called the Multi Template Approach, and used partial least squares (PLS) to analyze it, with applications for modeling structure 3D QSAR of colchicinoids, as potential anti-cancer leads. This method was yet taken further in the classification of dopamine 2 (D2) receptor antagonists [Hasegawa et al., 2002], which are believed to have effect on the corpus striatum and pallidum of the brain where mental diseases such as Parkinson's disease are caused due to dopamine imbalance. Similar methods as those used by Polanski [1999] in colchicinoid characterization were used as far as the mapping of the structural MEP onto the 2-D Kohonen map was concerned, but this time, instead of using the conventional PLS method, a 3-way PLS was used instead for the analysis enhancing the contour mapping density and including neighbouring relations providing the ability for the contour map to be visualized on the van der Waals surface of the molecules themselves making the interpretation of SAR more intuitive.

Kohonen neural networks are however not restricted to mapping single molecules modeling 3-D structural characteristics in QSAR applications. These

networks as well work with groups or library of molecules where self organization is based more on numerically measurable properties of the molecule that may not be inherent to the molecular structure but instead to its experimental reactivity, through more conventional Kohonen clustering. One such example is that of calculating the yield of *para*-xylene under specific reaction conditions [Petit et al., 2002]. *para*-Xylene is a very commonly used chemical compound in the synthesis of textile polyester fibers. It is commonly produced through the alkylation of toluene with methanol under acidic catalysis conditions, as shown in Figure 2.6.1.

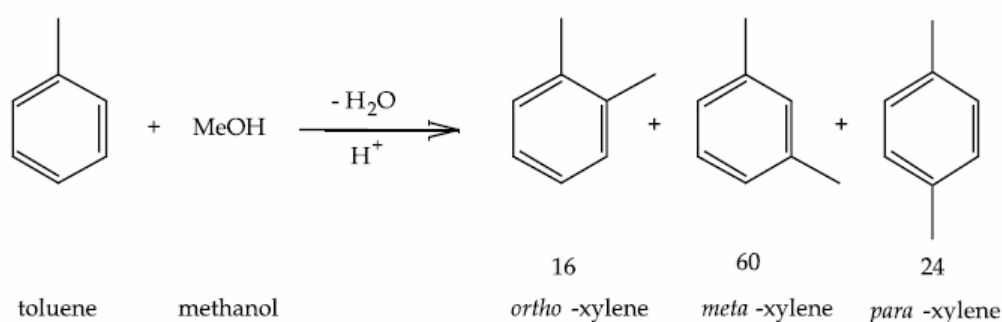


Figure 2.6.1: Reaction scheme on the production of xylene isomers (including relative distribution)

The relative proportions of the three isomers produced, *ortho*-, *meta*-, and *para*-xylene, as seen on the right hand side of Figure 2.6.1, are 16/60/24 [Kaeding et al., 1981]. The separation of the three to get the *para* isomer is made difficult due to their similar boiling points. Through catalysis with crystalline aluminosilicates called zeolites, particularly ZSM-5, yield of the *para*-isomer has been found to increase. This largely empirical catalysis process whose influential factors are still not fully understood is studied in this case using a combination of unsupervised and supervised neural network techniques through the use of Kohonen and counter-propagation (CPG) networks. The three inputs that, in

this case, were chosen based on availability of data, were chosen to be the temperature, the molar ratio of the reagents in the mixture, and the weight hourly space velocity. The output parameters (for the CPG networks), included the conversion of toluene, the weight percentage of the total xylene, and the proportion of *para*-xylene among all the three isomers (all for which experimental target data was available). In this study, Kohonen maps were first used in the division of the entire dataset of 79 samples into a training set and a test set. For the training set, the three dimensional inputs were applied to the network and 37 of the samples were chosen from them to represent the training set based on their uniform spread across the Kohonen map. These 37 samples were then used to train the network to obtain a distinction of regions on the Kohonen map based on the percentage of *para*-xylene among the xylenes. The 42 test samples were then run through the CPG network and their predicted values compared against the targets to find their model feasible in showing a correlation between the input and output parameters chosen.

This far, research in QSAR using Kohonen networks have been primarily discussed. They are however, not the only means of classification in QSAR studies. Backpropagation neural networks as well have been used as a categorization tool in QSAR. One such instance is the odor classification for chemical compounds [Song et al., 1993]. In this study, inputs were that of the plural semiconductor gas sensors' response data (SGSRD), and each of the set of 47 chemical compounds including alcohols and ketones, were to be classified into one of five categories based on their odor. The compounds were either to be classified as ethereal, pungent, minty, ethereal-pungent, and ethereal minty. For this to be accomplished, a three layer feedforward

backpropagation network, with a single input layer, one hidden layer, one output layer. The output layer comprised 3 output neurons, each one corresponding to one of the ethereal, pungent or minty categories. If a single output neuron in the trained network fires with the introduction of a sample, the introduced sample is then said to belong to the corresponding fired neurons category, e.g. ethereal. If however two of the output neurons fire, this would then indicate the classification of the sample into either the ethereal-pungent or ethereal-minty categories. The data coding in this experiment took three phases. In the initial stage, 6 inputs from the characteristic SGSRG were chosen, after which the squares of these 6 vectors were added to the 6 giving a total of twelve and finally, to best describe each chemical compound, a set of 5 more molecular structure codes, namely the first order connectivity index, the number of oxygen atoms, the number of double bonds, the number of carbonyl groups, and finally the number of hydroxy groups. While these parameters do not describe the three dimensional structure of the data, they do provide information on the molecular composition and substructures present. The addition of these molecular properties, though not structurally detail in nature, improved the neural network performance. As well, improvement was seen when using the squared values together with the original ones and correlation was found in between the molecular descriptives chosen and the SGSRD data, through the networks predictability.

Having discussed primarily the use of Kohonen neural networks, The focus from here on will be shifted to that of backpropagation and its applications. In the separation of solutes in capillary zone electrophoresis (CZE), one of the

major influencing factors to the separation is the electrophoretic mobility, μ_0 , whose general form is given below [Eq. 2.6.1],

$$\mu_0 = \frac{q}{f_h} \quad [\text{Eq.2.6.1}]$$

where μ_0 is the mobility at infinite dilution, q is the charge of the solute and f_h is the hydrodynamic friction factor for moving a solute through a continuous solvent of finite viscosity. Li et al. (2002) have developed a means of predicting the electrophoretic mobilities of aliphatic carboxylates and amines using other simpler experimental properties of the compounds as inputs to a feedforward multilayer backpropagation neural network using the extended delta-bar-delta algorithm, a modification to the standard algorithm chosen to overcome the long training times required in stabilizing the network to a suitable weight state. The network designed for this purpose, after a great amount of iterative testing, consisted of 4 input parameters (and neurons), a single hidden layer with 6 neurons, and a single neuron in the output layer for the prediction of the electrophoretic mobility. The input parameters chosen included the molecular volume, weight and charge distribution (pK) value through their influence on the solute radius and orientation of solvent dipoles relative to the solute charge. The last parameter used was a code representing the acidity of the solute, +1 representing a base and -1, an acid. 56 compounds were used in the experiment with 40 reserved for the training, 10 for validation, and the remaining 6 as the test set. The training was set to 28000 epochs, and the results obtained showed a positive correlation between the inputs and the predictability of electrophoretic motility.

While experimental parameters are useful in many cases of QSAR and QSPR, in some, structural inputs are required to better characterize the compounds involved and associate them to some property. For instance, molecular descriptors including 2-D structural, or more so topological input was used to predict the boiling point, density and refractive index of alkenes [Zhang et al., 1997]. In this study, topological indices were used to define the molecular descriptor, which was important to the experiments conducted as the intended predictions were based mainly on the interactions of the molecules used with respect to their size and symmetry. The primary structural parameters considered here were W , based on the molecular distance matrix, and the polarity number, P . The distance matrix, D of the molecule with N atoms is a symmetric $N \times N$ matrix whose elements $(D)_{ij}$ are defined as follows,

$$(D)_{ij} = \begin{cases} l_{ij} & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases} \quad [\text{Eq.2.6.2}]$$

where l_{ij} is the length of the shortest path between the two atoms i and j . The parameter W is then obtained from the distance matrix as follows,

$$W = \frac{1}{2} \frac{\sum_j \sum_i d_{ij}}{(N+1)^2} \quad [\text{Eq.2.6.3}]$$

where N is the total number of carbon atoms in the molecule. The polarity number P is equal to half the number of pairs of atoms that are separated by exactly three bonds. The three outputs to be predicted were also strongly dependent on the double bonding within the molecule. It was seen that the influence of the double bond on the molecule decreases with increasing

molecular size, and this was used to define two further input parameters w and p , as follows,

$$w = \frac{\sum_i (d_{ki} + d_{li})}{\sum_i \sum_j d_{ij}} \quad [\text{Eq.2.6.4}]$$

$$p = \sum_{d_{ik}} (p_3)_i + \sum_{d_{il}} (p_3)_i \quad [\text{Eq.2.6.5}]$$

where k and l represent two carbon atoms connected by a double bond. A fifth and final parameter, s was as well defined to take into consideration the influence of alkene enantiomers, as follows,

$$s = \begin{cases} 1 & \text{trans - isomer} \\ 0 & \text{no enantiomer} \\ -1 & \text{cis - isomer} \end{cases} \quad [\text{Eq.2.6.6}]$$

With these parameters, 80 input samples with number of carbon atoms ranging from 4 to 20, were divided into a training set of 51, a validation set of 18 and a test set of 16 samples were processed by a backpropagation network with 5 neurons in its only hidden layer to obtain a positive correlation between the input s and the outputs, the most accurate output being the refractive index.

As can be seen from the methods used above to describe correlations between structure and activity/property, many methods can be adapted to represent the data to be used in a neural network. The representation must always depend on the respective study and should be chosen to maximize the dependence of the output on the input. The next chapter will run through the methodologies

chosen to study interactions between generic ligands and protein active sites, and how well they bind.

Chapter 3. Methodologies

This chapter will state, define and describe the methodologies used in the experiments run in this thesis. Section 3.1 will first describe how the protein and ligand structures used in the experiments were *obtained and prepared* to be experimentally viable. Section 3.2 will then go into physico-chemical *autocorrelation*, the transformative technique enabling the scaling of a molecular structure file format to that acceptable as input into a feed-forward backpropagation neural network, as well as run through an example of its manipulation. The remaining sections of the chapter will then describe the techniques involved in the design and application of neural networks. Section 3.3 will describe how the data required was prepared for its use in neural networks. Section 3.4 will then explain how *Self-Organizing Maps* were used to study the experimental data categorically, while Section 3.5 will explain feed-forward *backpropagation* and the considerations taken in its exploitation. Finally Sections, 3.6 and 3.7 will respectively describe the use of Multiple Linear Regression as a comparative means of analysis and how the performance of the neural networks were studied using various error measurements. It is to be noted that this particular methodology can be categorized as a semi-flexible docking methodology. Even though there is no docking involved, the predictions on the binding affinity are made on a fixed active site and a fixed ligand. This ligand is however expected to be tested against the active site in all feasible conformations, thus making the system a semi-flexible one.

3.1 Preparation of Interacting Molecules

All molecular structures used in the experiments within this thesis were downloaded from the Research Collaboratory for Structural Bioinformatics' Protein Data Bank (PDB) [Berman et al., (2000)]. All structures were downloaded as protein-ligand complexes, and separation of the molecules was thus required in order for their characterization to be performed as individual elements. All molecular modeling was carried out using Tripos' Sybyl 6.8 [Tripos, USA].

The ligand molecules were first extracted from the protein ligand complexes and saved as individual PDB files. The interaction site residing on the protein molecule was then marked and all amino acid residues with atoms within 5 angstroms of the interaction site were then carved out and saved as the active site in PDB format. PDB files store the structure of molecules as a set of atom coordinates. The bonds between these atoms are however, not explicitly stated within the file format. The PDB format infers the bonds between any two atoms by referring to a table of *chemistry rules*. By mapping spatial Euclidean distance between two atoms to a particular bond type (e.g. single or double bond), software packages are able to infer bond types. As these rules have never been specifically enforced, various software packages may derive different bond types from the same PDB file. This introduces even further complication when non-biopolymer structures, such as those of ligands, are included in the PDB files. Specific atom types, as used in this thesis' experiments, are defined through the bonds they hold with their surrounding

atoms. Sticking to the PDB format for such experiments might thus prove detrimental to their accurate results.

To overcome the shortcomings of the PDB format (through which the protein-ligand complex structures were archived), each active site as well as ligand, was converted to their respective mol2 format (Tripos' native file format) equivalents. The mol2 format expresses the bonds between any two atoms explicitly, and additionally has provision to store the specific type of each atom within the file.

Once converted, each molecular structure was then manually checked and corrected to ensure the proper bonding and atom typing. Active site bonds were corrected according to each amino acids native structure, while each ligand was corrected according to their representation within PDBSum [Laskowski et *al.*, 1997]. Once all the bonds were corrected, the individual atoms were then typed using the Sybyl atom types reflected in Table 3.1.1.

Once the correct atom types were verified manually, all hydrogen atoms were removed from the active site, and ligand structures. This was done primarily to avoid any discrepancies in the hydrogen locations, which are usually there as locations of hydrogen atoms in space cannot be resolved through X-Ray Crystallographic methods, and as such most database structures lack appropriate hydrogen atom coordinates [Baxevanis, 1998].

Table 3.1.1 : Atom Types Selected For Characterization with Hydrogen Bonding Characteristics

Atom Type Definition	Mnemonic Code	Hydrogen Donor	Hydrogen Acceptor
Carbon sp3	C.3	No	No
Carbon sp2	C.2	No	No
Nitrogen sp3	N.3	Yes	Yes
Nitrogen sp2	N.2	Yes	Yes
Nitrogen sp	N.1	No	Yes
Nitrogen aromatic	N.ar	No	Yes
Nitrogen trigonal planar	N.pl3	Yes	No
Nitrogen ap3 positively charged	N.4	Yes	No
Nitrogen amide	N.am	Yes	No
Oxygen sp3	O.3	Yes	Yes
Oxygen SP2	O.2	No	Yes
Oxygen in carboxylate and phosphate groups	O.co2	No	Yes
Oxygen in Single Point Charge (SPC) water model	O.spc	Yes	Yes

The next step in preparing the ligand and active site structures was in the addition of electrostatic atom point charges to the atoms of the molecules. These parameters are essential in the consideration of electrostatic interactions between the protein and ligand leading to their binding [Honig et al., 1995]. As PDB files do not contain reliable electrostatic data, the partial atom point charges needed to be computed for electrostatic characterization of the ligands and active sites. Gasteiger-Huckel charges were used for this, computed by Sybyl. Gasteiger-Huckel charges are a combination of the Gasteiger-Marsili [Gasteiger et al., 1980] and Huckel [Streitwieser, 1961] method of charge calculation, the former incorporating the σ component while the later calculates the π component. No further formal charges were then added to the molecules.

Attention was taken to keep the molecules in their original conformation to retain the molecular shape at binding, which is essential input to the autocorrelation process described in Section 3.2.

3.2 Physico-chemical Autocorrelation

This section will describe the physico-chemical structure encoding methodology, *autocorrelation*, used to transform a chemical structure from the PDB to one suitable as input into a neural network.

Each molecule presents its unique set of characteristics. It comprises various atom types, each possessing its own range of properties, in differing quantity, in its unique topological arrangement. This poses a challenge to machine learning methods such as neural networks where a fixed number of descriptors is required despite the differentiation between molecules. The neural network requires a fixed number of inputs, which should contain within, the chosen properties of the molecular structure, that are responsible for the biological effect being investigated [Zupan et al., 1999]. A transformation is thus needed for the physico-chemical representation of a molecule into a fixed number of parameters.

The transformation method used in this thesis is autocorrelation [Moreau et al., 1980]. In autocorrelation, each property, p , of an atom, a , under investigation is correlated with the same property, p , on another atom, b . The summation of these autocorrelated products over all the atom pairs are then taken over

predefined topological distances (number of bonds between two atoms), d , as is described as the function, $A(d)$ in Eq. (3.2.1).

$$A(d) = \sum_{b=a+1}^n \sum_{a=1}^n \delta_{ab} p(a) p(b) \quad [\text{Eq. 3.2.1}]$$

The experiments run in this thesis modified Moreau's method in two ways. Firstly, topographical distances were used instead of topological, and secondly, molecular characteristics without magnitude, such as Atom Type, were given the value of 1, instead of the product, $p(a)p(b)$. These modifications were made firstly to account for differences in molecules of the same composition but with varying conformations. The distances taken into consideration are thus the Euclidean distances between the atoms in 3D space. The second modification was made primarily to enable representation of characteristics that were without magnitude but are essential to be considered spatially. To illustrate this, let us take the water molecule in Figure 3.2.1 as a simplified example.

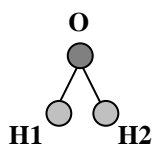


Figure 3.2.1 : 2D Structure of water molecule.

Table 3.2.1 below contains arbitrary 3D coordinate data along with charge of the atoms for explanatory purposes.

Table 3.2.1 – Atom coordinates and charge for water molecule

Atom	X-Coordinate	Y-Coordinate	Z-Coordinate	Charge
O	4.013	0.831	-9.083	1.2
H1	4.941	0.844	-8.837	2.3
H2	3.75	-0.068	-9.293	2.3

Assuming we would like to find $A(d)$, for all atoms whose charge, p , lies between 2.0 and 3.0, where d is a range of 1 to 2 angstroms. As we know the two hydrogen atoms present the only atom pair that meet the property requirement, we calculate their Euclidean distance,

$$d_{H_1H_2} = \text{Sqrt}((4.941-3.75)^2 + (0.844+0.068)^2 + (-8.837+9.293)^2) = 1.567 \quad [\text{Eq. 3.2.2}]$$

Therefore,

$$A(d) = 2.3 \times 2.3 = 5.29 \quad [\text{Eq. 3.2.3}]$$

Similarly, if the property required, p , was for the atom to be a hydrogen atom, given the same d , as there is only 1 such atom pair, $A(d) = 1$ in the latter case. This is different from Moreau's methodology as his would require the topological distance d to be 2 for the same $A(d)$ as in Equation 3.2.3.

In this study, the properties and respective distance ranges were selected for both the ligand and the active site structures, as tabulated in Table 3.2.2. The ranges were chosen through a process of iterative refinement to scale the values of $A(d)$ accordingly, thus avoiding large differences amongst the various parameters.

Table 3.2.2 : Parameters and respective ranges used for autocorrelation

Atom Parameter, p	Distance Ranges, $d/\text{Å}$
$T = \text{N.1, O.2, N.ar, O.co2}$	$d < 3, 3 \leq d < 6, 6 \leq d < 9, 9 \leq d < 12, 12 \leq d < 15, 15 \leq d < 18, 18 \leq d < 21, d \geq 21$
$T = \text{N.pl3, N.am, N.4}$	$D < 3, 3 \leq d < 6, d \geq 6$
$T = \text{N.2, N.3, O.3, O.spc}$	$D < 3, 3 \leq d < 6, d \geq 6$
$T = \text{C.2}$	$D < 3, 3 \leq d < 6, d \geq 6$
$T = \text{C.3}$	$d < 3, 3 \leq d < 6, 6 \leq d < 9, 9 \leq d < 12, 12 \leq d < 15, 15 \leq d < 18, d \geq 18$
$C > -0.5$	$d < 3, 3 \leq d < 6, 6 \leq d < 9, 9 \leq d < 12, d \geq 12$
$-1 < C \leq -0.5$	$d < 3, 3 \leq d < 6, 6 \leq d < 9, 9 \leq d < 12, 12 \leq d < 15, 15 \leq d < 18, 18 \leq d < 21, d \geq 21$
$C \leq -1$	$d < 3, 3 \leq d < 6, 6 \leq d < 9, 9 \leq d < 12, 12 \leq d < 15, 15 \leq d < 18, 18 \leq d < 21, 21 \leq d < 24, d \geq 24$

$T = \text{atom type, } C = \text{Gasteiger-Huckel charge}$

All autocorrelative vectors were obtained by parsing mol2 files using Perl scripts, extracting coordinate and physico-chemical data (charge and atom type) from them to produce the required $A(d)$ values. The Perl scripts used can be referred to in Appendix A.

3.3 Preprocessing, Postprocessing and Normalization

The data used in this thesis for the training and testing of the neural network vary in magnitude to a great extent. As such, to prevent the more significant vector components from dominating the training, in this case, for the Self-Organizing Map (SOM) and Feed-forward Backpropagation (FFBP) networks, the data(input and target) needs to be preprocessed through normalization.

The normalization method used for the SOM training was taken from the SOM Toolbox [Kohonen et al., (1996)], through the command SOM_NORMALIZE. For this method, variance normalization was selected to be performed such that the values of the input vectors are scaled through a linear transformation such that their variance is equal to 1. The command was used as follows:

$$inNorm = som_normalize (in, 'var')$$

where *inNorm* is the resultant set of normalized vectors, taking *in* as the actual input vector, and using the variance method, '*var*'.

