The Pennsylvania State University
The Graduate School
Department of Chemistry

METHODS TO IMPROVE THE RELIABILITY, VALIDITY AND INTERPRETABILITY OF QSAR MODELS

A Thesis in
Chemistry
by
Rajarshi Guha

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

August 2005

The thesis of Rajarshi Guha has been reviewed and approved* by the following:

Peter C. Jurs
Professor of Chemistry
Thesis Adviser
Chair of Committee

Kenneth M. Merz, Jr.
Professor of Chemistry

Juliette T.J. Lecomte
Associate Professor of Chemistry

Costas Maranas
Professor of Chemical Engineering

Ayusman Sen
Professor of Chemistry
Head of the Department of Chemistry

*Signatures are on file in the Graduate School

Abstract

Quantitative structure activity relationship (QSAR) models are are a statistical solution to the problem of directly calculating physical and biological properties of molecules from their physical structure. The direct prediction of properties is in general not feasible either owing to lack of computing resources or lack of knowledge about the relationship between structure and property. The goal of a QSAR model is to extract information from a set of numerical descriptors characterizing molecular structure and use this information to develop inductively a relationship between structure and property. Two important questions arise during the modeling process. First, are the data used to build the model representative of the whole dataset and can the model be extended to predict properties for new molecules? Second, given that a model encodes information about the structures of molecules and relates this to their properties, can we extract and interpret the encoded information? The focus of the work reported in this thesis is on the validation and interpretation of QSAR models and presents both applications of interpretation techniques as well as the development of validation and interpretation methodologies.

The first study describes a technique to develop representative QSAR sets using a self-organizing map (SOM). The SOM was used to classify a dataset consisting of dihydrofolate reductase inhibitors with the help of an external set of global descriptors. The resultant classification was used to generate training, cross-validation and prediction sets (collectively known as QSAR sets) for QSAR modeling using the ADAPT methodology. The results were compared to those of QSAR models generated using sets created by activity binning and a sphere exclusion method. The results indicated that the SOM was able to generate QSAR sets that were representative of the composition of the overall dataset in terms of similarity. The resulting QSAR models were half the size of those published and had comparable RMS errors. Furthermore, the RMS errors of the QSAR sets were consistent, indicating good predictive capabilities as well as generalizability.

The determination of the validity of a QSAR model when applied to new compounds is an important concern in the field of QSAR modeling. Various scoring techniques can be applied to specific types of models to obtain measures of confidence in the predicted property for new compounds. The second study describes the development of a methodology which allows one to state whether a new compound will be well predicted

by a previously built QSAR model. The study focuses on linear regression models only, though the technique is general and can also be applied to other types of quantitative models. The technique is based on a classification method that divides regression residuals from a previously generated model into a good class and bad class and then builds a classifier based on this division. The trained classifier is then used to determine the class of the residual for a new compound. The performance of a variety of classifiers, both linear and nonlinear, was investigated. The technique was tested on two data sets from the literature and an artificial data set. The data sets selected covered both physical and biological properties and also presented the methodology with quantitative regression models of varying quality. The results indicate that this technique can determine whether a new compound will be well or poorly predicted with weighted success rates ranging from 73% to 94% for the best classifier.

The remaining studies focus on methods to interpret QSAR models. The third and fourth studies describe applications of a partial least squares (PLS) based method to interpret linear regression models. The third study developed QSAR models to predict the biological activity of 179 artemisinin analogues. The structures of the molecules were represented by numerical descriptors. Both linear (multiple linear regression) and nonlinear (computational neural network) models were developed to link the structures to their reported biological activity. The best linear model was subjected to a PLS analysis to provide model interpretability. While the best linear model did not perform as well as the nonlinear model in terms of predictive ability, the application of PLS analysis allows for a sound physical interpretation of the structure-activity trend captured by the model. On the other hand, the best nonlinear model was superior in terms of pure predictive ability, as characterized by low training and prediction set root mean square errors.

The fourth study consisted of the development and interpretation of QSAR models to predict the activity of a set of 79 piperazyinylquinazoline analogues which exhibited platelet derived growth factor (PDGFR) inhibition. Linear regression and nonlinear computational neural network models were developed. The linear regression model was developed with a focus on interpretative ability using the PLS technique. However, it also exhibited good predictive ability after outlier removal. The nonlinear CNN model had superior predictive ability compared to the linear model, having a prediction set root mean square errors nearly half that of the linear model. A random forest model was also developed to provide an alternate measure of descriptor importance. This approach ranks descriptors, and its results confirmed the importance of specific descriptors as characterized by the PLS technique.

Studies five and six describe the development of methods to provide interpretability to computational neural network (CNN) models. The fifth study focuses on a measure of relative importance of the descriptors present in a CNN model. The approach is based on a sensitivity analysis of the descriptors and is similar in concept to the descriptor importance measure for random forest models. The method was tested on three published data sets for which linear and CNN models were previously built. The original work reported interpretations for the linear models. This study compared the results of this method to the importance of descriptors in the linear models as described by the PLS technique. The results indicate that the proposed method is able to rank descriptors such that important descriptors in the CNN model correspond to the important descriptors in the linear model.

The sixth study presents the development of a method to provide a detailed interpretation of a CNN model. This methodology provides a means to analyze the correlation between specific input descriptor and the predicted output of the network, rather than simply providing a ranking of all descriptors. The method consists of two parts. First, the nonlinear transform for a given neuron is linearized, allowing us to determine how a given neuron affects the downstream output. Next, a ranking scheme for neurons in the hidden layer is developed. This scheme allows for the development of interpretations of a CNN model similar in manner to the PLS interpretation method for linear models. The method was tested on three datasets covering both physical and biological properties. The results of this interpretation method correspond well to PLS interpretations for linear models using the same descriptors as the CNN models.

Table of Contents

List of Tables

List of Figures

Acknowledgments

Working on a PhD and writing a thesis is certainly a formidable endeavour. However, the last four years have been very enjoyable and a number of people have helped me reach this point.

I would like to thank my advisor, Dr. Jurs, for giving me the freedom to pursue interesting avenues of inquiry but also keeping me on track. The freedom to follow up on different ideas has made the last four years more fun than work. His advice has during these four years has helped me out of seemingly dead ends and I have certainly learnt a lot about the process of research from our conversations.

I'd also like to thank members of the group, especially Jon Serra and Brian Mattioni. Jon really helped me learn the ropes of a lot of things, ranging from running ADAPT to buying a car. We've had a lot enjoyable times both in and out of the lab and has been a great companion while working on code as well as when watching movie. I'd also like to thank Brian and Jon for being great sounding boards. It was, and still is, always enjoyable to bounce ideas of each other.

David Stanton has also played an important role in these last four years. The clarity with which Dave approached problems has been an inspiration. I am also grateful to Dave for pointing out, in a very nice way, that statistics is fundamental to QSAR modeling. In absence of his guidance, I would probably not have learnt as much as I did. It has been a pleasure working with Dave and I look forward to fruitful collaborations in the future.

I would like to express my gratitude to my parents who were generous enough to accept that I would be far away from them for a long time. In addition, they have always encouraged me to reach higher and this work is a result of their encouragement.

Finally I would like to thank Debarchana for the support she has given me over the last few years. It has been tough to stay apart from each other for so long, but with this thesis completed, our time apart is drawing to a close.

Chapter 1

Introduction

1.1 To Calculate or Predict?

Until recently advances in medicinal and pharmaceutical chemistry depended on a trial and error process aided by intuition. Though the properties that would indicate a certain molecule as a drug candidate were known, it was not really feasible to investigate large numbers of molecules for these types of properties. Of course, the nature of these properties would be represented by structural features of a molecule and thus examination of certain motifs provided a direction for experimental investigations.

The problem with this approach is that it does not always lead to an understanding of why a molecule behaves as a drug against its target or why it does so. Furthermore, given a series of compounds it is not always feasible to investigate experimentally which members of the series would be more potent or less toxic. As a result, though medicinal chemistry has resulted in a series of life saving drugs, the process has traditionally been slow and tedious, and in many cases advances have been due to serendipity rather than scientifically guided investigation.