The normalization method adopted for the backpropagation experiments, *prestd*, were adapted from the MATLAB software package. *prestd* scales the network inputs and targets by being normalized to have a mean of zero and a unity standard deviation, through the command,

$$[pn, meanp, stdp, tn, meant, stdt] = prestd (p,t)$$

where *p* = input vector, *t* = target vector, *pn* = normalized input vector, *meanp* = input vector mean, *stdp* = input standard deviation, *tn* = normalized target vector, *meant* = target mean, and *stdt* = target vector's standard deviation. Once training was complete, the normalized vectors were converted back to their original scale through the command *poststd* requiring the normalized input or target vectors along with their respective means and standard deviations as inputs to the function.

3.4 Self-Organizing Maps (SOM)

This section deals with a method of self-organized or unsupervised learning known as the SOM [Kohonen, 1982, 1989], which is often used to visualize and help better understand high dimensional data sets through the geometric clustering among data sets with similar characteristics on a 2-D display. In this thesis, the SOM is used to adaptively transform the input vectors of the protein-ligand complex, which is a 94 dimensional vector into a 2-D map. The aim of this is to uncover significant characteristic of the input data without the necessity of correlating them to any output. This means that the data organizes itself according to the similarities and differences inherent in its structure. Such a method was important in primarily ensuring that the input data was spread well throughout the SOM, to provide for a fair training in the later backpropagation stage (input data clustering around one region of a SOM would imply that the data is biased and would not necessarily be optimal for training a generic back propagation neural network that is to be used for all possible protein-ligand complexes).

The SOM comprises a grid of neurons each with a model vector, and upon completion of training, the models are arranged on the grid such that similar models are topologically closer to one another. SOM training is based on two processes, the first of competition and the next, cooperation. The process of competitive learning where the introduction of the input, i into a network induces the firing of one and only one of the neurons in the output layer. This output neuron is known as the *winning neuron*, c , or Best Matching Unit (BMU). The winning neuron then adapts its weight along with the weight of the neurons around it, called its *neighbourhood*, $h_{c(w)}$.

Let the dimension of the input space be n . Let any input vector within this input space be denoted by

$$\mathbf{x} = [x_1, x_2, \dots, x_n]^T \quad [\text{Eq. 3.4.1}]$$

Each model vector (also known as weight vector), \mathbf{w} , associated with each neuron, i , generally has the same dimension as the input vector,

$$\mathbf{w}_i = [w_{i1}, w_{i2}, \dots, w_{in}]^T \quad [\text{Eq. 3.4.2}]$$

for $i = 1, 2, \dots, l$, where l is the number of neurons in the network. The winning neuron, c is thus found by identifying the neuron with the minimum Euclidean distance between the vectors \mathbf{x} and \mathbf{w}_j [Eq. 3.4.3]:

$$|\mathbf{x}(t) - \mathbf{w}_c(t)| \leq |\mathbf{x}(t) - \mathbf{w}_i(t)| \quad \forall i \quad [\text{Eq. 3.4.3}]$$

Then comes the cooperation step. The process of regressing the ordered set of model vectors, \mathbf{w}_i into the space of input vectors, \mathbf{x} is traditionally made by the following equation :

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + h_{c(x),i}(\mathbf{x}(t) - \mathbf{w}_i(t))$$

where $h_{c(x),i}$ is the neighbourhood function which is often a Gaussian function :

$$h_{c(x),i} = \alpha(t) \exp\left(-\frac{|r_i - r_c|^2}{2\sigma^2(t)}\right) \quad [\text{Eq. 3.4.5}]$$

where $0 < \alpha(t) < 1$ is the learning-rate factor, decreasing monotonically with the number of iterations, $r_i \in \mathfrak{R}^2$ and $r_c \in \mathfrak{R}^2$ are the vector locations on the neuron map and $\sigma(t)$ corresponds to the width of the neighbourhood function, which also decreases monotonically with the number of iterations.

Instead of randomly initializing the weight vectors \mathbf{w}_i , the initial values of the weight vectors of the neurons are selected as a regular array of vector values that lie on the subspace spanned by the eigenvectors corresponding to the two largest principal components of input data. This makes the training of the organization of the SOM much more efficient since the SOM is already partially organized in the beginning [Kohonen, (1995)].

In this thesis, a modification of the traditional algorithm just discussed was used, known as a batch algorithm for significantly faster computation. In the batch method, once the model vectors, \mathbf{w}_i are initialized, a list of all the input samples $\mathbf{x}(t)$ is collected for each neuron i , whose most similar model vector belongs to the neighbourhood, N_i of node i . Then for each new model vector, the mean over the respective list is taken. The steps after the initialization are then iterated for a pre-defined number of times. The number of iterations used in this thesis' experiments ranged from 50 to 3500, based on when the SOM stabilized (when the neurons on the map to which the input vectors were most similar stopped changing). The SOM grid was chosen to be hexagonal in shape for better visualization, and the number of units in the grid was determined by choosing the 'big' size option on the *mapsize* variable of the SOM. This 'big' size translated to the size of the network being calculated as follows:

$$\text{No. neurons in 'big' mapsize} = 4 * 5 * (\text{No.of samples})^{0.54321} \quad [\text{Eq. 3.4.6}]$$

The dimensions of the map were then determined by taking the two biggest eigenvalues of the training data and the ratio between them sets the ratio between the sidelengths of the map grid. The actual sidelengths are then calculated in a manner that makes their product as close to the number of

neurons in the map as possible. In the experiments run, the dimensions of the maps used were 22 x 10 neurons. For visualization purposes, the actual maps were generated with the PDB IDs of the protein-ligand complexes displayed over their respective BMUs.

3.5 Feed-forward Backpropagation

This section will describe the fundamentals about Feed-Forward Backpropagation Neural Networks (FFBPNN), a supervised learning method (requiring pairs of input-targets), and will explain how this technology was adapted suitably to the experiments run in this thesis, how it was implemented as well as the justification of the parameters chosen for its running.

3.5.1 Introduction to Backpropagation

A neural network, can be viewed as a 'black box' whereby an 'm' variable input can be transformed into a 'n' variable output. The input and output variables are usually normalized real numbers, binary numbers (0 or 1), or bipolar numbers (-1 or +1) [Zupan et al., (1999)]. The problems neural networks are used to solve are that of association, classification, transformation, and modeling. The experiments run here were those of the modeling category using normalized real numbers as input and as outputs. Modeling in neural network problems searches for an analytical methodology to predict a particular 'n' variable output from an 'm' variable input. Neural networks make this possible without the advance knowledge of a mathematical function. Training of the neural networks (to be described in greater detail) involves finding the best fitting between the input parameters and the outputs. Accuracy of the predictions increases when

the experimental data is spread evenly and sufficiently over the entire region. For this purpose, Self-Organising Maps, as described in Section 3.5, were first used to ensure the even spread of the experimental data selected. In this thesis, there were 94 input parameters, collectively describing the spatial atomic distribution of charges and hydrogen bonding capacity as well as molecular weight of the protein active sites and the ligands, respectively, and a single output parameter, describing the binding affinity. The neural network designed can be better visualized through Figure 3.5.1.

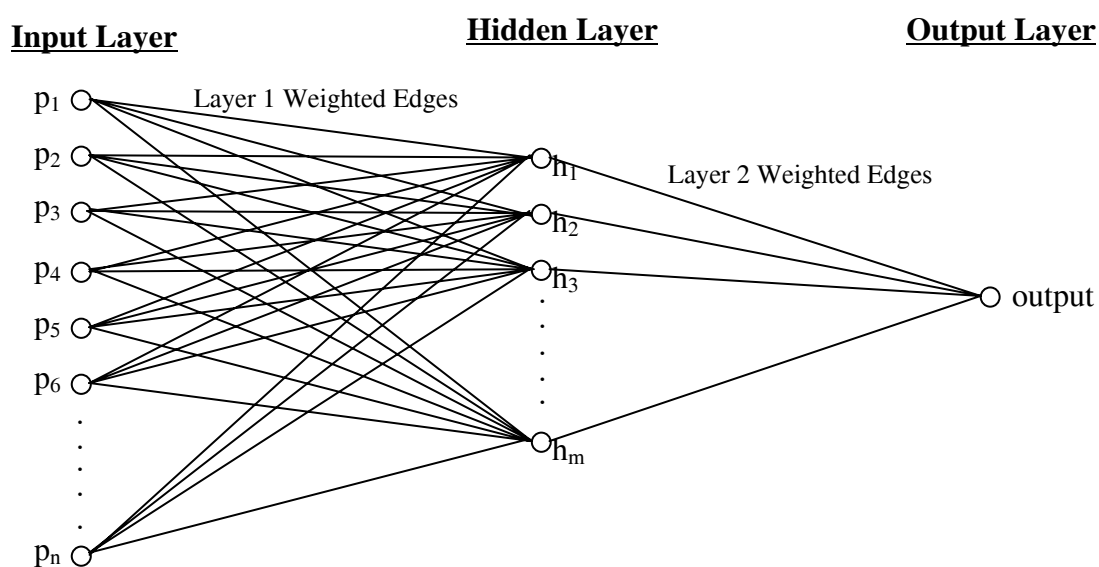


Figure 3.5.1 : Structure of a Feed-forward Backpropagation Neural Network

Section 3.5.1 Neural Network Training and Architecture

FFBPNs work through a process of input data first being fed forward into the neural network (Feed-forward part) through a series of weighted links, processed at each node (neuron) within the network to eventually come up with a prediction, have the output prediction compared to the actual results, and the

weights on the edges are corrected from the last layer back to the first (back-propagation part). This is illustrated in the schematic diagram in Figure 3.5.2 [Zupan et al., (1999)]

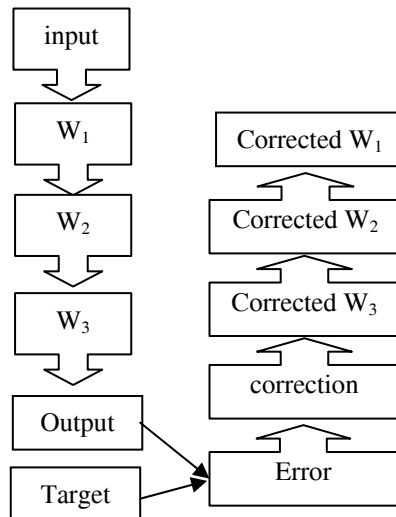


Figure 3.5.2 : Schematic presentation of weight correction with backpropagation. W_x represents weights in layer x .

The architecture is the most prominent characteristic that influences the performance of the neural network it represents. The architecture comprises the number of neurons there are in each layer of the network, the number of layers in the network and the way in which the neurons in one layer are connected to those in the next. In these experiments, fully connected inter-layer neural networks were used. This means that each neuron in a layer in the neural network was connected to every other neuron in the next layer. As in the case of most neural networks, the ones used for these experiments comprised a single input layer of 94 neurons or nodes, and two active layers, a hidden layer with a range of nodes, for testing purposes, and an output layer for a single result. The number of nodes in the hidden layer ranged from 5 to 60, with

intervals of 5 (5, 10, ..., 60) in order to find the best topology which presented the best fitting and lowest error.

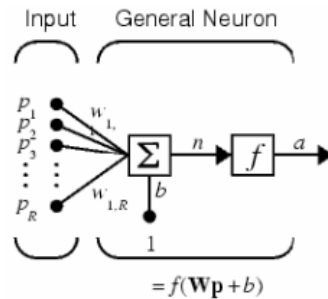


Figure 3.5.3: Illustration of a general neuron within a backpropagation neural network
 R =No. of elements in input vector, p , W = weight vector, w = individual weight,
 f = activation function, n = summation of input-weight products and bias b , a = output from node

At each layer of nodes, each neuron model has associated with it an activation function. Several activation functions exist including the tangent-sigmoid, logarithmic-sigmoid and linear activation functions. The activation functions can differ between layers and if strict customization is required, between neurons in the same layer as well. In this thesis, the neurons in the hidden layer all used the tangent-sigmoid activation function while the output neuron used the linear activation function. The graphs of the two functions are illustrated in Figure 3.5.4.

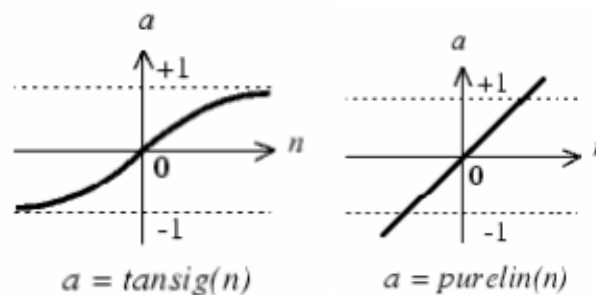


Figure 3.5.4: Tangent-Sigmoid (left) and Linear (right) Functions used in the experiments.

In a FFBPNN, there are two main phases, the training of the network, and the use of the trained network to model/predict results given a new input vector. Within the training phase, as described above, there is the feed-forward of the inputs and back-propagation of error. In the feed-forward phase, the input vector \mathbf{p} of input elements p_1, p_2, \dots, p_R , where R is the number of elements in the input vector, is first introduced into the network and propagated towards the first layer of hidden neurons, through the respective synaptic weights. The error is then calculated at the output node, and propagated backwards by computing the gradient using the chain rule and correcting the weights in the direction of the negative gradient of the performance function. The basis of the error calculation at the output node is the Delta Rule, where the error for N the output nodes at iteration n is calculated as,

$$E_j(n) = \frac{1}{2} \sum_{j=1}^N (t_j(n) - y_j(n))^2 \quad [\text{Eq. 3.5.1}]$$

where $t_j(n)$ is the target of the n th training sample at node j , and $y_j(n)$ is the actual output. In our case, however, $N = 1$, as there is only a single output node, giving us,

$$E(n) = \frac{1}{2} (t(n) - y(n))^2 \quad [\text{Eq. 3.5.2}]$$

Having now calculated $E(n)$, the local gradient $\delta_j(n)$ for the output node can be calculated through the formula [Haykin, 1999],

$$\delta_j(n) = e_j(n) \phi_j'(v_j(n)) \quad [\text{Eq. 3.5.3}]$$

where $\varphi_j'(v_j(n))$ is the derivative of the activation function used, $(v_j(n))$ being the induced local field produced at the input of neuron j 's activation function, and $e_j(n) = t_j(n) - y_j(n)$. We however need to perform a gradient descent for error calculation on all the weights, in the output layer as well as the hidden layer. The challenge is now to calculate the error for the hidden layer as we do not know its direct target output.

To calculate the local gradient at a single hidden node g , considering its connection to nodes in succeeding layer h , and using the same symbolic convention as in equation 3.5.3 above,

$$\delta_g(n) = \varphi_g'(v_g(n)) \sum_h \delta_h(n) w_{hg}(n) \quad [\text{Eq. 3.5.4}]$$

As this thesis' experiments used a single hidden layer with a single output node, the calculation for $\delta_g(n)$ at each hidden node j , can be described as follows,

$$\delta_g(n) = \varphi_g'(v_g(n)) \delta_{out}(n) w_{out,g}(n) \quad [\text{Eq. 3.5.5}]$$

where $\delta_{out}(n)$ and $w_{out,g}(n)$ are the gradient at the output node and the weight between hidden node j and the output node, respectively. Now that the gradients for both the output and hidden nodes are calculable, the change to the weights between any two nodes a and b at iteration n is made as follows,

$$\Delta w_{ab}(n) = \eta \delta_b(n) y_a(n) \quad [\text{Eq. 3.5.6}]$$

where η is the pre-defined learning rate, and $y_a(n)$ is the input signal of neuron b . In these experiments, to make the training faster, a momentum term was included in the training. The momentum term help training by adjusting the weights in proportion to the previous weight adjustment. The parameter λ represents this momentum term included in equation 3.5.7 below,

$$\Delta w_{ab}(n) = \eta \delta_b(n) y_a(n) + \lambda \Delta w_{ab}(n-1) \quad [\text{Eq. 3.5.7}]$$

The backpropagation algorithm used in these experiments used batch training, where the weights and biases of the network were updated only after the whole training set was applied to the network (also known as 1 epoch). The MATLAB Neural Network Toolbox was used for this using the function ***traindm***.

Due to the nature and quantity of the data sample size used, the leave-one-out procedure of training was adopted. This meant that the neural network was trained for each of the 128 samples individually, by leaving out the sample for which the binding affinity was to be predicted, and training the network using the remaining 127 samples. The binding affinity was then tested by applying the left out sample to the network, thus computing the predicted result. For 128 samples, the training of the network was therefore done 128 times, for all the topologies previously mentioned. The number of epochs each training session was run was for was 10,000, with a learning rate of 0.05 and a momentum coefficient of 0.9. These parameters were obtained through a series of iterative testing with various topologies, and chosen based on the lowest error readings.

3.6 Multiple Linear Regression Analysis

This section will explain the means by which Multiple Linear Regression(MLP) analysis was used as a comparison for the prediction of binding affinity, and how it was used.

MLP was used in this thesis to test the relationship between the input vectors/parameters chosen (independent variables) and the target binding affinity results (dependent variables). The SPSS software application was used for all regression runs and the results to these experiments can be referred to in Appendix B along with all the other binding results. The leave-one-out method was used once again for the MLP runs, as they were for the backpropagation experiments in this thesis. In this case, the regression line was obtained for the prediction of each complexes binding affinity by leaving that sample out and calculating the regression line using the remaining 127 samples. The binding affinity for the left out sample, Y_i , was then calculated using the equation

$$Y_i = a_i + \sum_{j=1,}^N b_j x_j \quad [\text{Eq. 3.6.1}]$$

Where $N = \text{no. of parameters}$, which is 94 in this case, a_i represents the constant, b_j , the respective slope, and x_i , the value if the parameter concerned. This provided an alternative set of predicted results for each of the 128 data samples. This methodology provided an alternative for the backpropagation methodology to be compared against, in order to grade its relative performance.

3.7 Error Calculation

This section will describe the various methods used to calculate how well the neural network performed compared to predictions by Multiple Linear Regression (MLP) as well as the BLEEP [Nobeli et al., (2001)] method, namely using Root Mean Squared Errors (RMSE), Mean Absolute Errors (MAE), Relative Root Mean Squared Errors (RRMSE) and finally Relative Mean Absolute Errors (RMAE) [Setiono et al., (2002)].

To calculate the performance of the neural network designed and implemented, and as well compare its performance to that of other methods, in this case, that of MLP and the BLEEP methods, a means of error calculation was needed. The following equations 3.7.1 to 3.7.4 define the RMSE and MAE methods used,

$$\mathbf{RMSE} = \sqrt{\sum_{p=1}^P \frac{(\tilde{y}_p - y_p)^2}{P}} \quad [\text{Eq. 3.7.1}]$$

$$\mathbf{MAE} = \frac{1}{P} \sum_{p=1}^P |\tilde{y}_p - y_p| \quad [\text{Eq. 3.7.2}]$$

where P is the total number of all samples, i.e. 128, \tilde{y}_p is the value predicted by the respective method used for sample p , and y_p is the target value of sample p . While these methods prove to be useful comparing one method against the next, a further analysis method was used to calculate the error produced relative to that produced through a naïve calculation of average values of the samples [Setiono et al., (2002)], through the calculation of the RRMSE and RMAE, as shown in equations 3.7.3 and 3.7.4 ,

$$\mathbf{RRMSE} = 100 * \mathbf{RMSE} / \sqrt{\frac{\sum_{p=1}^P (\bar{y}_p - y_p)^2}{P}} \quad [\text{Eq. 3.7.3}]$$

$$\mathbf{RMAE} = 100 * \mathbf{MAE} / \left(\frac{1}{P} \sum_{p=1}^P |\bar{y}_p - y_p| \right) \quad [\text{Eq. 3.7.4}]$$

where \bar{y}_p is the average result taken for all the samples. To ensure fairness in comparison, the leave-one-out method was used in the calculation of this average as well. To thus find the naïve result for a protein-ligand complex, \bar{y}_i , the following calculation was used

$$\bar{y}_i = \frac{1}{P-1} \left(\sum_{p=1, p \neq i}^P y_p \right) \quad [\text{Eq. 3.7.5}]$$

The calculation of relative errors RRMSE and RMAE provide the advantage by showing that a relative error result greater than 100 shows that the predictive methodology used performs worse than a method that uses averages of results for its predictions.

Chapter 4. Data Used

This chapter will discuss the relevance, format and structure of the data used for the experiments run in this thesis. Section 4.1 will discuss the requirements of the data chosen and the importance of the parameters being studied. Section 4.2 will then describe the selection process, and sources of the data. The data requirements for adaptation to neural networks will consecutively be explained in Section 4.3, which will as well describe the outputs of the neural networks, both for Self-Organizing Maps and Feed-Forward Back propagation Neural Networks.

4.1 Data Relevance and Requirements

In the drug design and development arena, the predictability of chemical processes is vital for the saving of cost and resources. In this thesis, protein-ligand interactions are being studied and their binding affinity predicted. Accurate predictions made *in-silico* give synthetic chemists, and biochemists an advantage by increasing the confidence levels of their respective experiments, *in-vitro* and *in-vivo*, enabling them to make more informed decisions on the viability of each experiment. It is essential, as well, that the data required for the predictive experiments to be run are easily available, and the outputs, in an easily understood format. Chemists synthesize ligands (drug leads), in the form of powders and solution by first having a single chemical structure in mind, such as that of benzamidine in Figure 4.1.1.