In an ideal world one would be able to take a 3-D molecular structure and calculate the required properties. This utopian goal has a number of problems associated with it. First, what type of properties are to be calculated? Certain intrinsic physical properties can be calculated using *ab initio* quantum mechanical computation techniques. Examples include dipole moments, charges and heats of formation. Though these are certainly useful, they do not provide much insight into drug-like properties such as potency and bioavailability. In addition, for large collections of molecules, *ab initio* techniques become very time consuming. Semi-empirical quantum mechanical methods alleviate the intensive nature of these calculations, but we are still faced with the restriction on the types of properties that can be calculated. Second, the drug-like activity of a molecule is intimately related to the target it is supposed to interact with. Targets generally involve some type of protein to which the putative drug will bind. Thus when considering the activity of a drug, we cannot simply consider the properties of the drug molecule itself. That is, the nature of the interaction between the drug and target must be investigated to

understand fully the activity of a drug. However, *ab initio* and semi-empirical techniques have traditionally not been suited for the modeling of large protein systems. Though recent advances in linear scaling[1,2] and hybrid techniques[3,4] have expanded the purview of quantum mechanical methods to systems containing tens of thousands of molecules, these methods are still not efficient enough to model thousands or millions of molecular structures, and their associated targets, at a time. Third, though the interactions of a drug with its target are certainly important, the drug must be absorbed by cells and the also metabolized and excreted from the body. Thus absorption properties, the nature of the metabolites and other characteristics must also be considered. Clearly, these are very complex properties that involve interactions with a large number of cellular processes. Modeling these quantum mechanically is nearly impossible.

The above discussion illustrates two fundamental problems. It is not feasible to calculate from theory all the properties of a drug molecule that would help us understand its activity and its utility, and we want to be able to analyze large sets of molecules for these properties.

Why do we need to analyze large sets of molecules? The reason for this is closely tied to the nature of drug discovery in recent years. The drug discovery process is time consuming and expensive. Often it can take 10 to 15 years for a drug to reach the market from the laboratory. Given this situation, it is important that a company select the proper compound for study. Combined with the results from high throughput screens[5] and in-house libraries, this can mean having to select tens or hundreds of compounds from a collection of millions. Furthermore, the ability to generate an arbitrary number of unique chemical structures *in silico*, to create virtual libraries, supplants the actual compounds that a company might have synthesized in its physical collection. Clearly, testing each compound libraries (virtual or real) for drug-like properties is out of the question. As we have seen above, calculating properties for collections of this size is either not feasible or impossible. The question thus comes down to this: how can we calculate arbitrary properties of hundreds of thousands of molecules rapidly and accurately? The short answer is that we avoid the calculation step completely and instead *predict* a property of a set molecules based on a model derived from the measured values of that property for a small subset.

1.2 Origins of QSAR

The predictive approach is essentially a statistical methodology and is known as the development of quantitative structure-activity relationship (QSAR)and quantitative structure-property relationship (QSPR) models, first described by Hansch[6,7] and Free et al.[8] In general, the term QSPR refers to the case where we are considering physical properties and QSAR refers to the situation where we are considering biological activities. However, in this work the term QSAR is used to include both cases.

Though Hansch was the first worker to define the term QSAR, A.F.A Cros, in 1863, had noted that the toxicity of alcohols in mammals increased with the decrease in water solubility.[9] Workers in the 1890's noted that toxicity of organic compounds depended on their lipophilicity. The precursor to QSAR models were linear free energy relationships such as the Hammett equation,[10] which was originally defined as a relationship between the electronic properties of acids (and bases) and their disassociation constants and reactivity. The equation is defined as

$$\log \frac{K}{K_0} = \rho \log \frac{K'}{K'_0} \tag{1.1}$$

where K and K' represent the dissociation constants for a set of substituted aromatic acids and K_0 and K'_0 are the constants for the unsubstituted acids. ρ is the slope of the best fit line from the model fitted to the observed constants. The term $\log(K'/K'_0)$ is denoted by σ and describes the substituents.

Hansch originally tried to develop QSAR models using the Hammett σ parameter but this did not lead to good results. He thus considered other parameters such as the lipophilicity and molecular size as represented by molar refractivity.

The essence of the QSAR methodology is thus developing a relationship between an observed property and structural features of a molecule. By considering a set of molecules, a predictive model is developed that can then be used to predict the activity of other molecules. The key words here are "structural features". The approach depends on being able to represent the structure of a molecule in numerical form. This is in contrast to the use of empirical parameters (σ) in the case of linear free energy relationships. The numerical representations of molecules are termed *descriptors*, and a wide variety of descriptors can be calculated. These include simple forms such as molecular weight and atom counts or more complex types such as partition coefficients and surface-property descriptors. Given a set of descriptors, a QSAR model can be built

by defining a relationship between these descriptors (also known as the independent variables) and the observed property (termed the dependent variable). The first QSAR models, developed by Hansch, specified linear relationships. Even now, linear models are widely used owing to their simplicity and ease of development. However, developments in the field of statistics have produced many new methods of building predictive models. These include nonlinear regression techniques and algorithmic techniques.[11] Other fields such as pattern recognition and machine learning have also developed methods that have been used successfully in QSAR modeling. These include neural networks and subsequent variants,[12,13] decision trees[14,15] and so on. Clearly, progress in the field of QSAR modeling is closely tied to developments in a number of fields including statistics, computer science and mathematics.

The process of QSAR modeling is summarized in Fig. 1.1. The diagram stresses the fact that a QSAR model is an alternate stepwise route to the calculation of molecular properties. That is, the direct calculation of molecular properties is generally not feasible. In addition, if we do not understand the nature of interactions that a molecule undergoes in expressing its activity, accurate calculation of its properties is impossible. Thus, we proceed by an indirect route, which we term the QSAR pipeline, in which we represent a molecule in a computer understandable format, distill molecular features by calculating molecular descriptors and then build predictive models. The important feature of the QSAR modeling process is that it *predicts* molecular properties rather than *calculating* them. This fact raises a number of issues such as the validity of predictions. Another important aspect is the nature of the information that is input to the model. That is, what types of descriptors should the model use, given that we can calculate thousands of them. This is the problem of feature selection. Finally, the model predicts molecular properties based on information present in the dataset that it has encoded. It is thus important to be able to extract the encoded information from the model, and this is the topic of interpretability. These issues are discussed later in this thesis.

Though the above discussion has focused on drug molecules, the QSAR methodology is certainly not restricted to these types of molecules. In fact a QSAR model can be built to predict any type of physical property of biological activity, given a set of observations and molecular structures. Examples of the prediction of physical properties include boiling points,[16–18] aqueous solubility,[19,20] glass transition temperatures[21] and ion mobility.[22] In the area of biological activities QSAR models have been developed to predict genotoxicity,[23–25] carcinogenicity[26,27] and mutagenicity.[28,29] Furthermore, the use of QSAR models is not restricted to their role in screening large libraries of compounds.

In some cases a series of compounds may be synthesized and assayed. The development of a QSAR model for these compounds would provide the synthetic chemist some idea of what types of compounds could be synthesized to exhibit better activity. In other cases, the structural features highlighted by a QSAR model can provide insight into the mode of action of a drug molecule, which might be otherwise difficult to ascertain by experimental means.

1.3 QSAR Methodologies

QSAR methodologies can be broadly divided into three groups. First, 2-D methodologies do not consider the 3-D structure of a molecule directly. Instead, the molecule is represented by a set of molecular descriptors, numerical values characterizing various aspects of molecular structure. Together with the observed activity, a predictive model is built. It should be noted, that even though some descriptors are based on 3-D coordinates, the method as a whole considers only the observed property and the descriptors, and hence is 2-D in nature. The ADAPT software suite implements the 2-D QSAR methodology.

The second type of methodology is 3-D in nature and is exemplified by the CoMFA[30] approach. In this case, the 3-D structure of the molecule is the object of study. The molecule is aligned on a grid and various properties are evaluated at a set of grid points. Clearly, this type of approach has many advantages over the more simplistic 2-D methodology. The fact that the molecule is studied directly in three dimensions, rather than being mapped to two, allows for a clearer view of the interactions between the molecule and its target that play a role in the observed activity. However it does require accurate alignments and only considers a single conformation of a molecule.

The 4-D QSAR methodology is an extension of the 3-D QSAR methodology developed by Hopfinger et al.[31] which considers conformational information as the fourth dimension. Similar to the CoMFA method, 4-D QSAR starts of by defining a set of grid points on which molecular properties will be evaluated. In addition to the grid points, the method performs conformational ensemble sampling and uses the information obtained to evaluate grid cell occupancies. These occupancies are then used to evaluate *interaction pharmacophore elements* (IPE's). The IPE's together with the molecular properties are then used to develop a predictive model.