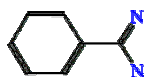


Figure 4.1.1: 2-D Chemical structure of

These synthetic chemists create these molecules through a process of *mixing and matching* molecular fragments in a series of one or more chemical reactions. As such, the computational predictive methodology used should enable these chemists to study reactions (in this case protein-ligand binding), with just a 2-dimensional structure in mind. 3-Dimensional protein structures are readily available from the Protein Data Bank. Once the chemists know the location of the active site on the protein concerned, which can be found by various computational methods including grid and solvation methods [Sybyl, Tripos], they should then be able to predict how well their ligands will bind to the selected active sites. The method used in this thesis does just that.

When a drug binds to a protein active site, many interactions take place on many levels. Two of the most important ones are hydrogen bonding (between hydrogen donors and hydrogen acceptors) and electrostatic interactions (where an atom with a negative charge attracts another with a positive charge, and vice versa). A drug binding to a protein active site has been very much associated to a key put into a lock. This further suggests that shape and size are important factors. As such, size (through molecular weight calculation), shape (through autocorrelation [Moreau et al., (1980)]), electrostatics (Gasteiger Huckel charges [Sybyl, Tripos]) and hydrogen bonding, through atom typing (Ref. Table 3.1.1), have been used as input parameters to this theses'

predictive methodology. The adaptation of the input requirements to the appropriate format for neural network study has been explained in Chapter 3.

4.2 Data Selection

This section will describe the source and selection criteria of the 128 samples, their molecular structures and the file formats of the manipulated molecular data. In order to perform experiments predicting binding affinity using neural networks, training data that contained actual results from wet-lab protein-ligand binding was required.

The protein-ligand complexes sought after were firstly, those whose bound structures were available, along with the experimental binding energy. 3-D structures, with atom coordinate data were important due to the necessity for inter-atomic Euclidean distance calculations. The desired complexes had to have a single active site, with a single ligand bound to it for two reasons. Firstly, the neural network was designed to accommodate one of each of the molecular structures. With a multiplicative effect, scaling and comparisons with 1-to-1 bound structures would not have produced fair results. Secondly, binding energy is calculated in its totality. The binding energy is not localized to each binding interaction, therefore modeling a multiple active site, multiple ligand bind would not truly reflect the local interactions that this thesis' experiments are modeling.

128 datasets that met these criteria (reflected in Table 4.2.1) were taken from the Protein Ligand Database (PLD) [Puvanendrampillai et al., 2003], a repository of binding information including wet-lab molecular binding energy of

protein-ligand complexes, whose structures were readily available from the PDB.

Table 4.2.1 : List of 128 Protein-Ligand Complexes Used in Experiments

PDB ID	Protein Name	Ligand Name	Binding Energy(kJ/mol)
1add	Adenosine deaminase	1 - deaza - adenosine (DAA)	-38.45
1adf	Alcohol dehydrogenase	Beta - methylene thiazole - 4 - carboxamide adenine dinucleotide (beta-tad) (inhibitor)	-26.11
1am6	Carbonic anhydrase ii	Acetohydroxamate	-24.7
1anf	Maltodextrin binding protein	GLC - GLC	-31.13
1b5g	Human thrombin	Novel synthetic peptide mimetic inhibitor and hirugen	-45.64
1bcd	Carbonic anhydrase ii	Trifluoromethane sulphonamide	-22.25
1bll	Leucine aminopeptidase	5 Residues [LEU - FOR - VAL - VAL - ASP] Amastatin	-38.22
1bn1	Carbonic anhydrase ii	N - [4 - methylohenyl) methyl] 2 , 5 - thiophenedisulfonamide [AI5917]	-53.29
1bn3	Carbonic anhydrase ii	2 - (3 - methoxyphenyl) - 2h - thieno - [3 , 2 - e] - 1 , 2 - thiazine - 6 - sulfonamide - 1 , 1 - dioxide	-56.42
1bnn	Carbonic anhydrase ii	3 , 4 - dihydro - 2 - (3 - methoxyphenyl) - 2h - thieno - [3 , 2 - e] - 1 , 2 - thiazine - 6 - sulfonamide - 1 , 1 - dioxide [AI7182]	-57.05
1bnq	Carbonic anhydrase ii	(R) - 4 - ethylamino - 3 , 4 - dihydro - 2 - (2 - methoylethyl) - 2h - thieno [3 , 2 - e] - 1 , 2 - thiazine - 6 - sulfonamide - 1 , 1 - dioxide [AI4623]	-54.14
1bnt	Carbonic anhydrase ii	3 , 4 - dihydro - 4 - hydroxy - 2 - (4 - methoxyphenyl) - 2h - thieno [3 , 2 - e] - 1 , 2 - thiazine - 6 - sulfonamide - 1 , 1 - dioxide [AI5424]	-55.92
1bnu	Carbonic anhydrase ii	3 , 4 - dihydro - 4 - hydroxy - 2 - (2 - thienymethyl) - 2h - thieno [3 , 2 - e] - 1 , 2 - thiazine - 6 - sulfonamide - 1 , 1 - dioxide [AI5300]	-55.33
1bnv	Carbonic anhydrase ii	(S) - 3 , 4 - dihydro - 2 - (3 - methoxyphenyl) - 4 - methylamino - 2h - thieno [3 , 2 - e] - 1 , 2 - thiazine - 6 - sulfonamide - 1 , 1 - dioxide [AI7099a]	-50.03
1bnw	Carbonic anhydrase ii inhibitor	N - (2 - thienylmethyl) - 2 , 5 - thiophenedisulfonamide	-51.8
1bra	Trypsin	Benzamidine	-10.44
1byg	Kinase domain of human c - terminal src kinase	Staurosporine	-56.6

1bzm	Human carbonic anhydrase i	Sulfonamide	-34.4
1c83	Tyrosine phosphatase 1b	6 - (oxalyl - amino) - 1H - indole - 5 - carboxylic acid	-19.24
1cbs	Retinoic - acid - binding protein type ii	All - trans - retinoic acid	-41.08
1cbx	Carboxypeptidase	L - benzylsuccinate	-36.23
1cf8	19a4	4a - methyl - 5 , 6 - epoxy - octahydroquinoline - N - oxide	-34.41
1cil	Carbonic anhydrase ii	(4S - trans) - 4 - (ethylamino) - 5 , 6 - dihydro - 6 - methyl - 4H - thieno (2 , 3 - b) thiopyran - 2 - sulfonamide - 7 , 7 - dioxide	-53.8
1cps	Carboxypeptidase a	S - (2 - carboxy - 3 - phenylpropyl) thiodiimine - S - methane (CPM)	-37.99
1ctr	Calmodulin	Trifluoperazine	-24.45
1ctt	Cytidine deaminase	3 , 4 - dihydrozebularine	-25.79
1dbb	Fab' fragment of the monoclonal antibody db3	Progesterone	-51.38
1dbj	Fab' fragment of monoclonal antibody db3	Aetiocholanolone	-43.83
1dbk	Fab' fragment of monoclonal antibody db3	5 - beta - androstane - 3 , 17 - dione	-46.22
1dbm	Fab' fragment of monoclonal antibody db3	Progesterone - 11 - alpha - ol - hemisuccinate	-53.9
1dwb	Alpha - thrombin	Benzamidine	-16.66
1dwc	Alpha - thrombin	Md - 805 (mitsubishi inhibitor)	-42.27
1dwd	Alpha - thrombin	Napap	-46.62
1e96	Ras - related c3 botulinum toxin substrate 1	Guanosine - 5' - triphosphate	-29.78
1eap	17E8	Phenyl [1 - (1 - N - succinylamino) pentyl] phosphonate	-35.42
1eed	Endothiapepsin	Cyclohexyl renin inhibitor pd125754	-27.39
1epo	Endothiapepsin (aspartic proteinase)	5 Residues [MOR - PHE - NLE - CHF - NME] cp-81,282	-45.41
1etr	Epsilon - thrombin	3 Residues [MQI - ARG - MCP] mqpa	-42.28
1fkf	FK506 binding protein (FKBP)	Fk506 (tacromilus)	-55.37
1fkg	Fk506 binding protein (fkbp)	(1R)- 1, 3 - diphenyl - 1 - propyl (2S) - 1 - (3 , 3 - dimethyl - 1 , 2 - dioxopentyl) - 2 - piperidinecarboxylate	-36.86
1flr	4 - 4 - 20 fab fragment	Fluorescein	-26.55
1hbv	Hiv - 1 protease	Sb203238 - 2 - [3 - benzyl - 5 - (1 - alanyl - aminoethyl) - 2 , 3 , 6 , 7 - tetrahydro - 1h - azepin - 1 - yl] - 1 - oxopropyl - valinyl - valine - methylester	-36.34
1hew	Lysozyme	3 Residues [NAG - NAG - NAG] N - acetyl - D - glucosamine (tri - N - acetylchitotriose)	-34.23

1hfc	Fibroblast collagenase	(N - (2 - hydroxamate methylene - 4 - methyl - pentoyl) phenylalanyl) methyl amine	-31.3
1hiv	HIV - 1 protease	(NOA - HIS - CHA - PSI [CH (OH) CH (OH)] VAL - ILE - APY) U75875	-75.34
1hvp	HIV - 1 protease	{ 3 - [(4 - amino - benzenesulfonyl) - isobutyl - amino] - 1 - benzyl - 2 - hydroxy - propyl } - carbamic acid tetrahydro - furan - 3 - yl ester	-52.64
1hri	Human rhinovirus 14	Sch 38057	-24.76
1hvi	HIV - 1 protease	A-77003 (c2 symmetry - based diol)	-57.51
1hvj	HIV - 1 protease	A-78791	-59.67
1hvk	HIV - 1 protease	A-76928	-57.73
1hvl	HIV - 1 protease	A-76889	-51.4
1hvr	HIV - 1 protease	[4R - (4 alpha , 5 alpha , 6 beta , 7 beta)] - hexahydro - 5 , 6 - dihydroxy - 1 , 3 - bis [2 - naphthyl - methyl] - 4 , 7 - bis (phenylmethyl) - 2H - 1 , 3 - diazepin - 2 - one	-54.26
1ida	HIV - 2 protease	Bila 1906	-49.63
1jao	Neutrophil collagenase	3- mercapto - 2 - benzylpropanoyl - ALA - GLY - NH2	-33.78
1kel	Antibody 28b4 fab fragment	Hapten	-41.56
1lgr	Glutamine synthetase	Adenosine monophosphate	-17.52
1mcb	Immunoglobulin lambda	N - acetyl - L - GLN - D - PHE - L - HIS - D - PRO - OH	-27.61
1mcf	Immunoglobulin lambda	N - acetyl - L - GLN - D - PHE - L - HIS - D - PRO - b - ALA - b - ALA - OH	-29.36
1mch	Immunoglobulin lambda	N - acetyl - L - GLN - D - PHE - L - HIS - D - PRO - b - ALA - b - ALA - OH	-29.36
1mcj	Immunoglobulin lambda	N - acetyl - D - PHE - L - HIS - D - PRO - NH2	-21.59
1mcs	Immunoglobulin lambda	N - acetyl - L - GLN - D - PHE - L - HIS - D - PRO - OH	-27.61
1mfe	Fab fragment (murine se155 - 4)	Dodecasaccharide { - 3) alpha - D - galactose (1 - 2) [alpha - D - abequose (1 - 3)] alpha - D - mannose (1 - 4) alpha - L - rhamnose (1 - }	-30.3
1mmb	Metalloproteinase - 8	4 - (N - hydroxyamino) - 2R - isobutyl - 2S - (2 - thienylthiomethyl) succinyl - L - phenylalanine - N - methylamide	-52.64
1mmq	Matrilysin	Hydroxamate inhibitor	-51.35
1mmr	Matrilysin	Sulfodiimine inhibitor	-33.6
1mnc	Neutrophil collagenase	Methylamino - phenylalanyl - leucyl - hydroxamic acid	-51.38
1mrk	Alpha - trichosanthin	Formycin	-25.84

1mtw	Factor Xa	[DX9056a] (+) - 2 - [4 -[[(S)- 1 - acetimidoyl - 3 - pyrrodinyl) oxy] - 3 - (7 - amidino - 2 - naphthyl) propionic acid	-42.15
1nnb	Neuraminidase	2 -deoxy - 2 , 3 - dehydro - N - acetly - neuraminic acid (DANA)	-22.83
1okl	Carbonic anhydrase ii	5 -(dimethylamino)- 1 - naphthalenesulfonamide (Dansylamide)	-34.43
1ola	Oligo - peptide binding protein	4 Residues [VAL - LYS - PRO - GLY]	-39.95
1phf	Cytochrome p450 - cam	2 - phenylimidazole	-25.1
1phg	Cytochrome p450 - cam	Metyrapone	-49.42
1ppc	Trypsin	4 Residues [NAS - GLY - APH - PIP] NAPAP	-36.85
1qbr	HIV - 1 protease	[4 R - (4 alpha, 5 alpha, 6 beta , 7 beta)- 3 , 3' - [[tetrahydro - 5 , 6 - dihydroxy - 2 - oxo- 4 , 7 - bis (phenylmethyl) - 1H - 1 , 3 - diazepine - 1 , 3 (2H)-diyl] bis (methylene)] bis [N - 2 - thiazolylbenzamide]	-60.32
1qbt	HIV - 1 protease	[4 R - (4 alpha , 5 alpha , 6 alpha , 7 alpha)] - 3 , 3' - { { tetrahydro - 5 , 6 - dihydroxy - 2 - oxo - 4 , 7 - bis (phenylmethyl) - 1H - 1 , 3 - diazepine - 1 , 3 (2H) - diyl] bis (methylene)} bis [N - 1H - benzimidazol - 2 - ylbenzamide]	-60.62
1qbu	HIV - 1 protease	[4 R - (1 alpha , 5 alpha , 7 beta)] - 3 - [(cycloprophylmethyl) hexahydro - 5 , 6 - dihydroxy - 2 - oxo - 4 , 7 - bis (phenylmethyl) - 1H - 1 , 3 - diazepin] methyl - 2 - thiazolylbenzamide	-58.43
1rbp	Retinol binding protein	Retinol	-38.33
1rgk	Ribonuclease T1(Rnase T1) mutant - E46Q	2' - adenylic acid	-24.59
1rgl	Ribonuclease T1(Rnase T1) mutant - E46Q	2' - guanylic acid	-25.27
1sln	Stromelysin - 1	L - 702,842 (N - (R - carboxy - ethyl) - alpha - (S) - (2 - phenylethyl) glycyL - L - arginine - N - phenylamide)	-37.89
1stp	Streptavidin	Biotin	-71.48
1tet	Te33 - Fab fragment of monoclonal antibody elicited against cholera toxin peptide 3 (CTP3)	Citrate	-35.41
1thl	Thermolysin	N -(1 -(2(R , S)- carboxy - 4 - phenylbutyl) cyclopentylcarbonyl)-(S) - tryptophan	-36.63
1tlp	Thermolysin	RHA - LEU - TRP (3 residues)	-43.12
1tmn	Thermolysin	1 - carboxy - 3 - phenylpropyl	-41.67
1tng	Trypsin	Aminomethylcyclohexane	-16.75

1tnh	Trypsin	4 - fluorobenzylamine	-19.22
1tni	Trypsin	4 - phenylbutylamine	-9.69
1tnj	Trypsin	2 - phenylethylamine	-6.15
1tnk	Trypsin	3 - phenylpropylamine	-8.5
1tnl	Trypsin	Tranylcypromine	-10.7
1uvs	Thrombin	Bm12.1700 complex (i11)	-30.81
1uvt	Thrombin	Bm12.1700 complex (i48)	-43.6
1zzz	Trypsin	SO2 - RON - GLY - 1PI	-29.27
2abh	Phosphate - binding protein	Phosphate ion	-37.13
2cgr	Igg2b	N -(P - cyanophenyl)- N ' - diphenylmethyl - guanidine - acetic acid	-41.53
2cmd	Malate dehydrogenase	Citrate	-26.1
2dbl	Fab' fragment of monoclonal antibody db3 (igg1, subgroup 2a, kappa 1)	5 - alpha - pregnane - 3 - beta - ol - hemisuccinate	-49.63
2er0	Endothiapepsin (Endothia aspartic proteinase)	8 Residues [IVA - HIS - PRO - PHE - HIS - CHS - LEU - PHE] 4 - amino - 5 - cyclohexyl - 3 - hydroxy - pentanoic acid	-36.51
2er6	Endothia aspartic proteinase	7 Residues [PRO - THR - GLU - PHE - PHE - ARG - GLU]	-41.22
2er9	Endothia aspartic proteinase	8 Residues [BOC - HIS - PRO - PHE - HIS - STA - LEU - PHE]	-44.56
2gbp	D - Galactose D - GLUCOSE BINDING PROTEIN (GGBP)	Beta - D - glucose	-43.36
2h4n	Carbonic anhydrase ii	5 - acetamido - 1 , 3 , 4 - thiadiazole - 2 - sulfonamide	-49.65
2ifb	Intestinal fatty acid binding protein (holo form) (I - FABP)	Palmitic acid	-30.98
2mcp	Immunoglobulin	Phosphocholine	-29.85
2r04	Rhinovirus 14 (HRV 14)	5 -(7 -(4 - (4 , 5 - dihydro - 2 - oxazolyl) phenoxy) heptyl) - 3 - methyl isoxazole compound IV	-35.51
2tmn	Thermolysin	3 Residues [PHO - LEU - NH2]	-33.6
3cla	Type III chloramphenicol acetyltransferase	Chloramphenicol	-28.18
3cpa	Carboxypeptidase a	GLY - TYR	-22.13
3er3	Endothia aspartic proteinase	5 Residues [BOC - PHE - HIS - CAL - LYS] (CP71,362)	-40.48
3ptb	Beta - trypsin	Benzamidine	-27.06
3tmn	Thermolysin	Val - TRP (VW)	-33.72
3ts1	Tyrosyl - transfer RNA	Tyrosinyl adenylate	-25.07
4cpa	Carboxypeptidase a	GLY	-47.38
4er1	Endothia aspartic proteinase	Pd125967	-37.83

4er4	Endothia aspartic proteinase	10 Residues [PRO - HIS - PRO - PHE - HIS - LEU - VAL - ILE - HIS - LYS] (H - 142)	-38.79
4sga	Proteinase A	5 residues [ACE - PRO - ALA - PRO - PHE]	-18.66
4tln	Thermolysin	L - leucyl - hydroxylamine	-21.23
4tmn	Thermolysin	CBZ - PHE == p ==- LEU - ALA (ZFPLA)	-58.16
5er2	Endothia aspartic proteinase	6 residues [BOC - PHE - HIS - AHS - LYS - PHE]	-37.49
5p21	C - h - ras p21 protein	Guanosine - 5' - (beta , gamma - imido) triphosphate (gpp np)	-30.35
5sga	Proteinase A	5 residues [ACE - PRO - ALA - PRO - TYR]	-16.26
5tmn	Thermolysin	4 residues [CBZ - PGL - LEU - LEU]	-45.89
6cpa	Carboxypeptidase a	O - [[(1R) - [[N - phenylmethoxycarbonyl) - L - alanyl] amino] ethyl] hydroxyphosphinyl] - L - 3 - phenyllactate	-65.77
6tim	Triosephosphate isomerase	Glycerol - 3 - phosphate	-18.31
6tmn	Thermolysin	4 residues [CBZ - PGL - OLE - LEU]	-28.83
7hvp	HIV - 1 protease	10 residues [ACE - SER - LEU - ASN - PHE - CH2 - PRO - ILE - VAL - OME]	-54.95

The selected complexes were downloaded in the PDB format, which while popular, tends to contain inherent atom typing errors. This problem is caused mainly by the inability of the file format to represent bond information. Another problem with the use of the PDB format was the lack of its ability to store the Gasteiger Huckel charges, used to represent the electrostatic component in the neural network inputs.

A new file format was required and the Tripos MOL2 format was chosen for meeting all the above requirements. The MOL2 format contains specific fields representing the Sybyl atom types used (which work best on the Sybyl 6.8 system used for all the molecular modeling done), atom coordinate data, as well as explicit bond information, as well as the capacity to store Gasteiger

Huckel charges, altogether in its @<TRIPOS>ATOM field (with exception of the bond data which is in the @<TRIPOS>BOND field) as depicted in the file extraction in Figure 4.2.1.

15	@<TRIPOS>ATOM					
16	1	C1	1.207	2.091	0.000	C.ar
17	2	C2	2.414	1.394	0.000	C.ar
18	3	C3	2.414	0.000	0.000	C.ar
19	4	C4	1.207	-0.697	0.000	C.ar
20	5	C5	0.000	0.000	0.000	C.ar
21	6	C6	0.000	1.394	0.000	C.ar
22	7	H1	1.207	3.175	0.000	H
23	8	H2	3.353	1.936	0.000	H
24	9	H3	3.353	-0.542	0.000	H
25	10	H4	1.207	-1.781	0.000	H
26	11	H5	-0.939	-0.542	0.000	H
27	12	H6	-0.939	1.936	0.000	H
28	@<TRIPOS>BOND					
29	1	1	2	ar		
30	2	1	6	ar		
31	3	2	3	ar		
32	4	3	4	ar		
33	5	4	5	ar		
34	6	5	6	ar		
35	7	1	7	1		
36	8	2	8	1		
37	9	3	9	1		
38	10	4	10	1		
39	11	5	11	1		
40	12	6	12	1		

Figure 4.2.1 : An extract of the MOL2 file format used

Each protein-ligand PDB file was thus first converted to the MOL2 format before the carving out of the active site using the Sybyl 6.8 software, and extraction of the ligand from the site, were performed. Each complex PDB file, was thus translated into two MOL2 files, one for the active site, and another for the ligand of each of the 128 data sets. The MOL2 files finally used after all the extraction and preparation can be referred to in Appendix C.

4.3 Input and Output Parameters

This section will describe the input parameters for the neural network experiments performed. As mentioned in Section 3.2, a fixed number of parameters are required to represent molecules of various shapes and of

various molecular weights. There needs to be a mapping of the required characteristics, in this case, electrostatic charge and atom type to a fixed number of variables. For this physico-chemical autocorrelation is used. Table 4.3.1 lists the parameters chosen for this. The input for each protein-ligand interaction was represented in a vector with 94 parameters, 47 containing information on the ligand and another 47 for the exact same characteristics of the active site. Table 4.3.1 lists the 47 parameters in the order used for the characterization of the ligand (and active site). Each protein-ligand complex is thus characterized by a single vector. The actual input vectors used are listed in Appendix D.

Table 4.3.1 : Parameters for characterization of ligands and active sites (47 from each)

Parameter No./ Ligand (Active Site)	Description (T = Atom Type, d = Distance, C = Charge)
1(48)	<i>Autocorrelative descriptor of T = N.pl3, N.am, N.4 and $d < 3$</i>
2(49)	<i>Autocorrelative descriptor of T = N.pl3, N.am, N.4 and $3 \leq d < 6$</i>
3(50)	<i>Autocorrelative descriptor of T = N.pl3, N.am, N.4 and $d \geq 6$</i>
4(51)	<i>Autocorrelative descriptor of T = N.1, O.2, N.ar, O.co2 and $d < 3$</i>
5(52)	<i>Autocorrelative descriptor of T = N.1, O.2, N.ar, O.co2 and $3 \leq d < 6$</i>
6(53)	<i>Autocorrelative descriptor of T = N.1, O.2, N.ar, O.co2 and $6 \leq d < 9$</i>
7(54)	<i>Autocorrelative descriptor of T = N.1, O.2, N.ar, O.co2 and $9 \leq d < 12$</i>
8(55)	<i>Autocorrelative descriptor of T = N.1, O.2, N.ar, O.co2 and $12 \leq d < 15$</i>
9(56)	<i>Autocorrelative descriptor of T = N.1, O.2, N.ar, O.co2 and $15 \leq d < 18$</i>
10(57)	<i>Autocorrelative descriptor of T = N.1, O.2, N.ar, O.co2 and $18 \leq d < 21$</i>
11(58)	<i>Autocorrelative descriptor of T = N.1, O.2, N.ar, O.co2 and $d \geq 21$</i>
12(59)	<i>Autocorrelative descriptor of T = N.2, N.3, O.3, O.spc and $d < 3$</i>
13(60)	<i>Autocorrelative descriptor of T = N.2, N.3, O.3, O.spc and $3 \leq d < 6$</i>
14(61)	<i>Autocorrelative descriptor of T = N.2, N.3, O.3, O.spc and $d \geq 6$</i>
15(62)	<i>Autocorrelative descriptor of T = C.2 and $d < 3$</i>
16(63)	<i>Autocorrelative descriptor of T = C.2 and $3 \leq d < 6$</i>
17(64)	<i>Autocorrelative descriptor of T = C.2 and $d \geq 6$</i>
18(65)	<i>Autocorrelative descriptor of T = C.3 and $d < 3$</i>
19(66)	<i>Autocorrelative descriptor of T = C.3 and $3 \leq d < 6$</i>

20(67)	<i>Autocorrelative descriptor of $T = C.3$ and $6 \leq d < 9$</i>
21(68)	<i>Autocorrelative descriptor of $T = C.3$ and $9 \leq d < 12$</i>
22(69)	<i>Autocorrelative descriptor of $T = C.3$ and $12 \leq d < 15$</i>
23(70)	<i>Autocorrelative descriptor of $T = C.3$ and $15 \leq d < 18$</i>
24(71)	<i>Autocorrelative descriptor of $T = C.3$ and $d \geq 18$</i>
25(72)	<i>Autocorrelative descriptor of $C > -0.5$ and $d < 3$</i>
26(73)	<i>Autocorrelative descriptor of $C > -0.5$ and $3 \leq d < 6$</i>
27(74)	<i>Autocorrelative descriptor of $C > -0.5$ and $6 \leq d < 9$</i>
28(75)	<i>Autocorrelative descriptor of $C > -0.5$ and $9 \leq d < 12$</i>
29(76)	<i>Autocorrelative descriptor of $C > -0.5$ and $d \geq 12$</i>
30(77)	<i>Autocorrelative descriptor of $-1 < C \leq -0.5$ and $d < 3$</i>
31(78)	<i>Autocorrelative descriptor of $-1 < C \leq -0.5$ and $3 \leq d < 6$</i>
32(79)	<i>Autocorrelative descriptor of $-1 < C \leq -0.5$ and $6 \leq d < 9$</i>
33(80)	<i>Autocorrelative descriptor of $-1 < C \leq -0.5$ and $9 \leq d < 12$</i>
34(81)	<i>Autocorrelative descriptor of $-1 < C \leq -0.5$ and $12 \leq d < 15$</i>
35(82)	<i>Autocorrelative descriptor of $-1 < C \leq -0.5$ and $15 \leq d < 18$</i>
36(83)	<i>Autocorrelative descriptor of $-1 < C \leq -0.5$ and $18 \leq d < 21$</i>
37(84)	<i>Autocorrelative descriptor of $-1 < C \leq -0.5$ and $d \geq 21$</i>
38(85)	<i>Autocorrelative descriptor of $C \leq -1$ and $d < 3$</i>
39(86)	<i>Autocorrelative descriptor of $C \leq -1$ and $3 \leq d < 6$</i>
40(87)	<i>Autocorrelative descriptor of $C \leq -1$ and $6 \leq d < 9$</i>
41(88)	<i>Autocorrelative descriptor of $C \leq -1$ and $9 \leq d < 12$</i>
42(89)	<i>Autocorrelative descriptor of $C \leq -1$ and $12 \leq d < 15$</i>
43(90)	<i>Autocorrelative descriptor of $C \leq -1$ and $15 \leq d < 18$</i>
44(91)	<i>Autocorrelative descriptor of $C \leq -1$ and $18 \leq d < 21$</i>
45(92)	<i>Autocorrelative descriptor of $C \leq -1$ and $21 \leq d < 24$</i>
46(93)	<i>Autocorrelative descriptor of $C \leq -1$ and $d \geq 24$</i>
47(94)	<i>Molecular Weight</i>

The Self-Organizing Map (SOM) used takes in the input parameters of each protein-ligand pair and returns an organized map of showing clusters of these pairs in the 2-D input space. The main graphical method used in analyzing the output from the SOMs was the unified distance matrix (U-matrix) [Ultsch et al., 1990], which shows the intensity of the input sample clustering through colour or shading. Figure 4.3.1 shows an example of a U-matrix.

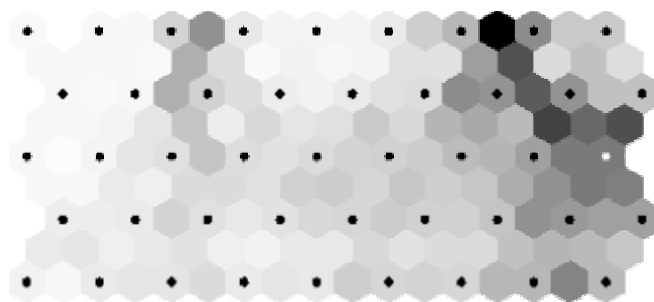


Figure 4.3.1 : An example of a U-matrix

The output of the neural network in the backpropagation phase is in the form of a normalized real number, as required by most neural network simulations, which after denormalization, will give the predicted binding affinity (in kJ/mol). Using these visualizations and numbers, analyses were carried out to identify patterns and correlations between molecular characteristics and binding affinity.

Chapter 5. Results and Analysis

This chapter will introduce the various results obtained from the experiments conducted in this thesis. Section 5.1 will first discuss the results obtained from the Self-Organizing Maps, through an analysis of a selected U-matrix. Its following subsections will then go into further detail on the clustering and the factors affecting it, namely the protein and ligand within the complex as individuals and as well the residues within the active site. Section 5.2 will then discuss the results obtained from the backpropagation experiments with comparison of error to two other methods, and do a comparative analysis with the results obtained from Section 5.1.

5.1 Self Organizing Maps (SOM)

The SOMs used to visualize the extent of protein-ligand complex clustering and distribution were run for 50, 250, 500, 750, 1000, 1250, 1500, 1750, 2000, 2250, 2500, 2750, 3000, 3250 and 3500 epochs respectively. The number of epochs chosen for testing was stopped at 3500 as the SOM visibly achieved stability at that point (runs for 3250 and 3500 epochs showed no further difference in the distribution and clustering of complexes). On each U-matrix, the inputs used (128 altogether) have been superimposed onto the neurons (map elements) to show which of the inputs have parameters most similar to the weights of the respective neurons onto which they have been superimposed. The stabilized U-matrix of the SOM at 3500 epochs is presented below in Figure 5.1.1.

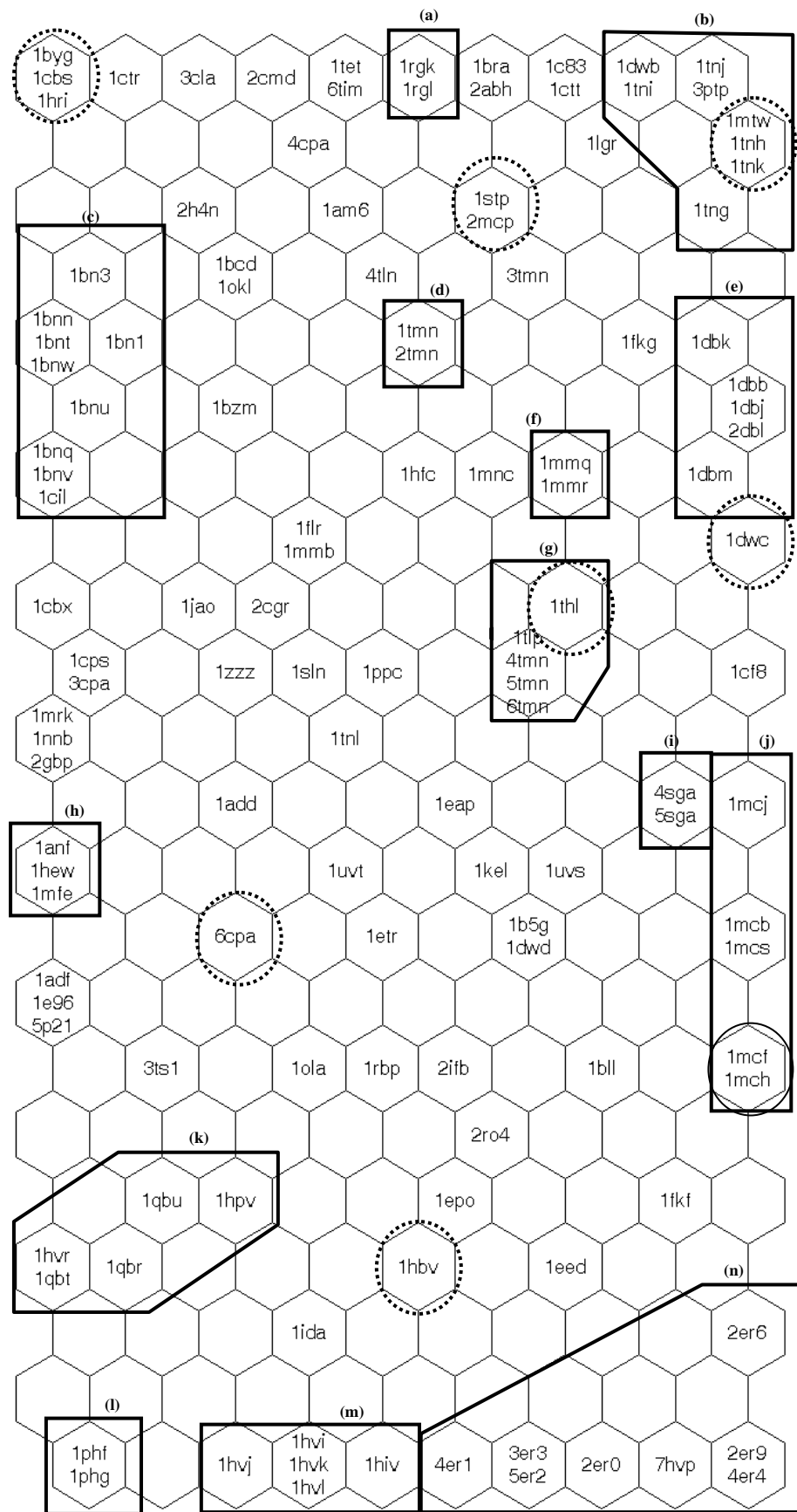


Figure 5.1.1: Stabilized U-Matrix of SOM at 3500 iterations

The occurrence of clusters throughout the SOM firstly confirms the even spread of input data used. This is vital in such experiments as unbiased data is a necessity to allow the following backpropagation method to be as generic as possible, enabling it to maximize its use to a variety of proteins and ligands rather than those belonging to a certain family or classification group.

5.1.1 Clustering Through Protein Active Site Similarity

While a protein structure can have more than a single active site, more often than not, the ligands have a tendency to bind to the same amino acid residues for the same physiological function. As such, the protein to which the ligand binds can be used as a clustering reference, similar proteins with similar physiological functions should be clustered together. This can be clearly seen in Figure 5.1.1. To illustrate this, further analysis will be carried out on the protein-ligand complexes listed in Table 5.1.1.

Table 5.1.1 : Breakdown of Protein-Ligand Complexes by cluster and majority protein

Group	Protein Type	Clustered Complexes
(a)	Ribonuclease T1(Rnase T1) mutant – E46Q	<i>Irgk, Irgl</i>
(b)	Trypsin	<i>Itng, Itnh, Itni, Itnj, Itnk, 3ptb(beta)</i>
(c)	Carbonic Anhydrase II	<i>Ibn1, Ibn3, Ibnn, Ibnq, Ibnt, Ibnu, Ibnv, Ibnw, Icil</i>
(d)	Thermolysin	<i>Itmn, 2tmn</i>
(e)	Fab' fragment of the monoclonal antibody db3	<i>Idbb, Idbj, Idbk, 2dbl, Idbm</i>
(f)	Matrilysin	<i>Immq, Immr</i>
(g)	Thermolysin	<i>4tmn, 5tmn, 6tmn, Itlh, Itlp</i>
(i)	Proteinase A	<i>4sga, 5sga</i>
(j)	Immunoglobulin Lambda	<i>Imcj, Imcb, Imcs, Imcf, Imch</i>
(k)	HIV – 1 Protease	<i>Ihvr, Iqbt, Iqbr, Iqbu, Ihpv</i>

(l)	Cytochrome p450 – cam	<i>Iphf, Iphg</i>
(m)	HIV – 1 Protease	<i>Ihiv, Ihvi, Ihvj, Ihvk, Ihvl</i>
(n)	Endothia Aspartic Proteinase	<i>2er0, 2er6, 2er9, 3er3, 4er1, 4er4, 5er2</i>

Clustering can be seen for the complexes above showing the influence of the protein involved. Analyzing individually the active site of each of the complexes, Table 6.1.2 details the common amino acid residues present within each group of protein types. The amino acid residues listed are contained in all the complexes listed in Table 51.2. The representation of the three character amino acid residue codes can be referred to in Appendix E.

Table 5.1.2 : Protein Families and Common Amino Acid Residues Present

Group	Protein-Ligand Complex PDB ID	Common Amino Acid Residues
(a)	<i>Irgk, Irgl</i>	<i>His40, Tyr38, Arg77, His92, Glu58</i>
(b)	<i>Itng, Itnh, Itni, Itnj, Itnk, 3ptb(beta)</i>	<i>Asp189, Gly219</i>
(c)	<i>Ibn1, Ibn3, Ibnn, Ibnq, Ibnt, Ibnu, Ibnv, Ibnw, Icil</i>	<i>His94, His96, His119, Thr199</i>
(d)	<i>Itmn, 2tmn</i>	<i>Asn112, Ala113, Arg203, Asp226, His231,</i>
(e)	<i>Idbb, Idbj, Idbk, 2dbl, Idbm</i>	<i>Asn35</i>
(f)	<i>Immq, Immr</i>	<i>Glu219, His228, Leu181, Pro238, Tyr240, Asn179</i>
(g)	<i>4tmn, 5tmn, 6tmn, Ithl, Itlp</i>	<i>Arg203, Asn112, Asp226</i>
(i)	<i>4sga, 5sga</i>	<i>His57, Asp102, Gly193, Ser195, Ser214, Gly216</i>
(j)	<i>Imcj, Imcb, Imcs, Imcf, Imch</i>	<i>Glu52</i>
(k)	<i>Ihvr, Iqbt, Iqbr, Iqbu, Ihpv</i>	<i>Asp25(A), Asp25(B)</i>
(l)	<i>Iphf, Iphg</i>	<i>Arg299, Asp297, Cys357, Arg112, His355</i>
(m)	<i>Ihiv, Ihvi, Ihvj, Ihvk, Ihvl</i>	<i>Asp25(B), Asp29(A), Asp29(B), Gly48(A)</i>
(n)	<i>2er0, 2er6, 2er9, 3er3, 4er1, 4er4, 5er2</i>	<i>Gly76, Thr219</i>

Table 5.1.2 above shows that the amino acids in the active sites, play a big part in the clustering, causing the complexes to either be found in the same neuron

or the same region (this difference is caused by differential conformation of the active sites as well as the ligand structure and properties). The numbers following the amino acid codes represent their position within the protein sequence and were included to emphasize that these clusters did not just contain similar amino acids, which can also cause grouping, but are the very same amino acids, and therefore are the same active sites. It should be further noted that these amino acids were present in all the complexes tabulated, and that more common amino acids occurred between individual acids in the same group, but not in all the group members.

5.1.2 Influence of Ligand Similarity on Clustering

It is also evident from both Figure 5.1.1 and Table 5.1.1 respectively, that while many of the complexes from similar proteins have clustered together, the ligands bound to them have an impact on the complex distribution as well. While groups (d) and (g), and (k) and (l), have ligands bound to the same respective proteins, the complexes have not been clustered together in Figure 5.1.1. This is due to differences in the ligands structure, with respect to their size (molecular weight) and atomic components that cause the complexes to be spread apart. For ease of viewing, the molecular coloring represents oxygens by red, nitrogens by blue, carbons by black, and sulphur atoms by yellow dots.

Tables 5.1.4 and 5.1.3 illustrate the effect of structural and physico-chemical difference of the ligands in groups (d) and (g) respectively, on the distribution of protein-ligand complexes within the U-matrix in Figure 5.1.1.