This work focuses on the 2-D QSAR methodology and presents investigations carried out on certain steps of the model building process. Compared to the 3-D and 4-D methodologies described above the 2-D approach has a number of advantages. First, owing to the variety of molecular descriptors available, optimized coordinates are not always required. In fact, connectivity information (in the form of SMILES strings or an adjacency matrix) alone, can be used to develop QSAR models. As a result models using these types of descriptors (termed topological descriptors) can be built rapidly for very large sets of molecules. However, these types of descriptors are in general quite abstract and so if the model is to be analyzed to extract information regarding structure-property trends, other, more physically meaningful descriptors will generally be required. Second, this approach avoids the alignment step and thus can be used in the absence of experimental information regarding the binding of a molecule to its target.

The downside to the 2-D QSAR methodology is that it does not provide a detailed answer to a number of questions regarding a molecule's activity. That is, by representing structural information in the form of descriptors, aspects of a molecules activity such as its absorption properties or degradability are hidden by a layer of abstraction or not addressed at all. Thus a molecule might be observed to have low activity. A 2-D model may not be able to indicate whether this is due to its inability to bind to the target or whether this is due to its inability to cross the cell membrane. The point is that, in a 2-D QSAR model, a lot of information about various aspects of a molecule's activity are combined together and are not always individually apparent. Though interpretation methods for linear QSAR models exist, they are obviously restricted to the information encoded by the descriptors in the model. This means that though 2-D QSAR models are certainly very useful, especially for screening purposes, they should be used in conjunction with other types of models to fully understand the role that various structural features play in determining the activity of a molecule.

2-D QSAR models can also be divided into two distinct groups, namely, qualitative and quantitative models. The former type of model, also known as classificatory models, consider a categorical dependent variable. That is, the observed property for each observation is represented by a label, such as toxic or non-toxic. Thus, if a dataset is available for which an assay has been carried out indicating whether a given molecule is carcinogenic or not, a 2-D qualitative model can be built that will predict whether a molecule , not belonging to the set, is carcinogenic or not. These types of models are not restricted to yes/no problems and datasets with multiple classes (say, active, moderately active and inactive) can be modeled. The second type of 2-D QSAR models

are referred to as quantitative (or regression) models. The function of these types of models is to predict a numerical value for a property, for example, boiling points or IC_{50} values. At the same time it should be pointed out that even when the observed property for a dataset is numeric in nature, it can be studied using qualitative models. This is generally achieved by selecting a break point in the range of the observed values and placing molecules whose property is above the break point in one class and the remaining molecules in another class. With these class assignments, a classificatory model can then be built. This thesis focuses on the development of regression models.

An important part of QSAR modeling is the use of software to create structures, calculate descriptors and build predictive models. A number of commercial packages provide QSAR modeling facilities, and examples include Cerius2[32] from Accelrys and Strike[33] from Schrodinger. These packages provide a comprehensive environment that is linked to chemical databases and a variety of cheminformatics functionality and as a result, encompass the whole process of model building and data analysis. Some examples of freely available programs include PowerMV[34] and the ADAPT system described in this thesis. Other programs tend to focus on specific aspects of the QSAR model building process. For example, a number of programs are available to calculate descriptors. Examples include Dragon,[35] JOELib[36] and Codessa.[37] Some programs focus on calculating a set of properties that can indicate the drug likeness of a molecule, such as metabolite types, bioavailability and so on. An example of such a program is QikProp developed by Jorgensen et al.[38–40] It is obvious that a fundamental component of QSAR modeling is the statistical analysis of chemical information. Thus, a number of statistical packages can be used to perform QSAR modeling such as SAS, Splus and R.[41] One problem with these environments is that they are geared towards statistics. As a result, having access to chemical functionality from within these statistical environments is attractive. An example of this type of environment is the combination of R and the Chemistry Development Kit (CDK)[42] described by Guha[43] allowing the user to have access to the full statistical capabilities of R as well as the cheminformatics capabilities of the CDK.

1.4 An Outline

This section briefly outlines the various topics considered in this thesis. Chapter 2 introduces the modeling techniques that are used in this work. Though a detailed presentation of the various algorithms and models that are used in QSAR modeling would take up a whole book, the chapter describes the broad classes of models and algorithms

employed in this work and focuses in the theoretical principles of some specific methods. Chapter 3 then gives a detailed description of the general QSAR methodology that is employed in the various studies presented in this work. Subsequent chapters represent investigations and applications that have been carried out on specific steps of the QSAR model building process.

Chapter 4 focuses on the set selection step. This step in the QSAR pipeline divides the original dataset into subsets which are collectively known as QSAR sets. These subsets are then used to build and test the QSAR model. A set selection procedure is developed to create representative sets for the purpose of building and testing QSAR models using a self-organized map.[44] The assumption underlying this method is that if the features of the dataset are proportionately represented in the subsets used to build and test a QSAR model, the resultant model should exhibit better predictive ability and should be more reliable, than models built with sets selected by random selection which does not necessarily represent different features proportionately.

Chapter 5 then focuses on the validation step of the QSAR pipeline and describes a technique that was developed to be able to ascertain the reliability of a QSAR when asked to predict properties of compounds that it has never seen. The validation of QSAR models, over and above the traditional methods, using scrambling tests and an external prediction set, is an important topic. The ability to obtain a measure of confidence in the predictions of a QSAR model is very important when such models are used to process incoming data from high throughput screens or when used by a bench chemist to decide whether to invest time and effort on the characterization of a new lead. Some model types do allow confidence measures to be calculated, but these are generally specific to the model type. The method described in Chapter 5 presents a much more general approach to this problem, applicable to any type of quantitative model.

The next four chapters focus on the topic of interpretability. Chapters 6 and 7 describe the development and interpretation of linear regression QSAR models. Chapter 6 presents a study of a set of artemisinin analogs that were designed for their anti-malarial activity. Both linear and nonlinear models are developed and the former is subsequently interpreted using the PLS technique. Chapter 7 describes a study of a set of PDGFR inhibitors, which are of interest owing to their ability to interfere with cell signal transduction mechanisms and are therefore of interest as anti-cancer drugs. As before, the study develops linear and nonlinear models and presents an interpretation of the linear model. In addition,a random forest model is developed to investigate the importance of the descriptors used in the study and in specific models.

The focus of interpretation techniques in the field of 2-D QSAR modeling has generally been restricted to the interpretation of linear regression models. In some cases, neural network models have been interpreted in a broad manner. Chapters 8 and 9 describe methods that were developed to interpret neural network models. Chapter 8 describes a simple method to provide a quantitative measure of descriptor importance in a neural network. The method is based on a sensitivity analysis of the model and is similar in nature to the descriptor importance measure that is available for random forest models. However, this method is similar to other approaches to the interpretation of neural networks since it only provides information about which descriptor is the most important for the model's predictive ability. It does not provide any insight into the nature of the correlation between the input to the network and the output from the network. A method to extract detailed information regarding the structure-property relationships encoded in the weights and biases of a trained neural network models is described in Chapter 9. This method is inspired by the PLS interpretation technique for linear models. The method simplifies the neural network and considers the hidden neurons of the network in a manner analogous to the latent variables of the PLS interpretation. In addition, plots analogous to the score plots of the PLS technique are presented. Combining the visual information provided by the score plots together with the analysis of the weights and biases, the method presented is able to provide a detailed view of the correlations between the input descriptors and the predicted property. The method thus provides for neural network models, what the PLS method has provided for linear regression models. Namely, an in-depth, compound-wise dissection of the structure-property trends encoded in the respective models

Finally, Chapter 10 summarizes the results of the studies presented in this work and concludes by highlighting the contributions of this thesis to the field of QSAR modeling.