Table 5.1.3 : Ligands for groups (g) [Ligand structures from PDBSum]

Group (g) Ligands	
PDB ID	Structure
4tmn	
5tmn	
6tmn	
1tlp	

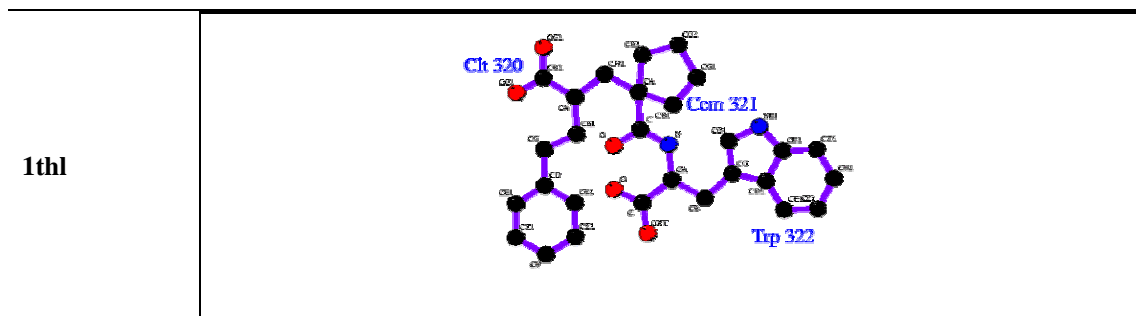
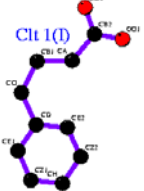
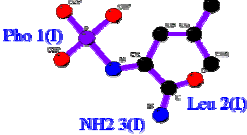


Table 5.1.4 : Ligands for group (d)[Ligand structures from PDBSum]

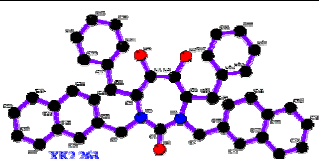
Group (d) Ligands	
PDB ID	Structure
1tmn (1CLT)	
2tmn	

Studying the structures in Table 5.1.3, it can be visibly seen that the ligands in group (g) are much larger than those in Table 5.1.4. They are as well very similar in their atomic composition, indicating they are probably analogues of the same molecule. 1thl, while being similar to the other atoms in the cluster, is the least similar having the lowest number of oxygen atoms and as well containing two ends of aromatic carbon atoms, for which the molecules were not characterized. This makes 1thl effectively a smaller molecule through its characterized vector. Even so, a combination of the similarities within the remaining characterized molecules and that of the active site bring 1thl into an

adjacent neuron, and part of the cluster. The ligands in Table 5.1.4 on the other hand, are much smaller in size, containing fewer oxygen, carbon and nitrogen atoms(except for 1CLT in 1tmn which contains no nitrogen atoms). This differs from the atoms in Table 5.1.3 which have a minimum of 5 oxygen atoms, and a maximum of 10 oxygen atoms (1tlp). Additionally, the dispersion of these oxygen and nitrogen atoms are much wider in the Table 5.1.3 atoms. All these factors, influencing the atom type (and therefore charge) and distance values therefore cause the separation in the two groups of complexes. The similarities in the active site composition have however caused their relative clustering.

The example given above illustrates one particular obvious difference in the chemical structure and properties of two groups of molecules. Sometimes however, a more defined grouping is required to enable differentiating a molecule that might bind well to an active site from one that does not. Tables 5.1.5 and 5.1.6 show two sets of molecules, both binding to the same active site, but yet separated on the SOM.

Table 5.1.5 : Ligands for group (k)[Ligand structures from PDBSum]

Group (k) Ligands	
PDB ID	Structure
1hvr	

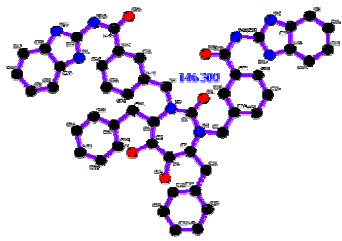
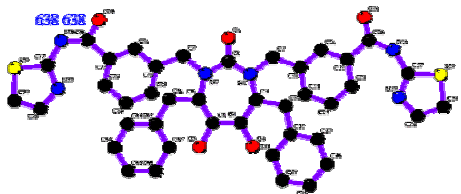
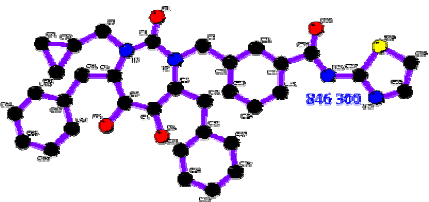
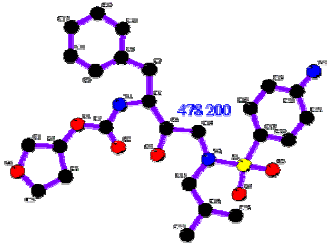
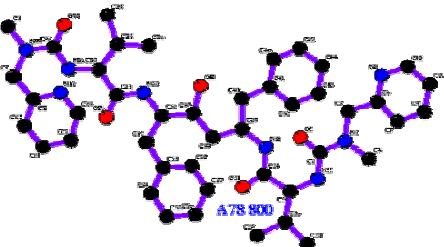
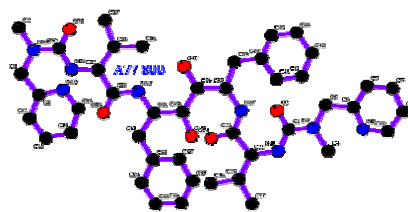
1qbt	
1qbr	
1qbu	
1hpv	

Table 5.1.6 : Ligands for group (m)[Ligand structures from PDBSum]

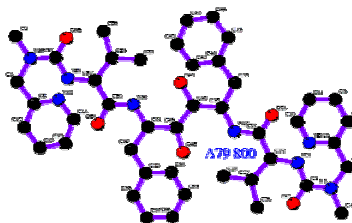
Group (m) Ligands

PDB ID	Structure
1hvj	

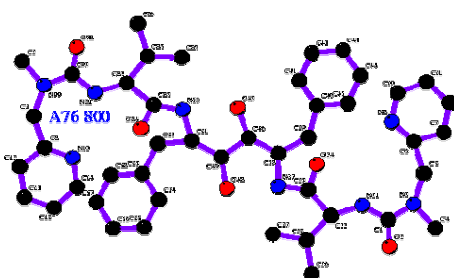
1hvi



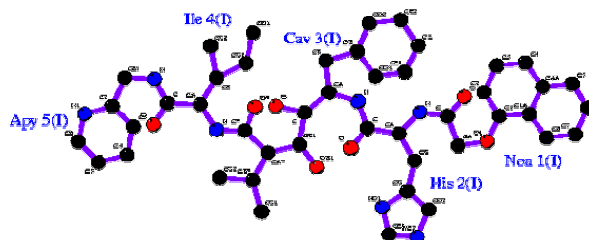
1hvk



1hvl



1hiv



Looking at the two groups of molecules does not show any obvious reason for their separation. More detailed analysis into the structures however, explain their separation on different levels. Looking closer at group (k), it is realized that all of the ligands, except for 1hvp, contain a substructure of a central ring of seven identical members of five carbon atoms and two amide nitrogen atoms shown in Figure 5.1.2.

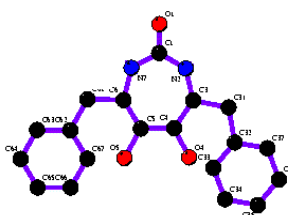


Figure 5.1.2: Common substructure found in group (k) ligands

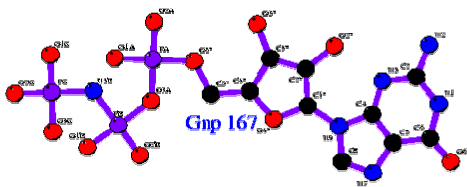
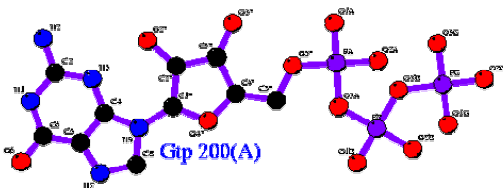
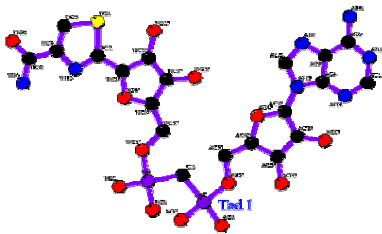
The presence of this substructure in the group (k) ligands and lack of it in those of group(m) explains their separation. 1 hpv dies not contain this substructure and is probably the reason it is relatively furthest from the group. Further analysis of the molecules in group (m) show their extremely similar structure indicating as well, that they are most probably derived from the same molecule. This similar structure has as well pushed 1hpv from its cluster to group (k), where the molecular weight of the molecules are more varied, and whose molecules do not comprise six membered rings containing aromatic nitrogen atoms.

5.1.3 Clustering Through Structural and Physico-Chemical Similarity

The examples given so far have all shown clustering of protein-ligand complexes and as well their separation using the same protein active sites. We need to show that the clustering works as well for situations where the proteins used are different and therefore have different active sites. Table 5.1.7 lists the ligands from group (h) of the U-matrix in Figure 5.1.1. These complexes have clustered on the same node even though they are all from different proteins. Analyzing their active sites, they are all seen to contain the common amino acids Serine, Glycine, Leucine, Valine, Aspartic acid, Lysine, Proline and Threonine. Even so, the active sites are nowhere as similar to one another as

could be in the case of various ligands binding to the same active site. As such further study into the ligand similarity brings us to the individual binding ligands listed in Table 5.1.7.

Table 5.1.7 : Ligands for group (h) [Ligand structures from PDBSum]

Group (h) Ligands	
PDB ID	Structure
5p21	
1e96	
1adf	

From Table 5.1.7, we can immediately see the similarity of the ligands, each having an almost identical cluster of oxygen atoms at one end and rings of nitrogen atoms on the other. An exception to the three is 1adf whose similarity

lies in the right half of the molecule being almost identical to the two other entire molecules. This similarity in shape and composition thus explains further why the three protein-ligand complexes were clustered on the same neuron by the SOM.

From this example, along with the previous two, it is seen that the SOM has clustered the complexes correctly, according to the parameters of the individual protein-ligand complexes. The examples have shown that this works for complexes with similar active sites and ligands, similar active sites and different ligands, as well as different active sites and similar ligands. The next section shall discuss the results obtained from the backpropagation experiments carried out.

5.2 Backpropagation Neural Networks

This section will list the results obtained from the various backpropagation experiments, and their performance (in terms of error) compared to predictions made by Multiple Linear Regression and the BLEEP method [Nobeli et al.,(2001)] of calculating binding affinity. The analysis of the results obtained will then be made with regard to the distribution of the neural network inputs on the SOM.

The backpropagation algorithm described in the Methodology chapter, was run for a series of topologies, each with a different number of neurons in the hidden layer, namely 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55 and 60 in order to find a

suitable neural network topology with the lowest error relative to the other topologies. Table 5.2.1 below lists the results, in terms of root mean square error (RMSE), mean absolute error (MAE), relative root mean square error (RRMSE) and relative mean absolute error (RMAE), when compared to the experimental results, from the runs of each of these topologies. Figure 5.2.1 and 5.2.2 in turn illustrate these differences in error graphically. The detailed results for each protein-ligand complex can be referred to in Appendix B.

Table 5.2.1 : Error calculated from backpropagation neural network with varying hidden neurons

Error Type	Number of Neurons in Hidden Layer											
	5	10	15	20	25	30	35	40	45	50	55	60
RMSE	20.80	20.68	18.70	16.99	29.63	20.53	17.46	19.00	22.91	21.23	25.45	34.43
RRMSE	147.23	146.36	132.37	120.25	209.72	145.30	123.59	134.51	162.15	150.24	180.16	243.69
MAE	15.74	16.35	14.09	12.93	16.52	14.98	13.43	14.63	15.60	14.49	16.11	19.72
RMAE	136.30	141.51	122.01	111.92	142.99	129.68	116.24	126.61	135.08	125.41	139.50	170.75

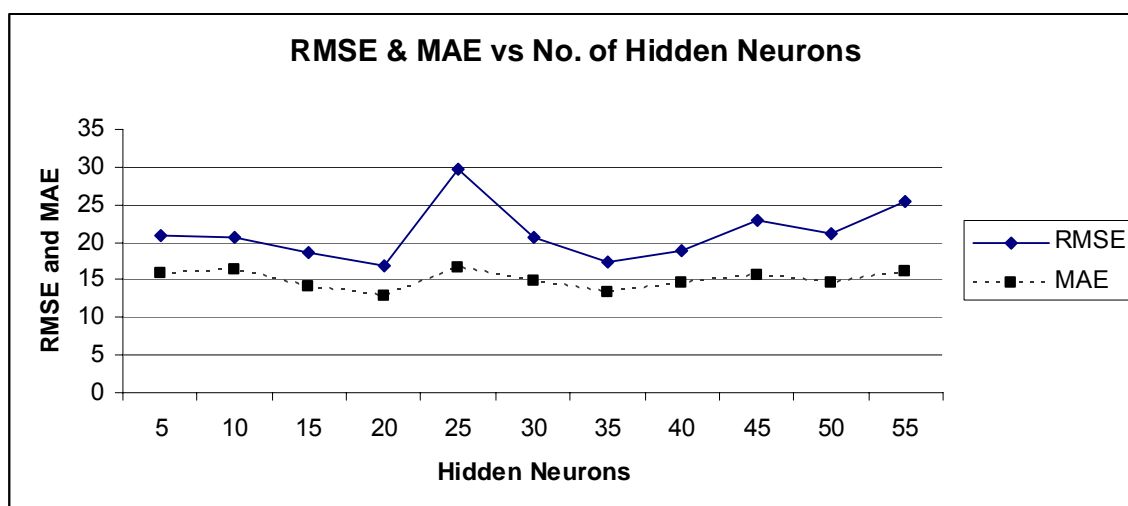


Figure 5.2.1 : Root Mean Squared Error and Mean Absolute Error vs No. of Hidden Neurons

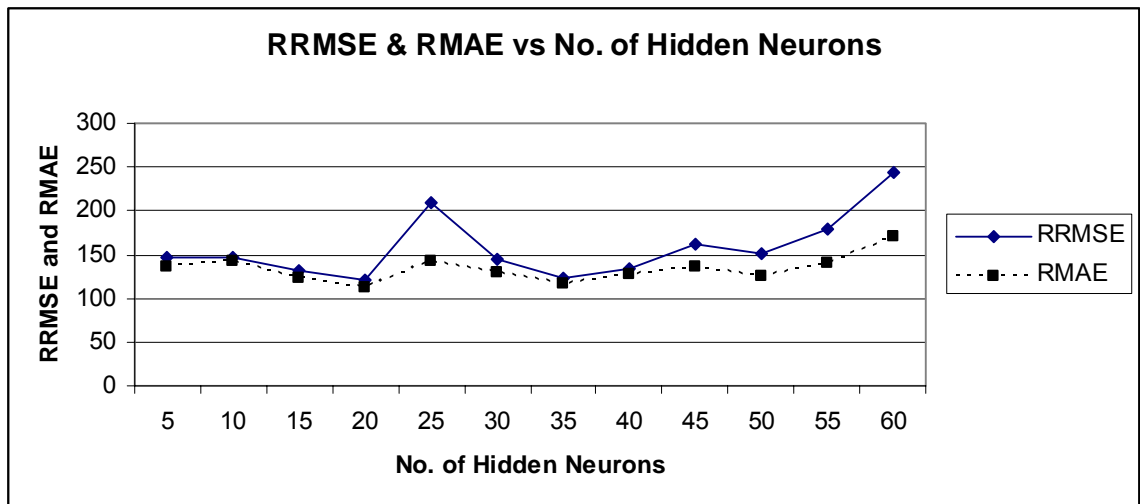


Figure 5.2.2 : Relative Root Mean Squared Error and Relative Mean Absolute Error vs No. of Hidden Neurons

As illustrated in the table and graphs above, the backpropagation neural network with 20 neurons in the hidden layer proved to be the best topology to use, giving the lowest MAE, RMAE, RMSE and as well RRMSE. As such, the rest of this chapter will focus on the 20 hidden neuron network.

In order to provide more perspective on the complexity and performance of the backpropagation neural network, further comparison was made with scores predicted through multiple linear regression methods, using the leave-one-out method and as well, with the binding scores predicted for the same dataset using the BLEEP [Nobeli et al.,(2001)] method. These comparative results are tabulated below in Table 5.2.2.

Table 5.2.2 : Comparison of backpropagation with Multiple Linear Regression and BLEEP methods

Error Type	Prediction Method		
	Backpropagation	Multiple Linear Regression	BLEEP
RMSE	16.99	30.20873	17.93821
RRMSE	120.25	213.8002	126.9602

MAE	12.93	20.74753	13.72516
RMAE	111.92	179.6167	118.8222

From Table 5.2.2, we see that the backpropagation method performs better than both the multiple linear regression method and the BLEEP method. The RMSE and MAE of the backpropagation method are 43.8% and 37.7% lower respectively when compared to that of the multiple linear regression method, and when compared to the BLEEP method, 5.3% and 5.8% lower. Even with these improvements over other methods, the backpropagation predictive method still has an RRMSE score of 120.25. This implies that the backpropagation method still performs worse than one that simply uses the average value of the samples [Setiono et al., (2002)]. As such, further analysis was carried out to find out at which point and which of the samples were responsible for pushing the RRMSE above 100.

To do this, the entire set of 128 samples was sorted according to ascending absolute error and the RMSE was calculated for the entire set but in increments of 10 samples each time. Therefore, the RRMSE was calculated for the top 10, 20, 30 all the way to 128 from the samples that gave the least error to those that produced the highest. Figure 5.2.3 shows how the RRMSE ranged with increasing number of samples from those with the smallest absolute error onward.

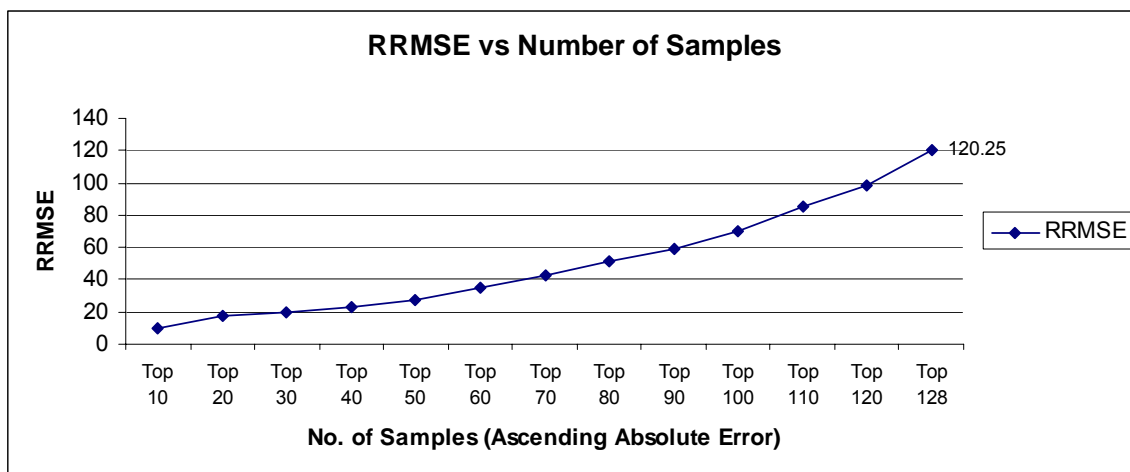


Figure 5.2.3 : Increase in RRMSE with increasing number of data samples

Figure 5.2.3 shows that for the top 93.75% of input samples (120 samples), the backpropagation neural network performs better than average, in fact for the first 100 samples, the RRMSE is just above 70% (70.04%). The eight bottom data samples with the highest absolute error responsible for the RRMSE moving above 100 are identified as 1dwc, 1mtw, 1hbv, 6cpa, 1hri, 1thl, 1stp, and 1mch with absolute errors of 52.40, 52.11, 43.65, 43.01, 37.21, 34.57, 34.10, and 34.01, respectively. These 8 protein-ligand complexes have been marked in Figure 5.1.1 by dotted circles. The inaccurate predictions imply that lack of similar training data is available to properly form correlations between the protein-ligand complexes and the active site. We thus expect these complexes to be alone in individual neurons in the U-matrix. This is however only true for the complexes 1dwc, 6cpa, and 1 hbv. 1hri, 1stp, 1mtw, 1mch all occur within neurons with clustered with other complexes, and while 1thl does occur in a neuron of its own, it has similarity to the complexes in group (g) as discussed earlier. As such, these complexes with similarity to others were studied further.

1hri was clustered with the complexes of 1byg and 1cbs. As similar amino acid residues were identified in their active sites, their individual ligand structures were studied. It was noticed from the atomic analysis of the three molecules that a large part of the molecules 1hri (6 out of 21) and 1byg (12 out of 35) were composed of aromatic carbon molecules. In the representation of the molecules using autocorrelation, these atom types were not accounted for due to their lack of hydrogen bonding/donating capacity. As such, the molecules were characterized based on their remaining atoms, to which similarity was found with 1cbs, which contained no aromatic carbon atoms. This collectively had the effect of bringing down the accuracy of their binding predictivity altogether. This is evident upon analysis of the other absolute binding accuracy of 1byg and 1cbs, which were in the bottom 19th and 29th positions, respectively. A similar situation occurred with 1mtw, which was composed of 33 atoms, 12 of which were aromatic carbons, thus causing it to be clustered with 1tnh and 1tnk, two other molecules comprising a majority of aromatic carbon atoms, clustered predominantly by active site similarities and similar lack of ligand characterization, which is expressed through low ligand parameter representation. This phenomenon, was expressed earlier in Section 5.1.2 with regard to the complex 1thl and its appearance adjacent but not within the same neuron as all the other complexes binding to the same protein active site, namely 1tlp, 4tmn, 5tmn, and 6tmn.

When a neural network is trained, it is as well dependent on the random initialization of the weights. This is usually seen in cases where the backpropagation neural network is trained over several different topologies and where there are mostly consistently good and bad predictions, due to the

random initialization of weights, some predictions are accurate at some runs, and inaccurate at others. This is as well made worse by a lack of firm similarity with its most similar training samples. 1mch is one such example. While being found in a cluster, the clustering is not as tight as required to qualify as similar training samples. This can be seen in the U-matrix in Figure 5.1.1, where the complexes in group (j) are found in three dispersed clusters on the right vertical edge of the matrix. This lack of 'closeness', together with the presence of aromatic carbons in the molecules lead the unsteady predictions of 1mch's binding affinity. While it produced a high error on the experiments with 20 neurons in the hidden layer, it did as well produce good predictions on the experiments run on topologies with 15 and 40 hidden neurons, with absolute errors of 1.48 and 3.39 respectively, markedly smaller than that of 34.01 in the run with 20 hidden neurons. While it is important to keep initialization of parameters random to avoid bias in the experiments, it is as well important to run each experiment with a particular topology several times, and take the eventual error to be recorded as the average of each set of runs for each single topology.