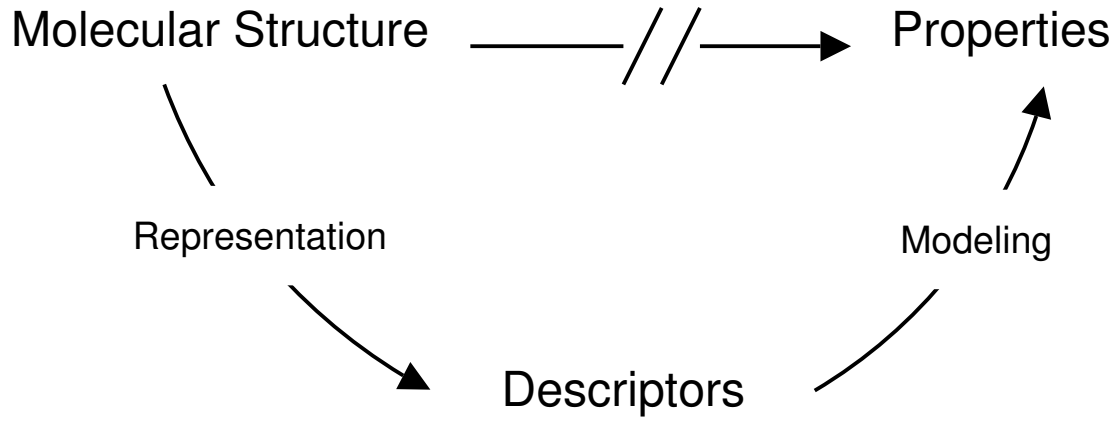


Fig. 1.1. A flowchart showing the steps involved in predicting molecular properties or activities from molecular structure

References

[1] Mei, Y.; Zhang, D. W.; Zhang, J. Z. H. New Method for Direct Linear-Scaling Calculation of Electron Density of Proteins. *J. Phys. Chem. A* **2005,** *109,* 2–5.

[2] Dixon, S.; Merz, K. Semiempirical Molecular Orbital Calculations with Linear System Size Scaling. *J. Chem. Phys.* **1996,** *104,* 6643–6649.

[3] Warshel, A. *Computer Modeling of Chemical Reactions in Enzymes and Solutions;* Wiley: New York, 1991.

[4] Clementi, E. *Computational Aspects for Large Chemical Systems;* Springer: New York, 1980.

[5] Hertzberg, R.; Pope, A. High-Throughput Screening: New Technology For the 21st Century. *Curr. Opin. Chem. Biol.* **2000,** *4,* 445–451.

[6] Hansch, C. A Quantitative Approach to Biochemical Structure-Activity Relationships. *Acc. Chem. Res.* **1969,** *2,* 232–239.

[7] Hansch, C.; Fujita, T. $\epsilon-\sigma-\pi$ Analysis. A Method for the Correlation of Biological Activity and Chemical Structure. *J. Am. Chem. Soc.* **1964,** *86,* 1616–1626.

[8] Free, S. M. J.; Wilson, J. W. A Mathematical Contribution to Structure Activity Studies. *J. Med. Chem.* **1964,** *7,* 395–399.

[9] Borman, S. New QSAR Techniques Eyed for Environmental Assessments. *Chem. Eng. News* **1990,** *68,* 20–23.

[10] Hammett, L. The Effect of Structure Upon the Reactions of Organic Compounds. Benzene Derivatives. *J. Am. Chem. Soc.* **1937,** *59,* 93–103.

[11] Breiman, L. Statiscal Modeling: Two Cultures. *Statistical Science* **2001,** *16,* 199–231.

[12] Chen, L.; Gasteiger, J. Knowledge Discovery in Reaction Databases: Landscaping Organic Reactions by a Self Organizing Neural Network. *J. Am. Chem. Soc.* **1997,** *119,* 4033–4042.

[13] Espinosa, G.; Arenas, A.; Giralt, F. An Integrated SOM Fuzzy ARTMAP Neural System for the Evaluation of Toxicity. *J. Chem. Inf. Comput. Sci.* **2002,** *42,* 343–359.

[14] Schuurmann, G.; Aptula, A. O.; Kuhne, R.; Ebert, R. Stepwise Discrimination between Four Modes of Toxic Action of Phenols in the *Tetrahymena pyriformis* Assay. *Chem. Res. Tox.* **2003,** *16,* 974–987.

[15] Butina, D.; Gola, J. M. R. Modeling Aqueous Solubility. *J. Chem. Inf. Comput. Sci.* **2003,** *43,* 837–841.

[16] Goll, E.; Jurs, P. Prediction of the Normal Boiling Points of Organic Compounds From Molecular Structures with a Computational Neural Network Model. *J. Chem. Inf. Comput. Sci.* **1999,** *39,* 974–983.

[17] Rucker, C.; Meringer, M.; Kerber, A. QSPR Using MOLGEN-QSPR: The Example of Haloalkane Boiling Points. *J. Chem. Inf. Comput. Sci.* **2004,** *44,* 2070–2076.

[18] Ehresmann, B.; de Groot, M. J.; Alex, A.; Clark, T. New Molecular Descriptors Based on Local Properties at the Molecular Surface and a Boiling-Point Model Derived from Them. *J. Chem. Inf. Comput. Sci.* **2004,** *44,* 658–668.

[19] Butina, D.; Gola, J. M. R. Modeling Aqueous Solubility. *J. Chem. Inf. Comput. Sci.* **2003,** *43,* 837–841.

[20] Yan, A.; Gasteiger, J. Prediction of Aqueous Solubility of Organic Compounds Based on 3D Structure Representation. *J. Chem. Inf. Comput. Sci.* **2003,** *43,* 429-434.

[21] Mattioni, B. E.; Jurs, P. C. Prediction of Glass Transition Temperatures from Monomer and Repeat Unit Structure Using Computational Neural Networks. *J. Chem. Inf. Comput. Sci.* **2002,** *42,* 232–240.

[22] Mosier, P.; Counterman, A.; Jurs, P.; Clemmer, D. Prediction of Peptide Ion Collision Cross Sections from Topological Molecular Structure and Amino Acid Parameters. *Anal. Chem.* **2002,** *74,* 1460–1370.

[23] Mosier, P. D.; Jurs, P. C.; Custer, L. L.; Durham, S. K.; Pearl, G. M. Predicting the Genotoxicity of Thiophene Derivatives from Molecular Structure. *Chem. Res. Toxicol.* **2003,** *16,* 721–732.

[24] McElroy, N. R.; Thompson, E. D.; Jurs, P. C. Classification of Diverse Organic Compounds That Induce Chromosomal Aberrations in Chinese Hamster Cells. *J. Chem. Inf. Comput. Sci.* **2003,** *43,* 2111–2119.

[25] Mattioni, B. E.; Kauffman, G. W.; Jurs, P. C.; Custer, L. L.; Durham, S. K.; Pearl, G. M. Predicting the Genotoxicity of Secondary and Aromatic Amines Using Data Subsetting To Generate a Model Ensemble. *J. Chem. Inf. Comput. Sci.* **2003,** *43,* 949–963.

[26] Hemmateenejad, B.; Safarpour, M. A.; Miri, R.; Nesari, N. Toward an Optimal Procedure for PC-ANN Model Building: Prediction of the Carcinogenic Activity of a Large Set of Drugs. *J. Chem. Inf. Model* **2005,** *45,* 190–199.

[27] Novak, M.; Rajagopal, S. Correlations of Nitrenium Ion Selectivities with Quantitative Mutagenicity and Carcinogenicity of the Corresponding Amines. *Chem. Res. Toxicol.* **2002,** *15,* 1495–1503.

[28] Basak, S. C.; Mills, D. R.; Balaban, A. T.; Gute, B. D. Prediction of Mutagenicity of Aromatic and Heteroaromatic Amines from Structure: A Hierarchical QSAR Approach. *J. Chem. Inf. Comput. Sci.* **2001,** *41,* 671–678.

[29] Kazius, J.; McGuire, R.; Bursi, R. Derivation and Validation of Toxicophores for Mutagenicity Prediction. *J. Med. Chem.* **2005,** *48,* 312–320.

[30] Cramer III, R.; Patterson, D.; Bunce, J. Comparative Molecular Field Analysis (CoMFA). I. Effect of Shape on Binding of Steroids to Carrier Protiens. *J. Am. Chem. Soc.* **1988,** *110,* 5959–5967.

[31] Hopfinger, A.; Wang, S.; Tokarski, J.; Baiqiang, J.; Albuquerque, M.; Madhav, P.; Duraiswami, C. Construction of 3D-QSAR Models Using the 4D-QSAR Analysis Formalism. *J .Am. Chem. Soc.* **1997,** *119,* 10509–10524.

[32] Accelrys, "Cerius2", `http://www.accelrys.com/cerius2`,.

[33] Schrödinger, "Strike", `http://www.schrodinger.com/Products/strike.html`,.

[34] Liu, K.; Feng, J.; Young, S. PowerMV: A Software Environment for Molecular Viewing, Descriptor Generation, Data Analysis and Hit Evaluation. *J. Chem. Inf. Model* **2005,** *45,* 515–522.

[35] Todeschini, R.; Consonni, V.; Pavan, M. "DRAGON", 2005.

[36] Wegner, J. "JOELib", `http://joelib.sf.net`, 2005.

[37] Semichem, Inc., "Codessa", `http://www.semichem.com/codessa/index.shtml`,.

[38] Schrödinger, "QikProp", `http://www.schrodinger.com/Products/qikprop.html`,.

[39] Duffy, E.; Jorgensen, W. Prediction of Properties from Simulations: Free Energies of Solvation in Hexadecane, Octanol, and Water. *J. Am. Chem. Soc.* **2000,** *122,* 2878–2888.

[40] Jorgensen, W.; Duffy, E. Prediction of drug solubility from Monte Carlo simulations. *Bioorg. Med. Chem. Lett.* **2000,** *10,* 1155–1158.

[41] R Development Core Team, "R: A Language and Environment For Statistical Computing", R Foundation for Statistical Computing, Vienna, Austria, 2004 ISBN 3-900051-07-0.

[42] Steinbeck, C.; Han, Y.; Kuhn, S.; Horlacher, O.; Luttmann, E.; Willighagen, E. The Chemistry Development Kit (CDK): An Open-Source Java Library for Chemo- and Bioinformatics. *J. Chem. Inf. Comput. Sci.* **2003,** *43,* 493–500.

[43] Guha, R. Using the CDK as a Backend to R. *CDK News* **2005,** *2,* 2–6.

[44] Kohonen, T. *Self Organizing Maps;* volume 30 of *Springer Series in Information Sciences* Springer: Espoo, Finland, 1994.

Chapter 2

Statistical & Optimization Techniques

QSAR studies can be broadly divided into two types - regression and classification. The development of QSAR models essentially consists of the application of statistical methods to chemical datasets. As such, the statistical and machine learning literature provides a number of useful techniques. Some techniques are specifically designed to build classification models whereas others can carry out both classification as well as regression. In addition to these techniques, a number of methods are available for the optimization of various parameters and selection of variables required in the model building process. These can be deterministic methods such as the BFGS algorithm[1–4] and the Nelder-Mead simplex algorithm[5] or stochastic methods such as genetic algorithms[6–8] and simulated annealing.[9] This chapter discusses the underlying details of the various modeling and optimization techniques used in this work.

2.1 Linear Methods

As the title of this section indicates linear methods employ a linear relationship between the predictor variables and the observed response to develop a predictive model. In many QSAR problems, structure property trends can be modeled reasonably well by linear approximations. In general it is observed that physical properties are well modeled by these types of methods. In the case of biological properties linear models do not always exhibit good predictive performance. The poorer behavior of linear models when faced with biological structure property trends is understandable when we consider the fact that biological properties in general are the result of a number of interactions that might include absorption, metabolic degradation, excretion and so on. Clearly the relationship between molecular structure and these factors is complex and in general nonlinear. However, linear methods are useful as a first step in the modeling process and, though not always very accurate, the simple interpretation methods that can be applied to linear models makes up, to some extent, for the lack of predictive ability for these methods. Though linear methods can be applied to both classification and regression we focus on the latter application in this section.

2.1.1 Multiple Linear Regression

A linear relationship between an observation's response (i.e., observed value) and its independent variables can be modeled by

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} + \epsilon_i \qquad i = 1, 2, \ldots, n \tag{2.1}$$

where y_i is the response for the i^{th} observation and $x_{i1}, x_{i2}, \ldots, x_{ip}$ are the independent variables for the i^{th} observation and n is the number of observations. $\beta_0, \beta_1, \ldots, \beta_p$ are parameters that are to be estimated. ϵ_i is the error term and is assumed to be a normally distributed random variable. Multiple linear regression is a technique by which y_i and $\beta_0 \ldots \beta_p$ can be estimated. Thus Eq. 2.1 may be written as

$$\hat{y}_i = b_0 + b_1 x_{i1} + b_2 x_{i2} + \ldots + b_p x_{ip} \tag{2.2}$$

where $b_0, b_1, \ldots, b_p$ are the estimated values of the parameters in Eq. 2.1. The most popular algorithm to estimate the parameters is the least squares method which considers the best fitting straight line to be that which minimizes the square of the error between the predicted response, $\hat{y}_i$ and the observed response, y_i.

Once the parameters have been estimated, the model quality can be ascertained in a number of ways. Two common measures of model quality are the R^2 value and the root mean square error (RMSE). The R^2 is also known as the Pearson coefficient and ranges from -1 to +1. The RMSE is defined as

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum (y_i - \hat{y}_i)^2} \tag{2.3}$$

Good models are characterized by high value of R^2 and low values of RMSE. However, it is well known that R^2 is not always a good indicator of model quality and in many cases can be misleading. An alternative to R^2 is Q^2 which is obtained by using a leave-one-out (LOO) cross-validation procedure. That is, the linear model is generated using the whole dataset excluding one point. The response for this point is then predicted using the model and this procedure is repeated for all the points in the dataset. The R^2 for these predictions is denoted by Q^2. Though this is more reliable than R^2, especially for small datasets, care should be taken as it has been shown to be a poor indicator of predictive ability in 3D QSAR[10] and 2D QSAR[11] models. Other indicators of model quality include the F-statistic and partial F-statistics.[12]

Individual predictions can be examined by a variety of methods. The simplest method is to plot residuals (the difference between the observed property and the predicted value) versus the observed response or index number. In either case a good model is characterized by a normal random distribution of residuals. Distinct patterns (such as upward or downward trends) are indicative of heteroscedasity and the dataset must be reexamined. The residuals may also be examined by making a normal probability plot (Q-Q plot). If the residuals do have a normal random distribution, this plot will be a straight line. Deviations from this will indicate shifts in location or scale as well as the presence of outliers.[13] When studentized residuals[12] are used, residuals lying above 2.0 or below -2.0 are traditionally designated as outliers. Outliers can also be determined by the use of regression diagnostics such as the Cooks distance and Mahalanobis distance. Once outliers have been detected the model can be regenerated excluding the outlying compounds from the dataset.

2.1.2 Robust Regression

An alternative to the multiple linear regression algorithm is to use a robust regression algorithm. As mentioned above the least squares algorithm attempts to minimize the squared deviations of the predicted response. This method is characterized by a breakdown point (a measure of the capability of an estimator to tolerate noisy data) of 0%. Robust regression utilizes alternative algorithms characterized by much higher breakdown points. Examples include the least median squares and least trimmed squares algorithms.[14] The advantage in using a robust regression method is that it uses algorithms that dampen the influence of *bad* points and attempts to take into account the whole dataset. Bad (or influential) points will be characterized by high residuals and thus robust regression combines model building and outlier detection in one operation. Thus the three-step process of model building, outlier detection, model regeneration is avoided.

2.2 Nonlinear Methods

Nonlinear methods can be considered a generalization of linear methods. Nonlinear estimation methods do not make any assumptions about the nature of the relationship between the predictor variables and the response. In general, the relationship must be specified in parametric form by the user. When considering nonlinear models, distinction

must be made between intrinsically linear and intrinsically nonlinear models. The former class of models can be transformed to a linear form and subsequently analysed using linear methods. Examples of these types of methods include logit and probit regression models. In the latter case, the nonlinear form of the model cannot be transformed to a linear form. Examples of this type of model include the general growth model and models used to determine drug responsiveness and half maximal response. In general, nonlinear models are essentially optimization problems. That is, the parameters are optimized to minimize certain criteria. Some approaches to nonlinear regression include least squares, maximum likelihood and function minimization (quasi-Newton and simplex methods).

In this work we focus specifically on the use of neural network algorithms for nonlinear classification and regression. Neural network algorithms are a specific class of nonlinear methods. They differ from traditional nonlinear methods in the representation of information extracted from the dataset. In contrast to nonlinear methods described above, neural networks do not represent the relationships within the data in an explicit functional form. The relationships in the dataset are encoded by a set of connections between units termed neurons. Methods have been described that attempt to represent this encoding in analytical form,[15] but this is not generalizable to all types of neural network algorithms. Essentially neural network algorithms attempt to mimic the behavior of a human brain and thus an essential feature of these algorithms is the ability to *learn* the relationships present within a dataset. In the words of Haykin[16]

> A neural network is a massively parallel distributed processor made up of simple processing units, which has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects:
>
> 1. Knowledge is acquired by the network from its environment through a learning process.
> 2. Interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge.

However, as pointed out by Ripley,[17] the above definition excludes a number of neural network algorithms such as the Kohonen network.[18] The neural network literature describes a large variety of neural network algorithms and Haykin[16] provides an extensive discussion of a variety of neural networks. In this section we focus on the two types

of neural networks used in this work, viz., feed-forward neural networks and the self organizing map.