The results from the experiments run have not only justified the strengths of the method used, producing better predictions than those of regression methods and the previously developed BLEEP method, but have also highlighted potential improvements in the structure of the characterized data, and training methodology. It is needless safe to say that with increasing realization in the importance of the life sciences and medicinal research, that more data will be made available, increasing the accuracy of predictions.

Chapter 6. Conclusion and Future Work

This final chapter will bring the report to a close with an overall perspective of the study carried out, its strengths, weaknesses as well as its potential improvements. The later part of the chapter will then discuss the potential of the study and how far it can be carried out into the field of drug design, to potentially enable drugs to be designed altogether with a high certainty, fully computationally without ever having to move to chemical synthesis until the outcome is certain.

6.1 Conclusion

In this paper, a predictive method of calculating binding affinity was developed using a combination of ideas from artificial neural networks, biochemistry and physics. Kohonen Self Organising Maps (SOMs) were first used to categorize protein-ligand complexes with known binding affinity according to their similarity. This similarity was based on two biochemical phenomena of protein-ligand interactions being based on electrostatic charge, hydrogen bonding and molecular weight parameters. As such, physico-chemical autocorrelative methods were adopted to create a representation of three dimensional chemical structures, along with these characteristics. Once the SOMs succeeded in clustering the complexes accordingly, showing sensitivity to similar active sites and ligands individually as well as together, the clusters were subjected to a backpropagation neural network to train it such that it would be able to predict the binding affinity of a test sample. Due to the inability

to acquire large numbers of protein-ligand complexes with known binding affinity, more conventional train-validate-test sets could not be used. Instead, the leave-one-out system of training was utilized. These results were then compared to those obtained through Multiple Linear Regression and a previously published method, the Biomolecular Ligand Energy Evaluation Protocol (BLEEP). While the method developed throughout this report performed better than both compared methods, it nevertheless produced Relative Root Mean Square (RRMSE) and Relative Mean Absolute errors exceeding 100%, implying its inferiority to a naïve averaging method. Even so, the RRMSE scores were maintained above average up to the prediction of the top 100 samples.

The data extraction and representation through autocorrelation proved to successfully characterize each protein-ligand complex. Due to the large amount of different atom types, priority was given to atom types which either contributed to intermolecular interactions as hydrogen donors or as hydrogen acceptors. For a fair balance of representations, certain non-donors and non-acceptors were chosen as well. In the SOM clustering, the chosen parameters for molecular representation proved successful for the majority of complexes. However, there were nevertheless outliers which highlighted certain important potential improvements to the characterization. Ideally, all atom types would be characterized. Practically, this is made difficult through long training due to the large amounts of weighted edges needing tuning, and is made worse if large amounts of data are not available, to represent equally each facet of the parameter representation. In this report, aromatic carbons were not included

and this resulted in a *mis-characterization* of the respective molecules, eventually leading to poorer predictive scores. The choice of parameters is thus a very important one, as it proves to introduce time and space tradeoffs.

Another subject of question this report highlighted was that of random initialization of backpropagation neural networks. While its importance lies in the fairness of experimentation, by reducing biases to a minimum, it can as well introduce undesirable “side-effects” to the training of the network, hinting at the necessity to train a backpropagation network with the same training parameters in order to average out this randomness. This as well introduces a tradeoff of time versus accuracy.

On the whole, while the experiments run showed positive results, at the same time highlighting certain factors that should be given attention. Possible improvements to the current study as well as the future of such a methodology will be discussed in the following section.

6.2 Future Work

The ideal situation in any machine learning environment comes about when ample training data is available. In the drug design arena, this means more wet-lab experiments being carried out with regard to binding affinity to allow for more training samples to be obtained. As well, a good variance in training data is required. Many laboratories study a single protein-ligand interaction. This usually comprises a protein with known physiological effect and a library of

ligands to screen against the protein active site. While this may prove useful for local predictions, a more generic predictive methodology is desirable. A large diversity of active sites and ligands should thus be studied, and their binding affinity recorded, good and bad. With this large amount of data, better and more even characterization of molecules will be able to be studied and their varying pharmacophore structures obtained.

Large amounts of data alone are however insufficient for good predictivity of neural networks. The characterization of the molecules involved at the same time require further research, in order to develop a mapping of molecules, allowing for every atoms within it, along with its properties to be mapped onto two dimensions, while at the same time maintaining its uniqueness, uniformity, reversibility and translational as well as rotational invariance. This full characterization of a molecule, together with high powered computers to tolerate a large number of descriptive parameters might just make the road to perfect drug design a lot shorter.

As such, it would be highly desirable for researchers of different physiological perspectives to come together with their molecular interaction data and together create a universal, generic drug binding system, which will create not only a huge saving in drug production costs, but as well, a much shorter time for drugs from conception to reach the market at much lower costs.

References

- Andrea, T.A., Hooshmand, K. (1991). Applications Of Neural Networks In Quantitative Structure Activity Relationships Of Dihydrofolate Reductase Inhibitors, *Journal of Medicinal Chemistry*, 34: 2824 – 2836.
- Anzali S., Gasteiger, J., Holzgrabe, U., Polanski, J., Sadowski, J., Teckentrup, A., Wagener, M. (1998). The Use of Self-Organizing Neural Networks in Drug Design, *Perspectives in Drug Discovery and Design*, 9/10/11: 273-299.
- Anzali, S., Barnickel, G., Krug, M., Sadowski, J., Wagener, M., Gasteiger, J., and Polanski, J. (1996) The comparison of geometric and electronic properties of molecular surfaces by neural networks: Application to the analysis of corticosteroid binding globulin activity of steroids, *J. Comput.-Aided Mol. Design*, 10: 521-534.
- Anzali, S., Barnickel, G., Krug, M., Sadowski, J., Wagener, M., Gasteiger, J. (1996) Evaluation of Molecular Surface Properties Using a Kohonen Neural Network: Studies on Structure –Activity Relationships, In Devillers, J. (Ed.) *Neural Networks in QSAR and Drug Design*, Academic Press, London, pp. 209-222.
- Aoyama, T., Suzuki, Y., Ichikawa, H. (1990). Neural Networks Applied To Structure-Activity Relationships. *Journal of Medicinal Chemistry*, 33: 905 – 908.
- Balbes, L., Mascarella, S., Boyd, D.(1994), A perspective of modern methods in computer-aided drug design. In Lipkowitz, K. and Boyd, D.(eds.), *Reviews in Computational Chemistry*, VCH Publishers, 5: 337 – 370.
- Barlow, T.W.J. (1995). Self-organizing maps and molecular similarity, *J. Mol. Graph.*, 13: 24 – 27.
- Bauknecht, H., Zell, A., Bayer, H., Levi, P., Wagener M., Sadowski, J., Gasteiger, J. (1996). Locating Biologically Active Compounds in Medium-Sized Heterogeneous Datasets by Topological Autocorrelation Vectors: Dopamine and Benzodiazepine Agonists, *J. Chem Inf. Comput. Sci.*, 36: 1205 – 1213.
- Baxevanis A.D., Ouellette B.F.F. (1998) *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins*, Wiley-Interscience, USA.
- Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., Bourne, P.E. (2000).The Protein Data Bank. *Nucleic Acids Research*, 28: 235-242.
- Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E. (2000). The Protein Data Bank. *Nucleic Acids Research*, 28: 235-242

Bernstein, F.C., Koetzle, T.F., Williams, G.J.B., Meyer, F., Bryce, M.D., Rogers, J. R., Kennard, O., Shikanouchi, T., Tasumi, M.(1977) The Protein Data Bank: A Computer Based Archival File For Macromolecular Structures. *Journal of Molecular Biology*, Vol. 112:535 – 542.

Böhm, H.J. (1992a). The computer program LUDI: A new method for the de novo design of enzyme inhibitors. *Journal of Computer Aided Molecular Design*, 6: 61 – 78.

Böhm, H.J. (1992b). LUDI: rule-based automatic design of new substituents for enzyme inhibitor leads. *Journal of Computer Aided Molecular Design*, 6: 593 – 606.

Chen, Y.Z. and Zhi, D.G. (2001)Ligand-Protein Inverse Docking and Its Potential Use in Computer Search of Putative Protein Targets of a Small Molecule, *Proteins*, 43(2): 217-26.

Chen, Y.Z.(2001). Computer search of putative protein targets of a small molecule. *Biophys. J.*, 80: 497A.

Connolly, M. L. (1983). Solvent-accessible surfaces of proteins and nucleic acids. *Science*, 221: 709 – 713.

Connolly, M. (1986) Shape complementarity at the haemoglobin $\alpha_1\alpha_1$ subunit interface. *Biopolymers*. Vol. 25, pp. 1229 – 1247.

DesJarlais, R.L., Sheridan R.P., Dixon, J.S., Kuntz, I.D., Venkataraghavan, R. (1986) Docking flexible ligands to molecular receptors by molecular shape. *Journal of Medicinal Chemistry*, 29: 2149 – 2153.

Devillers, J., Domine, D., and Boethling, R. S. (1996) Use of a backpropagation neural network and autocorrelation descriptors for predicting the biodegradation of organic chemicals, in: Devillers, J.(ed.), *Neural Networks in QSAR and Drug Design*, Academic Press, London, pp. 65 - 82.

Eldridge, M. E., Murray, C.W., Auton, T.R., Paolinine, G.V. and Mee, R.P. (1997). Empirical scoring functions: I. The development of a fast empirical scoring function to estimate the binding affinity of ligands in receptor complexes *Journal of Computer Aided Molecular Design*, 11: 425 – 445.

Finn, P.W., Kavrakı, L.E.(1999), *Computational Approaches to Drug Design*, *Algorithmica*, Vol. 25: 347 – 371.

Foley, J.D., Van Dam, A. (1982). *Fundamentals of Interactive Computer Graphics*, Addison-Wesley, Reading, MA, p. 664.

Fraga, S., Parker, J.M., Pocok J.M.(1995). *Computer Simulations Of Protein Structures And Interactions*. New York, Springer Verlag.

- Gasteiger, J. and Zupan, J. (1993). Neural Networks in Chemistry, *Angew. Chem. Int. Ed. Engl.*, 32: 503-527.
- Gasteiger, J., Marsili, M. (1980). Iterative Partial Equalization of Orbital Electronegativity – A Rapid Access to Atomic Charges. *Tetrahedron*, 36: 3219-3228.
- Glen, R., Martin, G., Hill, A., Hyde, R., Wollard, P., Salmon, J., Buckingham, J., Robertson, A. (1995). Computer-Aided Design And Synthesis Of 5-Substituted Tryptamines And Their Pharmacology At The 5-HT_{1D} Receptor: Discovery Of Compounds With Potential Anti-Migraine Properties. *Journal of Medicinal Chemistry*, 38: 3566-3580.
- Goodfellow, J. M.; Pitt, W. R.; Smart, O. S.; Williams, M. A. (1995). New methods for the analysis of the protein-solvent interface, *Comput. Phys. Commun.*, 91, 321.
- Halperin, I., Ma, B., Wolfson, H., Nussinov, R. (2002) Principles of Docking: An Overview of Search Algorithms and a Guide to Docking Functions, *Proteins*, 47: 409 – 443.
- Hansch, C. (1969). A Quantitative Approach to Biochemical Structure-Activity Relationships *Acc. Chem. Res.*, 2: 232.
- Hasegawa, K., Matsuko, S., Arakawa, M., Funatsu, K. (2002). New Molecular Surface Based 3D-QSAR Method Using Kohonen Neural Network and 3-way PLS, *Computers and Chemistry*, 26: 583-589.
- Haykin, S. (1999). *Neural Networks : A Comprehensive Foundation*, Prentice Hall, New Jersey.
- Hecht-Nielsen, R. (1990). *Neurocomputing*. Addison-Wesley, Massachusetts: 138-147.
- Hemmer, M. C., Steinhauer, V., Gasteiger, J. (1999) Deriving the 3D structure of organic molecules from their infrared spectra, *Vibrational Spectroscopy*, 19: 151-164.
- Hertz, J., Krogh, A., and Palmer, R. G. (1991). *Introduction to the Theory of Neural Computation*. Addison-Wesley, California: 236-244.
- Hollas, B. (2003) An analysis of the autocorrelation descriptor for molecules, *J. Math. Chem.*, 33(2): 91-101.
- Honig B., Nicholls A. (1995). Classical electrostatics in biology and chemistry. *Science*, 268:1144-1149.

- Jones, G., Willet, P. (1995). Molecular Recognition Of Receptor Sites Using A Genetic Algorithm With A Description Of Desolvation. *Journal of Molecular Biology*, 254: 43 – 53.
- Jones, G., Willet, P., Glen, R.C., Leach, A.R., Taylor R. (1997). Development And Validation Of A Genetic Algorithm To Flexible Docking. *Journal of Molecular Biology*, 267: 727 – 748.
- Kaeding, W.W., Chu, C., Young, L. B., Weinstein, B., Butter, S.A.(1981). Selective alkylation of toluene with methanol to produce para-Xylene, *J. Catal*, 67: 159.
- Klebe, G., Mietzner, T.(1994). A fast and efficient method to generate biologically relevant conformations. *Journal of Computer Aided Molecular Design*, 8: 583 – 606.
- Kohonen, T. (1982) Self-Organized Formation of Topologically Correct Feature Maps, *Biol. Cybern.* 43: 59-69.
- Kohonen, T. (1995). Self-Organizing Maps, *Series in Information Sciences*, 30, Springer, Heidelberg.
- Kohonen, T.(1989). *Self-Organization and Associative Memory*. Springer-Verlag, Berlin.
- Kohonen, T., Hynninen, J., Kangas, J., Laaksonen, J. (1996). SOM_PAK: The self-organizing map program package. Report A31. Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland.
- Kontoyianni, M., McClellan, L.M., Sokol, G.S. (2004). Evaluation of Docking Performance: Comparative Data on Docking Algorithms. *Journal of Medicinal Chemistry*, 47: 558 – 565.
- Kuntz, I., Blaney, J., Oatley S., Langridge, R., Ferrin, T. (1982). A Geometric Approach To Macromolecule-Ligand Interactions. *Journal of Molecular Biology*, 161: 269-288.
- Laskowski R A, Hutchinson E G, Michie A D, Wallace A C, Jones M L, Thornton J M (1997). PDBsum: A Web-based database of summaries and analyses of all PDB structures. *Trends Biochem. Sci.*, **22**: 488-490.
- Laskowski R A, Luscombe N M, Swindells M B, Thornton J M (1996). Protein clefts in molecular recognition and function. *Protein Science*, **5**: 2438-2452.
- Leach, A.R., Kuntz, I.D. (1992). Conformational analysis of flexible ligands in macromolecular receptor sites. *Journal of Computational Chemistry*, 13: 730 – 748.
- Lee, R.H., Rose, G.D. (1985). Molecular recognition. I. Automatic identification of topographic surface features. *Biopolymers*, 24:1613 – 1627.

Lengauer, T. (1993). Algorithmic Research Problems In Molecular Bioinformatics. In *IEEE Proceedings of the 2nd Israeli Symposium on the Theory of Computing and Systems*: 177 – 192.

Lengauer, T., Rarey, M. (1996). Computational Methods For Biomolecular Docking. *Current Opinion In Structural Biology*, 6: 402 – 406.

Lennard-Jones, J.E.(1932). Processes of absorption and diffusion on solid particles, *Trans. Faraday Soc*, 28: 333.

Li, Q., Dong, L., Jia, R., Chen, X., Hu, Z., Fan, B.T., Development of a quantitative structure-property relationship model for predicting the electrophoretic mobilities, *Computers and Chemistry*, 26, 245-251(2002).

Lichtarge, O., Bourne H.R., Cohen, F.E. (1996). An Evolutionary Trace Method Defines Binding Surfaces Common to Protein Families. *Journal of Molecular Biology*, 257(2): 342 – 358.

Linnainmaa, S., Harwood, D., Davis, L.S. (1988). Pose determination of a three-dimensional object using triangle pairs. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 10(5): 634 – 646.

Marcotte, E.M., Pellegrini, M., Ng H.L., Rice D.W., Yeates T.O., Eisenberg, D. (1999). A combined algorithm for genome-wide prediction of protein function. *Nature*, 402(6757): 83 – 86.

Marcotte, E.M., Pellegrini, M., Ng H.L., Rice D.W., Yeates T.O., Eisenberg, D. (1999). Detecting Protein Function and Protein-Protein Interactions from Genome Sequences *Science*, 285(5428): 751 – 753.

Martin, Y.C. (1978). *Quantitative Drug Design, Medicinal Research Series*, Marcel Dekker, New York, 8.

McDonald, I. K., Thornton, J. M. (1994). Satisfying hydrogen bonding potential in proteins, *Journal of Molecular Biology*, 238: 777.

Mitchell, J. B. O., Laskowski, R. A., Alex, A. and Thornton, J. M. (1999) BLEEP – Potential of Mean Force Describing Protein-Ligand Interactions: I. Generating Potential, *Journal of Computational Chemistry*, 20: 1165-1176.

Moreau, G., Broto, P. (1980). Autocorrelation of molecular structures: Application to SAR studies, *Nouv. J. Chim.*, 4: 757-764.

Moriguchi, I. (1986). In *Structure-Activity Relationship – Quantitative Approaches*, Fujita, T.(ed.), Nankodo: Tokyo, Japan, Chapter 9.

Muegge, I. And Martin, Y.C. (1999) A general and fast scoring function for protein-ligand interactions: a simplified potential approach. *Journal of Medicinal Chemistry*, 42: 791-804.

Nobeli, I., Mitchell J. B. O., Alex, A., Thornton, J.M.(2001). Evaluation Of A Knowledge-Based Potential Of Mean Force For Scoring Docked Protein-Ligand Complexes, *Journal of Computational Chemistry*, 22: 673-688.

Oshiro, C.M., Kuntz, I.D. and Knegt, R.M.A. *Encyclopedia of Computational Chemistry*, P.v.R. Schleyer(ed.) (1998). Wiley: Chichester, West Sussex, U.K., pp.1606-1613.

Pearlman D.A., Case, D.A., Caldwell, J.W., Ross, W.R., Cheatham, T.E., III, DeBolt, S., Ferguson, D., Seibel, G., Kollman, P. (1995). AMBER, A Computer Program For Applying Molecular Mechanics, Normal Mode Analysis, Molecular Dynamics And Free Energy Calculations To Elucidate The Structures And Energies Of Molecules. *Comp. Phys. Commun.*, 91: 1-41.

Pearlman, D.A., Charifson, P.S. (2001). Are Free Energy Calculations Useful In Practice? A Comparison With Rapid Scoring Functions For The P38 MAP Kinase Protein System. *Journal of Medicinal Chemistry*, 44: 3417 – 3423.

Petit, J., Zupan, J., Leherter, L., Vercauteren, D. P. (2002). Application of a Kohonen neural network to the analysis of data regarding the alkylation of toluene with methanol catalyzed by ZSM-5 type zeolites, *Computers & Chemistry*, 26: 557-572.

Pitt, W. R., Murray-Rust, J., Goodfellow J. M. (1993). AQUARIUS2: Knowledge- based modeling of solvent sites around proteins, *J. Comput. Chem*, 14: 1007.

Polanski, J. (1997). The receptor-like neural network for modeling corticosteroid and testosterone binding globulins, *J. Chem. Inf. Comput. Sci.*, 37: 553-561.

Polanski, J. (1999). Self-Organising Neural Network for Modeling 3D QSAR of Colchicinoids, *Acta Biochimia Polonica*, 47(1/2000): 37-45.

Polanski, J.(1996) Neural nets for the simulation of the molecular recognition of MS-Windows environment, *J. Chem. Inf. Comput. Sci.*, 36: 694 – 705.

Polanski, J., Gasteiger, J., Wagener, M., Sadowski, J. (1998) The comparison of molecular surfaces by neural networks and its application to quantitative structure activity studies, *Quant. Struct.-Act. Relat.*, 17: 27-36.

Polanski, J., Ratajczak, A., Gasteiger, J., Galdecki, Z., Galdecka, E. (1997). Molecular modeling and X-Ray analysis for a structure-taste study of α -Arylsulfonylalkonic acids, *J. Mol. Struct.*, 407: 71-80.

Puvanendrapillai, D., Mitchell J.B.O. (2003) Protein Ligand Database(PLD): Additional Understanding Of The Nature And Specificity Of Protein-Ligand Complexes, *Bioinformatics*, 19: 1856-1857.

Rarey, M., Kramer, B., Lengauer, T., Klebe, G. (1996). A fast flexible docking method using an incremental construction algorithm. *Journal of Molecular Biology*, 261: 470 – 489.