2.2.1 Feed-Forward Neural Networks

The fundamental components of a neural network are neurons. Neurons are essentially computation units that accept an input value and generate an output value. In this sense, a neuron can simply be considered a mathematical function. The actual behavior of the network emerges from the connections between the neurons and weights assigned to these connections.

The neural network models considered in this work are termed 3-layer, fully-connected, feed-forward neural network models (which is a specific case of the multilayer perceptron[16]). The first layer is termed the input layer and each neuron in this layer corresponds to the input variables for the model. The second layer is termed the hidden layer and is responsible for nonlinearly combining the inputs. The final layer is termed the output layer, and in all the neural network models in this work, contains a single neuron whose output is the predicted property. The term fully-connected indicates that all the neurons in a given layer are connected to all the neurons in the next layer. Fig. 2.1 shows a schematic of this type of neural network. Let us now consider the internals of the network in a little more detail.

The role of a neuron is to accept input values and weights associated with connections to the neurons in the preceding layer. Essentially, a neuron consists of a function whose output represents a hyperplane that divides the input space of the neuron into two regions - an *on* region and an *off* region. The function used in a neuron is termed the transfer function and when this function is linear this interpretation holds exactly. However, the main reason for the utility of a neural network is that it exhibits the *universal function approximation* property[19,20] and to achieve this, the transfer function is generally nonlinear in nature. In this case, the above interpretation still holds to a good degree. A number of transfer functions have been reported in the neural network literature and the implementation used in this work utilizes a sigmoidal transfer function given by

$$O = \frac{1}{1 + \exp(-\sum x_i w_i + b)} \tag{2.4}$$

where O is the output of the neuron, x_i is the output value of the i^{th} neuron in the preceding layer, w_i is the weight for the connection between this neuron and the i^{th} neuron in the preceding layer and b is the value of the bias term. Fig. 2.2 is a graphical

representation of a hidden neuron. The diagram stresses the fact that the bias term can be considered as a neuron whose output value is always 1. The value of the bias term thus represents the weight for the connection between the *bias neuron* and the hidden neuron in question.

Fig. 2.3 shows a plot of the transfer function defined by Eq. 2.4. The weights provide a means for the neural network to assign importance to specific neurons. In the case of a 3-layer network, weights between the input and hidden layer neurons allow the network to be configured so that more important input variables will have greater contributions to the hidden layer neurons. It should be noted that in this type of configuration, the input layer neurons do not utilize the transfer function. Instead, the job of the input neuron is simply to scale the raw descriptor values to a suitable range (0.05 to 0.95 in the case of the current implementation). The role of the bias term can be understood in terms of a hyperplane interpretation of the transfer function. In this interpretation the bias term plays the role of the intercept term. This view is exact when the transfer function is linear. In effect the bias term shifts the hyperplane in the input space of a neuron to obtain an optimal partitioning of the space into on and off regions. An alternative view is that the bias term controls whether the neuron output is on (0 or close to 0) or off (1 or close to 1). In the case of a neuron with a sigmoidal transfer function, the bias term thus controls where on the sigmoidal function the output will lie. When the output value is close to zero, the neuron will have little contribution to the next layer, whereas when the output value is high, the neuron will have a significant contribution to the next layer. In effect, the weights and biases allow the network to *learn* the features present in the dataset and the optimal set of weights and biases encode these features allowing the network to make accurate predictions.

At this point we have described the anatomy of a neural network. The next step is to obtain a set of weights and biases that will allow the network to make accurate predictions. The number of weights and biases is defined by the configuration of the neural network. In the case of the 3-layer network, the number of weights and biases is defined by the number of input and hidden layer neurons. Intuitively, increasing the number of hidden layer neurons will lead to a more accurate network. However this will also lead to overfitting. One rule of thumb used to determine the suitable number of weights and biases is that the total number of parameters should be less than half the size of the training set used to build the model,[21] that is,

$$n_I n_H + 2n_H + n_O \leq \frac{n_{TSET}}{2} \tag{2.5}$$

where n_I, n_H and n_O are the number of input neurons, hidden neurons and output neurons respectively and n_{TSET} is the number of observations in the training set. The number of neurons in each layer define the *architecture* of the neural network. Thus, using the above notation, a 3-layer neural network is said to have a $n_I - n_H - n_O$ architecture. For all the neural network models reported in subsequent chapters the value of n_O is set to 1.

Once the number of parameters have been chosen, the next step is to obtain a set of optimal parameters. The network is initialized with a set of weights and biases generated using a combination of generalized simulated annealing and the BFGS algorithm. Then, each member of the training set is presented to the network. For each member, the network generates a prediction of the activity or property in question. After each training sample is presented to the network the prediction error is used to update the weights and biases. Traditionally the training procedure utilizes the backpropagation algorithm.[16] However, this algorithm is relatively inefficient and hence we use the BFGS quasi-Newton algorithm.[22] The important feature of the training phase is that it is supervised. That is, to train the network, the observed values of the training samples are required. This is in contrast to unsupervised methods (such as the self-organizing map) that do not require the observed values of the training set. Once all the examples in the training set have been presented to the network the process is repeated for a user specified number of cycles. It is also important to note that since the network is trained using a quasi-Newton algorithm, the optimized weights and biases depend on the initial configuration. As a result multiple runs are required to ensure that a representative result is obtained.

As mentioned above, too many parameters can lead to overfitting. In addition, as training progresses, the training RMSE will continually decrease and after a certain point the network will start to memorize the noise in the dataset. That is, the network will overfit the data. To prevent this, cross-validation is used. This procedure uses a portion of the dataset to measure the performance of the network, in terms of RMSE, at regular intervals during training. In effect, the cross-validation set acts as a pseudo external prediction set. Ideally the RMSE for the cross-validation set will smoothly decrease as training progresses and at one point will start to increase. This point represents the optimal configuration of the weights and biases and any further training beyond this point will lead to overfitting. In practice, the cross-validation RMSE does not always smoothly decrease, but in general, a global minimum does occur. The behavior of the RMSE values for the training and cross-validation sets are shown in Fig. 2.4. The use of the cross-validation set allows us to define a cost function which can be used to assess the

quality of a model by simultaneously taking into account its training and cross-validation performance. The cost function in the CNN algorithm used in this work is defined as

$$\text{Cost} = \text{RMSE}_{TSET} + 0.5 \times |\text{RMSE}_{TSET} - \text{RMSE}_{CVSET}| \tag{2.6}$$

where RMSE_{TSET} and RMSE_{CVSET} represent the RMSE values for the training and cross-validation sets. The form of the cost function penalizes models that have overfit, represented by high RMSE for the cross-validation set, and thus characterizes a given model better than simply considering the RMSE of the training set.

Once a model has been trained, its generalizability is then evaluated by passing an external prediction set and noting the RMS error. Ideally, one would expect similar RMSE values for the training, cross-validation and external prediction sets, but in general this is not the case. This aspect of model assessment is discussed in more detail in subsequent chapters.

2.2.2 Kohonen Self-Organizing Maps

A Kohonen self-organizing map (SOM) is an unsupervised neural network that uses only the independent variables of the dataset and is generally applied to classification problems. The SOM was first described by Kohonen[18] in the 1980's. The use of SOM's is widespread and examples of their application include the analysis and prediction of NMR spectra,[23,24] classification of reactions[25] and QSAR analysis.[26–28]

The SOM can be viewed as an elastic net of points in 2-D, which are molded to the specific features of the compounds used for training. In this sense, the SOM is also a dimension reduction algorithm. Training occurs as the SOM's neurons compete with each other for selection. At each training iteration, the selected neuron and its neighbors are modified to resemble the applied example compound.

SOM's can appear in a variety of forms[18] ranging from a square (or rectangular) grid to a hexagonal array. In this work we use a square configuration. In order that each neuron has the same number of neighbors, the grid is designed so that it wraps around the edges, effectively transforming the grid of neurons into a torus. However, for ease of visualization and discussion we will refer to the arrangement as a square grid.

Each compound in the training set is represented by a vector,

$$X_i = (x_{i1}, x_{i2}, \cdots, x_{in}) \tag{2.7}$$

where n is the number of independent variables employed. Each neuron on the square SOM grid is also a vector,

$$M_i = (m_{i1}, m_{i2}, \cdots, m_{in}) \tag{2.8}$$

where n is defined above. The neurons on the grid are initialized with random vectors. The size of the grid is chosen by trial and error, guided by a rule of thumb described by Chen,[29] which states that the number of neurons should be approximately one to three times the number of examples in the training set.