Rocchia, W., Alexov, E., Honig, B. (2001). Extending The Applicability Of The Nonlinear Poisson-Boltzmann Equation : Multiple Dielectric Constants And Multivalent Ions. *Journal of Physical Chemistry B*, 105: 6507 – 6514.

Rogers D.F. (1985). Procedural Elements for Computer Graphics, McGraw-Hill, New York, p. 433.

Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1986). Learning Internal Representations by Error Propagation, in *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, Rumelhart, D.E. and McClelland, J.L.(eds.), MIT Press, Cambridge, MA, USA, 1: 318 – 362.

Schuur, J. H., Selzer, P., Gasteiger, J. (1996), The Coding of the Three-Dimensional Structure of Molecules by Molecular Transforms and Its Application to Structure-Spectra Correlations and Studies of Biological Activity, *J. Chem. Inf. Comput. Sci*, 36: 334-344.

Setiono, R., Wee, K. L., Zurada, J. M. (2002). Extraction of Rules From Artificial Neural Networks for Nonlinear Regression, *IEEE Transactions on Neural Networks*, Vol. 13(3): 564 – 577.

Shehadi, I.A. (2003) The Identification of Active Site Residues in Proteins Using the Microscopic Titration Curves(THEMATICS). *The Fourth Annual U.A.E. University Research Conference*.

Soltzberg, L. J., Wilkins, C. L. (1977). Molecular Transforms: A Potential Tool for Structure-Activity Studies. *J. Am. Chem. Soc.* 99: 439 – 443.

Song, X.H., Chen, Z., Yu, R.Q. (1993). Artificial Neural Networks Applied to Odor Classification For Chemical Compounds, *Computers & Chemistry* 17(3): 303-308.

Streitwieser, A.,(1961) *Molecular Orbital Theory for Organic Chemists*, Wiley, NY.

SYBYL® 6.9.2 Tripos Inc., 1699 South Hanley Rd., St., Louis, Missouri, 63144, USA.

Ultsch, A., and Siemon, H. P. (1990) Kohonen's self organizing feature maps for exploratory data analysis. In *Proc. INNC'90, Int. Neural Network Conf.*, Dordrecht, Netherlands, Kluwer, pp. 305-308.

Venkatachalam, C.M., Jiang, X., Oldfield, T., Waldan, M. (2003). Ligandfit: A Novel Method For The Shape Directed Rapid Docking Of Ligands To Protein Active Sites. *Journal of molecular Graphics Modelling*, 21: 289 – 307.

Weirl, R. (1931). Elektronenbeugung und Molekulbau. *Ann. Phys. (Leipzig)*, 8: 521-564.

Werbos, P. (1982). Applications of Advances in Nonlinear Sensitivity Analysis, in *System Modeling and Optimization: Proc. Of the Int. Federation for Information Processes*, Drenick, R. and Kozin, F. (eds.), Springer Verlag, New York, USA: 762 – 770.

Widrow, B. and Hoff, M.E. (1960). Adaptive Switching Circuits, *1960 IRE WESCON Convention Records*: 96-104.

Wiener, H.(1947). Structural determination of paraffin boiling points, *Journal of the American Chemical Society*, 69: 17.

Zhang, R., Liu, S., Liu, M., Hu, Z.(1997). Neural network-molecular descriptors approach to the prediction of properties of alkenes, *Computers and Chemistry*, 21(5): 335-341.

Zupan, J. and Gasteiger, J.(1999). Neural Networks in Chemistry and Drug Design, 2nd Edition, Wiley-Vch, Weinheim.

Zupan, J., Gasteiger, J. (1991).Neural networks: A new method for solving chemical problems or just a passing phase? *Anal. Chim. Acta*, pp. 1 – 30.

Zupan, J., Novic, M.(1997).., General type of a uniform and reversible representation of chemical structures, *Anal. Chim. Act.*,348: 409-418.

Appendix A.

Perl Script For Autocorrelation of Mol2 Files.

```
#!/usr/bin/perl
# Extract inter-atomic distances from a Tripos Mol2 file and
# autocorrelate
# the distances accordingly into a matrix file

# Read in the folder 'mol2used' present in the same directory

@files = ();
$folder = 'mol2used';

opendir(FOLDER,$folder);
@files = grep (!/^\.\/, readdir(FOLDER));
closedir(FOLDER);

foreach $file (@files){

$molfilename = "$folder/$file";

# Initialise check to 0
$check = 0;

# Open and handle the file
open(MOL2FILE, $molfilename);

%type = ();
%charge = ();
%x = ();
%y = ();
%z = ();

# Store file in variable
while ($molecule = <MOL2FILE>) {
    if ($molecule =~ /@<TRIPOS>ATOM/) {
        $check = 1;
    }
    if ($molecule =~ /@<TRIPOS>BOND/) {
        $check = 0;
    }
    if ($check == 1){
        unless ($molecule =~ /@<TRIPOS>ATOM/){

            $id = substr($molecule, 4,3);
            $xcoor = substr($molecule, 19,7);
            $ycoor = substr($molecule, 29,7);
            $zcoor = substr($molecule, 39,7);
            $atype = substr($molecule, 47,5);
            $acharge = substr($molecule, 70,7);
            $id =~ s/^\s*//;
            $xcoor =~ s/^\s*//;
            $ycoor =~ s/^\s*//;
            $zcoor =~ s/^\s*//;
            $atype =~ s/^\s*//;
            $acharge =~ s/^\s*//;
            $type{$id} = $atype;
        }
    }
}
}
```



```

    ++$vector[4];
}# elsif 1

elseif (($distance >= 6) && ($distance < 9)){
    ++$vector[5];
}# elsif 1

elseif (($distance >= 9) && ($distance < 12)){
    ++$vector[6];
}# elsif 1

elseif (($distance >= 12) && ($distance < 15)){
    ++$vector[7];
}# elsif 1

elseif (($distance >= 15) && ($distance < 18)){
    ++$vector[8];
}# elsif 1

elseif (($distance >= 18) && ($distance < 21)){
    ++$vector[9];
}# elsif 1

elseif ($distance >= 21){
    ++$vector[10];
}# elsif 2

}#if
}#elsif

```

If Atoms are N.2, N.3, O.3, or O.SPC

```

elseif ($type{$counter1} eq "N.2 " || $type{$counter1} eq "N.3 "
|| $type{$counter1} eq "O.3 " || $type{$counter1} eq "O.SPC"){
    if ($type{$counter2} eq "N.2 " || $type{$counter2} eq "N.3 "
|| $type{$counter2} eq "O.3 " || $type{$counter2} eq "O.SPC"){

        if ($distance < 3){
            ++$vector[11];
        }# if distance less than 3 A

        elseif (($distance >= 3) && ($distance < 6)){
            ++$vector[12];
        }# elsif 1

        elseif ($distance >= 6){
            ++$vector[13];
        }# elsif 2

    }#if
}#elsif

```

If Atoms are C.2

```

elseif ($type{$counter1} eq "C.2 "){
    if ($type{$counter2} eq "C.2 "){

```

```

if ($distance < 3){
  ++$vector[14];
}# if distance less than 3 A

elseif (($distance >= 3) && ($distance < 6)){
  ++$vector[15];
}# elseif 1

elseif ($distance >= 6){
  ++$vector[16];
}# elseif 2

}#if
}#elseif

```

If Atoms are C.3

```

elseif ($type{$counter1} eq "C.3 "){
  if ($type{$counter2} eq "C.3 "){

    if ($distance < 3){
      ++$vector[17];
    }# if distance less than 3 A

    elseif (($distance >= 3) && ($distance < 6)){
      ++$vector[18];
    }# elseif 1

    elseif (($distance >= 6) && ($distance < 9)){
      ++$vector[19];
    }# elseif 1

    elseif (($distance >= 9) && ($distance < 12)){
      ++$vector[20];
    }# elseif 1

    elseif (($distance >= 12) && ($distance < 15)){
      ++$vector[21];
    }# elseif 1

    elseif (($distance >= 15) && ($distance < 18)){
      ++$vector[22];
    }# elseif 1

    elseif ($distance >= 18){
      ++$vector[23];
    }# elseif 2

  }#if
}#elseif

```

End Of Atom Type Autocorrelation

Now For The Charges

```

if ($charge{$counter1} > -0.5){
  if ($charge{$counter2} > -0.5){

```

```

        if ($distance < 3){
            $vector[24] = sprintf("%.3f", $vector[24] +
abs($charge{$counter1} * $charge{$counter2}));
        }# if distance less than 3 A

        elseif (($distance >= 3) && ($distance < 6)){
            $vector[25] = sprintf("%.3f", $vector[25] +
abs($charge{$counter1} * $charge{$counter2}));
        }# elseif 1

        elseif (($distance >= 6) && ($distance < 9)){
            $vector[26] = sprintf("%.3f", $vector[26] +
abs($charge{$counter1} * $charge{$counter2}));
        }# elseif 1

        elseif (($distance >= 9) && ($distance < 12)){
            $vector[27] = sprintf("%.3f", $vector[27] +
abs($charge{$counter1} * $charge{$counter2}));
        }# elseif 1

        elseif ($distance >= 12){
            $vector[28] = sprintf("%.3f", $vector[28] +
abs($charge{$counter1} * $charge{$counter2}));
        }# elseif 2

    } #if
} #if

elseif (($charge{$counter1} <= -0.5) && ($charge{$counter1} > -
1)){
    if(($charge{$counter2} <= -0.5) && ($charge{$counter2} > -1)) {

        if ($distance < 3){
            $vector[29] = sprintf("%.3f", $vector[29] +
($charge{$counter1} * $charge{$counter2}));
        }# if distance less than 3 A

        elseif (($distance >= 3) && ($distance < 6)){
            $vector[30] = sprintf("%.3f", $vector[30] +
($charge{$counter1} * $charge{$counter2}));
        }# elseif 1

        elseif (($distance >= 6) && ($distance < 9)){
            $vector[31] = sprintf("%.3f", $vector[31] +
($charge{$counter1} * $charge{$counter2}));
        }# elseif 1

        elseif (($distance >= 9) && ($distance < 12)){
            $vector[32] = sprintf("%.3f", $vector[32] +
($charge{$counter1} * $charge{$counter2}));
        }# elseif 1

        elseif (($distance >= 12) && ($distance < 15)){
            $vector[33] = sprintf("%.3f", $vector[33] +
($charge{$counter1} * $charge{$counter2}));
        }# elseif 1

        elseif (($distance >= 15) && ($distance < 18)){
            $vector[34] = sprintf("%.3f", $vector[34] +
($charge{$counter1} * $charge{$counter2}));
        }# elseif 1
    }
}

```

```

        elseif (($distance >= 18) && ($distance < 21)){
            $vector[35] = sprintf("%.3f", $vector[35] +
($charge{$counter1} * $charge{$counter2}));
        }# elseif 1

        elseif ($distance >= 21){
            $vector[36] = sprintf("%.3f", $vector[36] +
($charge{$counter1} * $charge{$counter2}));
        }# elseif 2

    }#if
}#elseif

elseif ($charge{$counter1} < -1){
    if ($charge{$counter2} < -1){

        if ($distance < 3){
            $vector[37] = $vector[37] + ($charge{$counter1} *
$charge{$counter2});
            $vector[37] = sprintf("%.3f", $vector[37]);
        }# if distance less than 3 A

        elseif (($distance >= 3) && ($distance < 6)){
            $vector[38] = sprintf("%.3f", $vector[38] +
($charge{$counter1} * $charge{$counter2}));
        }# elseif 1

        elseif (($distance >= 6) && ($distance < 9)){
            $vector[39] = sprintf("%.3f", $vector[39] +
($charge{$counter1} * $charge{$counter2}));
        }# elseif 1

        elseif (($distance >= 9) && ($distance < 12)){
            $vector[40] = sprintf("%.3f", $vector[40] +
($charge{$counter1} * $charge{$counter2}));
        }# elseif 1

        elseif (($distance >= 12) && ($distance < 15)){
            $vector[41] = sprintf("%.3f", $vector[41] +
($charge{$counter1} * $charge{$counter2}));
        }# elseif 1

        elseif (($distance >= 15) && ($distance < 18)){
            $vector[42] = sprintf("%.3f", $vector[42] +
($charge{$counter1} * $charge{$counter2}));
        }# elseif 1

        elseif (($distance >= 18) && ($distance < 21)){
            $vector[43] = sprintf("%.3f", $vector[43] +
($charge{$counter1} * $charge{$counter2}));
        }# elseif 1

        elseif (($distance >= 21) && ($distance < 24)){
            $vector[44] = sprintf("%.3f", $vector[44] +
($charge{$counter1} * $charge{$counter2}));
        }# elseif 1

        elseif ($distance >= 24){
            $vector[45] = sprintf("%.3f", $vector[45] +
($charge{$counter1} * $charge{$counter2}));
        }# elseif 2

```

```
        }#if
    }#elsif

##### End Of Charge Code #####

#Close the file
close MOL2FILE;

print "\n\n";
print $file;
print "\n";
printf "[%vector]";

}#endfor
exit;
```

Appendix B

Binding Scores For All Topologies Including BLEEP Scores and Multiple Linear Regression(MLR)

PDB ID	Experimental	5 Neurons	10 Neurons	15 Neurons	20 Neurons	25 Neurons	30 Neurons	35 Neurons	40 Neurons
1add	-38.45	-17.813	-24.588	-35.095	-27.6	-14.926	-36.318	-13.605	-25.779
1adf	-26.11	-78.654	-63.192	-41.071	-41.95	-66.286	-48.386	-50.825	-46.714
1am6	-24.7	-52.367	-22.898	-40.234	-33.304	-31.135	-33.255	-31.455	-31.081
1anf	-31.13	-18.944	-31.641	-12.339	-39.349	-38.424	-18.368	-35.483	-31.53
1b5g	-45.64	-43.35	-51.414	-52.523	-34.263	-45.347	-61.198	-68.609	-34.151
1bcd	-22.25	-43.538	-31.287	-28.611	-20.103	-31.467	-31.945	-29.056	-40.298
1bll	-38.22	-38.723	-23.996	-14.434	-10.508	16.271	-7.5847	-20.744	-23.364
1bn1	-53.29	-46.282	-60.151	-58.733	-63.208	-56.388	-59.704	-52.657	-58.38
1bn3	-56.42	-20.991	-21.833	-32.217	-26.438	-32.207	-39.049	-48.873	-37.413
1bnn	-57.05	-56.104	-48.143	-55.866	-52.783	-52.94	-56.173	-56.2	-52.492
1bnq	-54.14	-51.701	-53.779	-57.309	-53.347	-51.569	-50.235	-55.043	-45.851
1bnt	-55.92	-53.193	-86.912	-67.663	-61.33	-68.752	-70.057	-61.601	-58.432
1bnu	-55.33	-64.599	-63.54	-56.946	-61.639	-55.258	-62.129	-67.676	-52.533
1bnv	-50.03	-55.757	-52.513	-53.091	-57.549	-51.34	-55.518	-53.497	-55.762
1bnw	-51.8	-61.765	-63.982	-52.333	-55.412	-61.913	-63.832	-53.801	-55.981
1bra	-10.44	-16.24	-34.8	-28.357	-35.523	-25.125	-19.171	-27.159	-35.92
1byg	-56.6	-12.754	-21.632	-19.609	-31.146	-39.994	-25.654	-19.03	-45.458
1bzm	-34.4	-34.31	-61.284	-48.3	-46.461	-47.316	-50.565	-50.786	-54.56
1c83	-19.24	-42.207	-46.604	-41.174	-52.653	-31.546	-70.481	-44.925	-51.566
1cbs	-41.08	-18.575	-10.087	-52.792	-25.542	-35.142	-58.761	-40.624	-33.576
1cbx	-36.23	-42.578	-57.132	-46.458	-37.983	-40.905	-38.823	-42.658	-39.974
1cf8	-34.41	-29.273	-25.866	-17.33	-22.677	-39.083	-39.3	-14.778	-28.095
1cil	-53.8	-53.914	-49.831	-51.103	-51.065	-53.216	-42.637	-55.146	-52.768
1cps	-37.99	-31.215	-17.739	-35.092	-28.184	-29.135	-33.972	-30.096	-34.498
1ctr	-24.45	-15.475	-3.5954	-12.29	-25.872	-30.338	-13.792	-16.185	-24.373
1ctt	-25.79	-27.684	-14.322	-38.149	-28.106	-22.781	-27.217	-25.006	-20.648
1dbb	-51.38	-35.562	-39.893	-49.141	-40.258	-37.64	-33.093	-44.674	-38.445
1dbj	-43.83	-58.883	-53.172	-57.959	-52.886	-48.769	-40.546	-52.909	-51.238
1dbk	-46.22	-55.098	-40.868	-44.436	-54.16	-49.161	-43.297	-46.356	-33.842
1dbm	-53.9	-28.166	-39.786	-35.013	-23.368	-47.347	-68.93	-52.392	-45.773
1dwb	-16.66	-57.671	-20.713	-36.534	-30.308	-25.207	-23.333	-28.854	-9.9183
1dwc	-42.27	5.3896	10.042	-36.581	-94.669	-278.44	-15.898	-23.629	-16.489
1dwd	-46.62	-56.12	-46.076	-11.588	-43.715	-18.877	-2.224	-28.427	-44.135
1e96	-29.78	-14.948	-26.604	-40.887	-31.969	-40.55	-36.809	-42.188	-35.545
1eap	-35.42	-3.2036	-39.976	-29.529	-26.813	-25.935	-38.058	-27.093	-23.44
1eed	-27.39	-20.113	-51.077	-35.547	-41.677	-36.793	-35.97	-39.282	-38.87
1epo	-45.41	-42.668	-55.284	-33.64	-46.915	-53.262	-35.586	-30.605	-28.909
1etr	-42.28	-92.222	-73.562	-47.399	-62.877	-34.112	-119.95	-55.69	-53.183
1fkf	-55.37	-9.0194	-48.565	22.204	-60.05	-9.2292	-93.167	-31.734	-14.875
1fkg	-36.86	-40.89	-35.423	-26.548	-53.52	-35.866	-43.355	-13.616	-24.99
1flr	-26.55	-38.734	-42.167	-47.928	-49.373	-64.837	-65.673	-64.566	-38.663
1hbv	-36.34	-83.825	-74.62	-65.82	-79.99	-53.605	-69.066	-84.569	-60.03
1hew	-34.23	-43.594	-14.925	-31.279	-33.637	-9.096	-42.544	-30.077	-61.264
1hfc	-31.3	-52.149	-52.126	-51.268	-39.904	-44.95	-35.677	-52.542	-42.083
1hiv	-75.34	-41.187	-33.426	-55.229	-46.604	-44.197	-48.322	-49.722	-40.393
1hpv	-52.64	-32.323	-30.411	-11.856	-36.491	-44.991	-45.086	-50.347	-53.661