The training process for a SOM is iterative. Each training iteration involves comparing each member of the dataset to all the neurons in the grid and determining the grid neuron that is closest, in terms of Euclidean distance,

$$d_{pq} = \sqrt{\sum_{i=1}^{n}(x_{pi} - m_{qi})^2} \tag{2.9}$$

to the submitted neuron. The grid neuron that is most similar to the input vector is the winner. Then, the winning neuron and the surrounding neurons are modified, according to this equation:

$$m_i(t+1) = m(t) + h_{ci}(t)[x(t) - m_i(t)] \tag{2.10}$$

where t represents training iterations, m_i represents the winning neuron and x represents the training set member. Here $h_{ci}(t)$ is termed the neighborhood kernel, and it determines which neurons are neighbors and how such neighboring neurons will be modified. Neurons that are further away (in a topological sense) from the winning neuron are modified to a smaller degree than neurons that are closer. The simplest neighborhood kernel is the bubble function[18,30] (also referred to as a fixed window) which is non-zero for the neighborhood but zero elsewhere. The map in this work implemented a Gaussian kernel,[18] defined as

$$h_{ci}(t) = \alpha(t) \exp\left(-\frac{||r_c - r_i||^2}{2\sigma^2(t)}\right) \tag{2.11}$$

where $\sigma(t)$ is the neighborhood radius at time t which monotonically decreases with time. Thus, the number of neurons considered to be neighbors decreases as training progresses. The term $||r_c - r_i||$ represents the Euclidean distance between the winning neuron and the neighboring neuron. Thus, neighbors closer to the selected neuron will undergo a larger modification than neurons further away from the selected neuron. $\alpha(t)$ is the learning

factor, and it influences the extent to which a neuron should be modified. Initially, neurons within a large radius surrounding the selected neuron are considered neighbor neurons. The radius of the neighborhood is decreased in successive training iterations, and in the last stages of training only the nearest neighbors of the selected neuron are modified. The effect of this variable neighborhood function is that in the early stages of training the neurons are modified on a global scale, which leads to a global ordering. Near the end of training, the smaller neighborhood results in fine-tuning of the map features. The neighborhood function thus controls the sensitivity of the map.

The actual modification is controlled by the learning factor, $\alpha(t)$. The learning factor is a function that monotonically decreases from 1 to 0 as training progresses. Once $\alpha(t)$ reaches zero, training stops. Kohonen[18] mentions several ways of modifying $\alpha(t)$, and the implementation used in this work employs a constant decrement,

$$\alpha(t+1) = \alpha(t) - 0.01 \tag{2.12}$$

which implies that after 100 training iterations $\alpha(t)$ will be zero. This represents an upper limit on the number of training iterations.

As mentioned, the result of the training procedure is to create regions of cells on the map that are similar to each other. After training the neurons can be assigned classes by determining which training set member is the closest (in an Euclidean sense) to a given neuron and assigning the class of that training set member to the neuron. Once this is done for all the neurons, a new observation can be classified by assigning the class of the closest neuron in the map, to it. An example of this approach to classification can be found in Chapter 4. An alternative usage is to avoid explicit classification of the neurons in the map and instead cluster the training set members (as well as new observations). In this approach one would simply assign observations to neurons based on Euclidean distance between the members and each neuron. As a result, certain neurons may be assigned more than one training set member and some neurons will not be assigned any. The number of members assigned to a given neuron can be used to compute a density which can then be used to color code the map providing an easy visual display of the topology of the dataset. An example of this approach is shown in Fig. 2.5, which represents a SOM trained using a portion of the NCI AIDS dataset.[31]

2.3 Algorithmic Methods

Breiman[32] categorized a number of statistical methods as algorithmic owing to the fact that they are essentially *model free*. That is, these methods do not help us to understand the relationship between predictor variables and the response by developing a model relationship. However, their utility lies in the fact that they can be used as black box prediction methods and usually show good predictive ability for both classification and regression. There have been a number of applications of algorithmic methods in the physical and medical sciences.[33–36] This class of modeling techniques includes prototype methods such as k-means clustering and learning vector quantization[37] as well ensemble methods such as the random forest technique. In this section we describe two algorithmic techniques used in this work.

2.3.1 Random Forests

The random forest (RF) method was developed by Breiman as an extension of the decision tree[38] technique. Decision trees are non-parametric, nonlinear models that allow the user to easily understand how or why an observation is classified or predicted. They have been extensively used in the medical field[39,40] as well as in various chemical[41–43] and biological[44,45] applications.

The goal of the decision tree technique is to split the dataset into a tree-like structure, using a single descriptor at each split point. A variety of algorithms to achieve this are available and we focus on the recursive partitioning algorithm.[46] The dataset (also known as the root node), D, is first split into two nodes, say D_1 and D_2, based on the value of a selected descriptor, say X_i. If X_i is binary in character then the j^{th} observation from D is placed in D_1 or D_2 depending on whether the value of X_i for the j^{th} observation, denoted by X_{ij}, is 0 or 1. In case X_i is real valued, a specific value of X_i, say x_i is calculated such that if $X_{ij} < x_i$ then the j^{th} observation goes into D_1 or goes into D_2 otherwise. The method by which a descriptor is selected is based on a quantity known as *purity*. All the available descriptors are considered individually as candidates for the splitting decision. The purity of a split can be defined in a number of ways. One possible approach is to define the purity of a split as the fraction of observations, in the resultant nodes, that will be of a single class. The descriptor that leads to the highest value of purity will be selected to perform the splitting. Other definitions of the purity include the Gini index, χ^2 and G^2.[38] It should be noted that the same descriptor can be chosen at multiple split points.

After all the observations have been placed in either one of the nodes, the algorithm considers each node and performs the same operation described above. Thus D_1 would be split into two nodes, say D_3 and D_4 and similarly for D_2. A flowchart summarizing the algorithm is shown in Fig. 2.6. If this process is repeated continously, the end result will be a *perfect tree*, where each node contains a single observation. Nodes which cannot be split further are termed leaf nodes. Such a tree will perfectly predict the data used to build the tree but will be nearly useless for new observations. Thus a perfect tree will overfit the data. As a result, heuristics are used to determine when the tree should stop growing. These include specifying a minimum node size, such that nodes with fewer observations will not be split further. Alternatively, a node is not split, if it exhibits a purity greater than a certain specified value or the purity does not increase as a result of the split. Other possibilities include a variety of cross-validation methods. These heuristics are collectively known as pruning rules and detailed discussions of this area can be found in the statistical literature.[39,46,47] An example of a tree is shown in Fig. 2.7. The figure represents a decision tree for a hypothetical classification problem. Three descriptors were available at each split point and the fractional purity measure was used to select a descriptor at each split. As can be seen, this measure results in each node, generated by a split, consisting mainly of a single class.

After a tree of the required size (i.e., number of nodes) has been created it can then be used for predictive purposes by determining in which leaf node a new observation would belong, and assigning a class based on the majority class of the leaf node or calculating a property value, by averaging the property values of the observations contained in the leaf node.

The philosophy behind the the RF method is that the technique is able to provide high predictive ability by averaging the predictions of a large number of individual decision trees. In other words, the RF technique is an ensemble method based on *forests* of decision trees. The technique is applicable to both classification and regression. The following discussion presents the fundamentals of the RF technique and its use in providing a measure of variable importance.

Since the RF method is based on decision trees, the features of the latter that make it an attractive option in the development of predictive models also apply to RF's. These features include the ability to handle high-dimensional data, the ability to ignore irrelevant variables (thus obviating the need of feature selection) and the ability to provide some measure of interpretability. However one of the major drawbacks of the decision tree method is its low predictive ability. Random forests, as an extension of the

decision tree method, exhibit a higher predictive ability coupled with other features such as an internal measure of accuracy and a measure of variable importance.

Random forests are closely related to other tree based ensemble techniques such as bagging,[48] boosting[49] and random split selection.[50] The method consists of generating an ensemble of N trees denoted by[51]

$$R = \{T_1(X), T_2(X), \ldots, T_N(X)\} \tag{2.13}$$

where X is defined as a p-dimensional vector of descriptors. The output of this ensemble is be denoted by

$$\{\hat{Y}_1, \hat{Y}_2, \ldots, \hat{Y}_N\} \tag{2.14}$$

where $\hat{Y}_i$ is the output of the i^{th} tree. The predicted value reported by the random forest, $\hat{Y}$, is then the majority vote of the N outputs (for classification) or the mean of the N output values.