PDB ID	Experimental	5 Neurons	10 Neurons	15 Neurons	20 Neurons	25 Neurons	30 Neurons	35 Neurons	40 Neurons
1hri	-24.76	-29.812	-71.432	-43.656	-61.974	-47.649	-74.689	-68.386	-85.035
1hvi	-57.51	-53.099	-56.326	-54.353	-60.858	-51.075	-60.251	-53.899	-50.87
1hvj	-59.67	-55.278	-43.319	-54.321	-68.385	-49.683	-75.563	-52.7	-37.78
1hvk	-57.73	-59.885	-35.619	-53.52	-61.486	-62.162	-59.216	-69.377	-66.567
1hvl	-51.4	-52.247	-75.314	-55.957	-65.139	-58.8	-59.299	-64.512	-55.566
1hvr	-54.26	-75.434	-59.498	-68.5	-57.66	-63.58	-60.918	-43.941	-60.838
1ida	-49.63	-71.584	-72.936	-58.704	-70.868	-61.516	-73.048	-57.565	-68.082
1jao	-33.78	-39.787	-25.548	-33.149	-50.057	-38.382	-30.419	-35.959	-32.18
1kel	-41.56	-37.742	-40.252	-25.908	-66.088	-38.096	-58.211	-25.835	-33.94
1lgr	-17.52	-28.628	-20.555	-24.085	-9.4335	-2.6302	-26.799	-22.32	-19.105
1mcb	-27.61	-45.633	-11.1	-50.832	-28.375	-30.825	-29.11	-61.806	9.2267
1mcf	-29.36	-69.414	-56.777	-35.434	-35.809	-56.662	-29.592	-55.931	-49.756
1mch	-29.36	-37.012	3.0141	-27.879	-63.37	-54.991	2.6697	-2.6681	-25.962
1mcj	-21.59	-7.2015	-20.107	-9.6037	-22.394	-3.3885	-30.513	-29.576	-27.743
1mcs	-27.61	-24.502	12.555	-25.558	-25.509	-7.0073	-36.602	-24.177	-31.683
1mfe	-30.3	-31.514	-71.387	-40.91	-35.665	-67.639	-36.668	-24.038	7.7542
1mmb	-52.64	-37.445	-45.403	-41.592	-50.868	-38.262	-49.605	-44.782	-49.582
1mmq	-51.35	-37.931	-40.597	-32.5	-34.411	-37.816	-34.957	-35.642	-35.208
1mmr	-33.6	-47.993	-52.079	-56.858	-48.347	-52.733	-52.486	-53.853	-56.175
1mnc	-51.38	-27.983	-24.77	-43.683	-37.612	-29.422	-25.692	-22.46	-25.639
1mrk	-25.84	-27.453	-26.919	-21.349	-18.293	-14.43	-19.237	-48.682	-20.954
1mtw	-42.15	2.3649	-29.313	-33.891	9.9597	3.6168	-2.1096	-9.7298	-6.0201
1nnb	-22.83	-41.573	-42.624	-36.626	-49.089	-27.63	-42.462	-34.48	-58.289
1okl	-34.43	-34.604	-27.052	-28.912	-31.679	-35.907	-27.96	-27.516	-22.572
1ola	-39.95	-76.147	-56.71	-57.35	-64.491	-76.535	-40.424	-40.991	-66.904
1phf	-25.1	-51.964	-48.295	-52.502	-45.201	-49.682	-74.871	-40.697	-50.212
1phg	-49.42	-26.535	16.356	-45.926	-37.434	-31.999	-56.025	-45.64	-4.181
1ppc	-36.85	-30.935	-21.566	-43.589	-26.356	-40.667	-58.1	-50.905	-49.49
1qbr	-60.32	-30.52	-51.545	-53.59	-55.489	-59.227	-55.07	-55.227	-55.873
1qbt	-60.62	-104.23	-63.732	-60.464	-76.153	-67.561	-70.744	-57.072	-90.596
1qbu	-58.43	-56.562	-60.635	-56.097	-62.472	-52.846	-50.855	-60.429	-52.13
1rbp	-38.33	-1.8502	-69.014	-23.02	-14.441	-34.455	6.0784	-45.284	-58.965
1rgk	-24.59	-26.455	-30.226	-28.792	-29.051	-31.595	-33.114	-31.654	-30.386
1rgl	-25.27	-26.004	-58.729	-42.326	-26.394	-30.293	-25.848	-83.243	-30.89
1sln	-37.89	-39.615	-62.81	-71.528	-41.658	-46.603	-56.035	-51.77	-44.916
1stp	-71.48	-42.917	-34.488	-27.212	-37.384	-12.659	-24.016	-32.998	-48.594
1tet	-35.41	-38.52	-31.926	-22.557	-18.125	-10.111	-13.505	-30.675	-11.813
1thl	-36.63	-69.215	-46.737	-78.633	-71.202	-67.181	-64.784	-67.896	-63.445
1tlp	-43.12	-50.163	-78.174	-49.599	-45.652	-35.781	-62.337	-46.654	-36
1tmn	-41.67	-22.987	-15.77	-28.774	-26.058	-15.352	-20.795	-22.523	-24.869
1tng	-16.75	-24.004	-28.635	-20.096	-25.824	-15.84	-16.381	-22.954	-18.521
1tnh	-19.22	-5.0782	-8.8622	-13.974	-10.176	-19.668	-14.753	-10.761	-11.856
1tni	-9.69	-6.8783	-11.506	-10.007	-5.6448	-6.8533	-7.3066	-6.5328	-10.269
1tnj	-6.15	-11.208	-8.0308	-7.4591	-11.105	-8.8197	-13.298	-13.785	-9.4889
1tnk	-8.5	-13.004	-17.171	-16.291	-15.763	-17.21	-23.934	-17.007	-26.539
1tnl	-10.7	-14.515	-25.108	-25.911	-16.136	-21.336	-16.706	-11.073	-7.7219
1uvs	-30.81	-66.113	-56.995	-38.951	-52.39	-27.449	-46.3	-11.996	-37.208
1uvt	-43.6	-6.58	-15.94	18.385	-26.427	68.102	23.93	-4.9906	-8.3968
1zzz	-29.27	-19.173	-10.376	0.27584	-15.102	-34.682	-17.412	-26.552	-66.705
2abh	-37.13	-19.196	-46.447	-24.677	-28.749	-44.898	-19.026	-44.736	-24.548
2cgr	-41.53	-28.47	-39.592	-12.584	-30.777	-7.3739	-37.083	-13.555	-23.641
2cmd	-26.1	-35.65	-31.897	-44.056	-31.402	-23.739	-29.371	-54.835	-24.4
2dbl	-49.63	-56.146	-63.147	-56.871	-44.682	-39.356	-40.121	-43.709	-13.862

PDB ID	Experimental	5 Neurons	10 Neurons	15 Neurons	20 Neurons	25 Neurons	30 Neurons	35 Neurons	40 Neurons
2er0	-36.51	-33.101	-39.815	-44.299	-33.254	-40.151	-40.849	-46.892	-64.659
2er6	-41.22	-62.143	-74.033	-55.859	-33.028	-40.909	-44.261	-46.64	-60.131
2er9	-44.56	-27.977	-33.723	-9.7054	-36.042	-27.752	-31.845	-41.423	-37.112
2gbp	-15.8	-46.337	-24.406	-35.626	-29.609	-25.116	-33.605	-25.201	-32.951
2h4n	-49.65	-14.345	-36.198	-20.907	-22.356	-38.004	-34.952	-23.211	-36.405
2ifb	-30.98	-63.372	-57.315	-54.558	-33.001	-62.051	-44.009	-42.812	-35.665
2mcp	-29.85	-13.037	-11.437	-0.58315	-33.186	-7.031	-3.04	-12.045	-11.775
2ro4	-35.51	-75.365	-79.302	-52.048	-29.884	-26.003	-27.889	-41.986	-77.299
2tmn	-33.6	-39.881	-48.511	-41.063	-38.56	-33.028	-35.743	-35.85	-33.461
3cla	-28.18	-44.055	-38.488	-46.705	-32.631	-50.745	-36.178	-21.742	-28.834
3cpa	-22.13	-61.007	-40.665	-30.303	-52.937	-34.563	-37.529	-47.157	-53.267
3er3	-40.48	-44.958	-43.232	-45.093	-33.424	-26.635	-37.632	-32.661	-36.013
3ptp	-27.06	1.8376	-14.585	-9.2145	-9.5086	-12.247	-3.6238	-6.3324	-9.7393
3tmn	-33.72	-43.573	-56.131	-46.407	-45.246	-38.839	-51.9	-42.933	-46.849
3ts1	-25.07	-21.227	-34.169	-62.132	-11.49	-15.581	-23.586	-49.351	-14.907
4cpa	-47.38	-39.529	-28.711	-30.878	-21.689	-19.71	-27.269	-21.27	-22.686
4er1	-37.83	-24.784	-37.254	-46.37	-42.475	-4.8212	-52.121	-38.214	-38.905
4er4	-38.79	-51.443	-38.968	-64.444	-49.621	-67.113	-43.74	-49.482	-44.525
4sga	-18.66	-13.087	-17.588	-21.37	-20.529	-28.88	-21.877	-22.835	-24.277
4tln	-21.23	-28.489	-38.217	-34.853	-34.629	-33.653	-30.52	-37.712	-31.62
4tmn	-58.16	-45.081	-52.597	-49.185	-53.055	-53.66	-47.053	-41.024	-69.03
5er2	-37.49	-37.731	-32.209	-43.263	-39.807	-42.499	-36.993	-46.759	-60.314
5p21	-30.35	-21.996	-20.887	-42.223	-19.447	-8.4623	-4.5498	-22.709	-22.158
5sga	-16.26	-22.536	-42.671	-26.106	-24.508	-28.779	-32.089	-28.031	-23.471
5tmn	-45.89	-30.867	-29.614	-45.203	-30.802	-33.297	-34.817	-45.716	-37.734
6cpa	-65.77	-35.408	-42.959	-28.637	-22.756	-16.921	-33.548	-89.001	-24.549
6tim	-18.31	-28.63	-34.17	-28.859	-31.559	-30.448	-35.13	-30.13	-39.414
6tmn	-28.83	-34.965	-55.085	-45.397	-50.15	-41.089	-43.686	-47.804	-43.698
7hvp	-54.95	-67.792	-37.997	-46.824	-25.108	-12.807	-11.913	-41.576	-9.2373
	RMSE	20.801622	20.679473	18.702127	16.989507	29.631234	20.52988	17.4617	19.004412
	RRMSE	147.22643	146.3619	132.36696	120.24564	209.71926	145.30314	123.58765	134.50642
	MAE	15.743788	16.345682	14.093876	12.927631	16.517341	14.979286	13.426666	14.625087
	RMAE	136.29805	141.5088	122.01434	111.91785	142.9949	129.67956	116.23813	126.61317

PDB ID	45 Neurons	50 Neurons	55 Neurons	60 Neurons	BLEEP	MLR
1add	-34.748	-15.145	-20.301	-18.064	-27.28	5.360651
1adf	-45.629	-30.398	-45.676	-61.618	-28.95	-10.83647
1am6	-24.435	-45.598	-43.317	-22.809	-3.79	-22.0016
1anf	-27.151	-50.354	-27.754	-74.364	-28.03	-24.39436
1b5g	-93.079	-50.799	-51.301	-34.381	-32.41	-94.82947
1bcd	-38.106	-33.152	-33.612	-41.651	-12.9	-43.03428
1bll	29.029	-55.394	41.055	205.95	-25.09	3.9098668
1bn1	-61.6	-57.393	-61.317	-56.019	-25.38	-47.82966
1bn3	-41.557	-44.823	-22.87	-32.343	-27.31	-28.91048
1bnn	-49.148	-45.707	-50.104	-54.247	-28.63	-51.8972
1bnq	-59.618	-51.375	-44.358	-48.95	-29.96	-61.80163
1bnt	-58.953	-59.447	-64.73	-77.775	-32.45	-50.71039
1bnu	-62.461	-51.006	-64.557	-60.465	-31.29	-63.59276

PDB ID	45 Neurons	50 Neurons	55 Neurons	60 Neurons	BLEEP	MLR
1bnv	-53.426	-59.691	-62.601	-56.314	-32.96	-52.45682
1bnw	-61.694	-81.194	-52.908	-17.413	-23.28	-60.88594
1bra	-36.123	-26.912	-31.441	-24.754	-1.34	-17.20958
1byg	-24.435	-30.699	-18.824	-29.74	-54.29	-10.11504
1bzm	-48.725	-47.272	-52.754	-17.926	-19.44	-38.05532
1c83	-45.041	-44.169	-53.671	-38.394	-19.23	-61.03022
1cbs	-63.14	-34.703	-69.39	-72.917	-41.27	-47.69156
1cbx	-36.47	-61.625	-13.059	-19.873	-28.78	-45.85763
1cf8	-27.202	-3.3414	-39.865	-23.26	-35.09	-48.71704
1cil	-67.228	-55.401	-53.172	-26.561	-26.92	-56.32598
1cps	-27.219	-45.273	-34.025	-30.916	-20.77	-41.47457
1ctr	-63.11	-32.655	-17.071	-11.014	-24.43	-30.18622
1ctt	-17.299	-34.104	-29.445	-41.994	-22.87	-32.45087
1dbb	-41.098	-42.334	-26.374	-38.429	-27.88	-41.49349
1dbj	-43.387	-50.722	-52.231	-37.038	-33.9	-53.56427
1dbk	-31.844	-41.838	-32.97	-45.866	-33.35	-57.23415
1dbm	-47.938	-49.621	-45.839	-47.204	-26.53	-37.35019
1dwb	-22.761	-14.98	-13.982	-16.13	-6.7	-48.55336
1dwc	-22.938	5.853	7.9677	-8.5942	-35.84	-35.85784
1dwd	-39.265	-88.072	-46.996	-76.028	-31.24	-13.83432
1e96	-20.999	-30.203	-15.413	6.8451	-53.58	-33.68243
1eap	-23.767	-21.42	-22.975	-37.923	-32.49	-24.64858
1eed	-43.155	-40.762	-37.152	-20.32	-55.53	-21.98477
1epo	-33.632	-30.361	-29.903	-24.728	-50.73	-115.5333
1etr	-46.468	-87.405	-46.795	-93.81	-39.98	-41.00977
1fkf	-51.479	17.49	30.326	-24.748	-42.34	-3.072732
1fkg	-37.603	-31.941	-25.128	-53.576	-36.84	-33.17162
1flr	-56.262	-37.607	-57.754	-49.398	-26.77	-47.75799
1hbv	-75.193	-70.065	-49.063	-73.094	-55.08	-17.17352
1hew	-122.98	-36.483	-35.266	-58.919	-37.8	-56.48376
1hfc	-55.152	-30.288	-51.374	-50.656	-34.77	-59.23829
1hiv	-45.437	-36.097	-44.377	-57.33	-79.93	-47.82382
1hpv	-26.969	-48.853	-49.691	-21.907	-45.91	-36.52997
1hri	-52.34	-59.49	-72.405	-62.332	-31.61	-39.72048
1hvi	-60.188	-52.415	-54.822	-46.11	-79.27	-64.74279
1hvj	-42.501	-59.811	-54.329	-55.952	-75.77	103.58693
1hvk	-61.722	-56.055	-61.267	-61.208	-73.82	-72.49398
1hvl	-53.298	-55.18	-62.484	-63.086	-75.25	-32.38906
1hvr	-56.789	-54.871	-59.965	-57.649	-64.72	-102.9656
1ida	-61.33	-55.914	-68.91	-55.137	-73.82	-98.33388
1jao	-15.906	-44.076	-42.726	-29.819	-25.18	-36.26335
1kel	-20.744	-43.559	-35.904	-81.768	-30.24	9.3204569
1lgr	-5.7913	-15.229	-20.934	-22.604	-28.99	-18.14402
1mcb	-16.462	-31.197	-35.1	-39.677	-34.43	-42.12545
1mcf	-71.574	-46.882	-23.871	-115.56	-38.43	-42.78391
1mch	-68.613	-44.702	125.86	-27.472	-52.94	-55.99329
1mcj	-14.1	-29.062	-19.559	-31.528	-32.52	4.2627389
1mcs	-32.572	-61.512	-20.167	-37.445	-32.01	-21.0688
1mfe	-38.314	-31.412	-34.976	-36.482	-27.66	-1.160578
1mmb	-35.568	-38.747	-45.796	-36.39	-36.37	-26.54248
1mmq	-36.755	-33.525	-36.996	-33.375	-34.23	-37.47334
1mmr	-57.088	-50.597	-56.352	-51.641	-29.24	-60.08334
1mnc	-29.218	-25.468	-26.53	-25.046	-33.61	-34.73934

PDB ID	45 Neurons	50 Neurons	55 Neurons	60 Neurons	BLEEP	MLR
1mrk	-7.92	-12.032	-1.4857	-9.8644	-24.16	-27.30874
1mtw	-28.202	0.98939	2.5722	-5.9335	-28.04	-40.49146
1nnb	-25.7	-51.825	-37.541	-36.104	-23.81	-27.16819
1okl	-29.605	-22.806	-42.768	-37.105	-26.75	-42.77077
1ola	-37.444	-50.501	-40.692	-50.584	-40.2	-102.4568
1phf	-44.161	-43.8	-52.241	-62.266	-86.56	-41.13791
1phg	-27.6	-47.319	-28.744	-36.517	-96.61	-36.73098
1ppc	-42.61	-42.053	-16.772	-65.65	-32.05	-10.57321
1qbr	-51.263	-52.177	-55.522	-61.395	-65.76	-36.82952
1qbt	-73.63	-59.829	-63.617	-58.203	-73.35	-73.14919
1qbu	-50.658	-55.902	-59.185	-47.27	-57.65	-59.82452
1rbp	-28.263	-31.632	-49.368	18.605	-41.29	-16.01811
1rgk	-29.245	-54.072	-37.403	-31.072	-22.63	-30.14643
1rgl	-33.571	-37.689	-24.134	-40.147	-19.75	-37.61635
1sln	-60.397	-43.069	-52.362	-20.326	-43.23	-59.97601
1stp	-36.512	-39.964	-22.336	-22.475	-25.27	-8.750975
1tet	-39.606	-33.586	-16.817	-21.65	-53.93	-11.00445
1thl	-72.486	-76.98	-71.56	-68.464	-34.93	-64.70907
1tlp	-65.221	-18.794	-47.429	-39.767	-27.35	-73.55076
1tmn	-25.628	-22.348	-27.166	-29.576	-37.52	-30.01495
1tng	-21.478	-25.627	-15.407	-23.14	-14.66	-15.34461
1tnh	-12.082	-10.493	-15.823	-11.227	-15.86	-13.30505
1tni	-7.2758	-11.045	-6.5698	-7.8349	-2.86	-19.10679
1tnj	-10.933	-11.784	-9.9523	-12.533	-5.49	-10.88752
1tnk	-15.883	-16.548	-18.296	-18.963	-8.42	-10.89278
1tnl	-37.865	-12.033	-12.48	-21.187	-6.33	4.3317124
1uvs	-44.867	18.426	-49.57	-59.502	-32.63	-53.4947
1uvt	-18.283	30.922	-36.09	-75.049	-23.89	-30.30484
1zzz	-34.749	-27.877	-39.572	-45.05	-17.04	-29.52086
2abh	-41.393	-40.552	-34.413	-30.148	-19.22	-33.53702
2cgr	-37.224	-24.513	-33.928	-34.124	-24.22	-37.16395
2cmd	-29.424	-24.688	-34.005	-48.895	-20.91	-57.38261
2dbl	-47.07	-39.392	-45.884	-24.5	-29.87	-41.49038
2er0	-25.645	-40.809	-42.076	-33.26	-76.29	-24.57333
2er6	-73.541	-53.974	-78.177	-39.124	-74.93	-39.84075
2er9	-39.243	-42.884	-46.604	-39.106	-66.02	-81.18833
2gbp	-45.617	-23.778	-36.091	-39.328	-29.39	-35.45512
2h4n	-19.663	-29.891	-27.405	-25.126	-14.05	-22.51004
2ifb	98.95	-58.938	-11.245	-50.997	-33.74	-69.25606
2mcp	-5.0106	1.9166	-10.91	-7.8123	-7.61	7.4217113
2ro4	-52.368	-62.357	-44.518	-77.704	-42.68	-16.47788
2tmn	-37	-34.17	-32.76	-34.994	-18.01	-40.47972
3cla	-35.879	-30.912	-16.394	-45.074	-32.69	-45.35104
3cpa	-40.221	-43.325	-12.262	-45.084	-27.46	-71.77603
3er3	-41.245	-34.884	-24.366	-31.656	-68	-57.94815
3ptp	-14.726	-12.212	-14.392	-7.0545	-4.48	-8.22444
3tmn	-37.78	-44.938	-44.657	-48.236	-25.29	-10.68501
3ts1	-40.839	-33.295	-93.424	-31.784	-30.95	13.821158
4cpa	-28.161	-34.854	-21.683	-25.289	-64.88	-25.84186
4er1	-51.086	-27.168	-44.85	-40.2	-76.3	-17.52369
4er4	-38.444	-27.435	-18.294	-48.263	-77.26	68.90024
4sga	-21.456	-25.623	-20.712	-36.547	-29.51	-18.92978
4tln	-29.221	-27.74	-31.992	-32.537	-10.25	-34.69118

PDB ID	45 Neurons	50 Neurons	55 Neurons	60 Neurons	BLEEP	MLR
4tmn	-48.607	-56.328	-54.584	-52.303	-42.19	-39.93665
5er2	-43.805	-38.047	-39.149	-37.671	-74.62	-2.303404
5p21	-23.366	-30.488	-42.033	-28.81	-60.96	-37.41412
5sga	8.4634	-26.735	-19.922	-22.237	-29.93	-35.52933
5tmn	-28.933	-48.416	-37.493	-41.054	-35.99	-46.39222
6cpa	-21.111	27.811	-20.055	-248.09	-41.31	-4.575591
6tim	-28.945	-29.736	-36.095	-29.482	-10.93	-50.52273
6tmn	-24.463	-43.213	-40.455	-33.829	-34.59	-30.49769
7hvp	-40.431	-28.284	4.5091	-22.467	-73.78	17.26058
RMSE	22.909963	21.227373	25.454161	34.43159	17.938208	30.208732
RRMSE	162.14851	150.23974	180.15543	243.69446	126.96021	213.80023
MAE	15.602607	14.485637	16.113694	19.723174	13.725156	20.747528
RMAE	135.07581	125.40592	139.50042	170.74863	118.82223	179.61673

Appendices C & D

Due to the extremely large sizes of Appendices C & D respectively, they have been attached in the accompanying CD as a MS Word(C) and Excel File(D). Thank you for your understanding.

Appendix E.

Character Codes(1 and 3) For Amino Acid Residues

G

Glycine
Gly

P

Proline
Pro

A

Alanine
Ala

V

Valine
Val

L

Leucine
Leu

I

Isoleucine
Ile

M

Methionine
Met

C

Cysteine
Cys

F

Phenylalanine
Phe

Y

Tyrosine
Tyr

W

Tryptophan
Trp

H

Histidine
His

K

Lysine
Lys

R

Arginine
Arg

Q

Glutamine
Gln

N

Asparagine
Asn

E

Glutamic Acid
Glu

D

Aspartic Acid
Asp

S

Serine
Ser

T

Threonine
Thr