The algorithm to build the forest can be described as follows. Consider a training set, D, of n observations and p independent variables (represented by p-dimensional vectors). The observations can be categorical (for classification) or real valued (for regression). A random subset of the training data is selected with replacement and a decision tree is built using this *bootstrap* sample. The evolution of the decision tree is slightly different from the original algorithm described by Breiman[38] in that, at each node m ($1 \leq m \leq p$) descriptors are selected randomly from the descriptor pool. When $m = 1$ this is equivalent to random splitting and when $m = p$ this is equivalent to bagging.[48] Using this rule to split nodes, a tree is grown to its maximal size and pruning is not carried out. Another bootstrap sample is selected and another tree is grown. This procedure is repeated till the requisite number of trees (N) have been grown. Fig. 2.8 summarizes the steps involved in the creation of a random forest model as a flowchart.

The fact that subsets of the dataset are used to build the forest allows us to use the observations not used during building (termed out-of-bag or OOB observations) to obtain a measure of predictive performance. This measure is termed the out-of-bag estimate[52] and can be considered a parallel cross validation since it is estimated for each training step. The OOB estimate is obtained by considering the OOB part of the data for the i$^{\text{th}}$ tree, denoted by D_i^{OOB}. The i^{th} tree is used to predict the property of the observations in D_i^{OOB}. It has been shown[36] that on average each tree uses approximately $1 - e^{-1} = 2/3$ of the whole dataset and hence the size of D_i^{OOB}, is on average 1/3 of the dataset. This implies that each observation will be in the OOB data about 1/3 of

the time. Consequently, the OOB estimates can be aggregated to provide an ensemble prediction for each observation. In the case of regression, this result is an out-of-bag estimate of the mean square error (MSE) that can be used to approximate the MSE for the entire ensemble of trees and can be written as[51]

$$MSE \approx MSE^{OOB} = \frac{1}{n}\sum_{i=1}^{n}\left[\hat{Y}^{OOB}(X_i) - Y_i\right]^2 \tag{2.15}$$

It has been shown[51] that the MSE obtained by Eq 2.15 agrees well with the results of a k-fold cross validation scheme.

In addition to obtaining performance estimates of the random forest, the OOB dataset also allows us to obtain a measure of importance of the descriptors in the pool supplied to the algorithm. The measure is obtained by first calculating the predicted values for the OOB data for a given tree. Next, each descriptor in the OOB dataset is individually scrambled and predictions are made on the scrambled OOB dataset, for each scrambled descriptor. This procedure is repeated for all the trees grown in the forest. After training is complete, the overall OOB estimates (MSE) for the unscrambled and scrambled sets can be evaluated using Eq 2.15. The importance of the j$^{\text{th}}$ descriptor is then given by

$$\text{Importance}_j = MSE - MSE_j \tag{2.16}$$

where MSE represents the OOB estimate for the unscrambled data and MSE_j represents the OOB estimate for the datasets in which the j^{th} descriptor was scrambled. This procedure allows the ranking of the descriptor used in the random forest in order of relative importance. Descriptors that play a more important role in the predictive ability of the model will have a higher value of importance compared to descriptors that play an insignificant role. The importance measure for each descriptor can be plotted to allow easy visual inspection as shown in Fig. 2.9. From this figure it is clear that SURR-5 is significantly more important than the other descriptors owing to its large separation on the X axis. These plots are used in Chapter 7 to provide a comparison with interpretation schemes for other types of models.

2.3.2 k-Nearest Neighbor Algorithm

The k-nearest neighbor (kNN) method is very simplistic in nature and assumes that observations that are close in the space of the predictor variables will be close to each other in the space of the response variable. This method can be applied to both

regression as well as classification problems, though when faced with high-dimensional data, kNN regression does not perform very well.[36]

In the case of regression the kNN fit for the i^{th} observation is defined as

$$\hat{Y}(x_i) = \frac{1}{k} \sum_{x_j \in N_k(x)} y_j \tag{2.17}$$

where $N_k(x)$ is the set of k points closest to x_i, that is, the neighborhood of x_i. Eq. 2.17 simply averages the observed values of the k nearest neighbors of x_i to obtain the predicted response for this observation. From the above discussion, the first step of the kNN method is to obtain the neighborhood for a given observation. This implies the choice of a distance metric. The most common metric used is the Euclidean distance, defined as

$$d_{ij} = ||x_i - x_j|| \tag{2.18}$$

where x_i and x_j are the independent variables for the query observation and prospective neighbor respectively. Other possibilities include the Manhattan distance and Mahalanobis distance though the choice of distance metric does not appear to affect the results significantly.[53]

In the case of kNN classification, the class of a query observation is simply the majority class of its nearest neighbors. Fig. 2.10 shows a schematic diagram of the working of the kNN algorithm. In the figure, the central white point is the query point and the points connected to it correspond to its three nearest neighbors. In the case of kNN regression, the property of the query point would be the average of the property of the nearest neighbors. In the case of kNN classification, the class of the query point would be the majority class of the nearest neighbors and in this case, the query point would be classified as *blue*. As opposed to kNN regression, kNN classification performs reasonably well when faced with high-dimensional data. This is due to the trade off between bias and variance. It has been shown[54] that in the case of a 1-nearest neighbor classifier the asymptotic error rate is never more than twice the Bayes error rate. It is evident that this observation considers the asymptotic region and hence assumes that the bias of the nearest neighbor rule is zero. In real problems (especially high-dimensional cases) this is not always the case and the bias term can be substantial. Nearest neighbor classification implicitly assumes that the class probabilities are approximately uniform within the neighborhood of the query point. In the case of high-dimensional datasets, this assumption does not necessarily hold and in fact, when the number of dimensions is high

the class probabilities might vary significantly in a certain direction. One approach to alleviating this problem is the use of adaptive metrics which modify the distance metric so that class probabilities in the resultant neighborhoods do not vary significantly. Examples of such adaptive nearest neighbor methods include that proposed by Friedman[55] and the discriminant adaptive nearest neighbor (DANN) rule described by Hastie et al.[56]

Given a suitable distance metric a kNN algorithm only requires that a suitable value of k be chosen. In many cases setting k to 1 provides reasonably good predictive performance for classification purposes. In general, optimal values of k are obtained via trial and error. A more systematic approach is to use a cross-validation scheme to obtain the best value of k for a given dataset. One example of this approach has been described by Shen et al.[57] in which the value of k along with the selection of variables used for classification were optimized simultaneously.

2.4 Optimization Methods

Optimization is a fundamental topic in QSAR modeling. In this context, there are two main applications of optimization techniques. The first involves the optimization of parameters for a model such as the weights and biases in a feed-forward neural network described previously. The other area where optimization plays an important role is in the selection process. Here, selection can mean the selection of compounds from a library or design of a library from various components[58–60] or it can mean the selection of descriptors to build models with. In this section we concentrate on the latter form of selection. As will be described in Chapter 3, the model building process begins with the generation of a large number of descriptors. In the interests of parsimony, our goal is to use the minimum number of descriptors to develop a good predictive model. Thus, we must select *good* subsets of descriptors. The definition of *good* depends on the modeling technique used to build the models after descriptor selection and will be discussed in the following sections. Though statistical methods exist to perform descriptor selection (or variable selection as it is termed in the statistical literature) such as stepwise regression, backward elimination and forward selection, these methods are generally restricted to linear regression models. However, a more important reason for avoiding these methods is due to a number of inherent drawbacks which include falsely narrow confidence intervals,[61] incorrect p-values, biased regression coefficients that require shrinkage,[62] severe problems with collinearity[63] and so on. Furthermore backward or forward selection algorithms by their nature will ignore certain combinations of variables (since in the former

case variables removed from consideration are not considered again and in the latter case variables are based on the current subset that has already been selected). Owing to these restrictions we avoid statistical selection algorithms and instead focus on optimization algorithms to carry out descriptor selection.

Optimization methods can be divided into two broad classes: deterministic and stochastic. Examples of deterministic methods include the BFGS algorithm[1–4] and the Nelder-Mead simplex algorithm.[5] Examples of stochastic methods include the genetic algorithm[6] and the simulated annealing algorithm.[9] The choice of method largely depends on the nature of the solution space. Deterministic methods are preferred in cases where there is known to be one global minimum in the solution space and when the dimensionality of the solution space is relatively small. Multiple local minima can exist and various improvements to the standard algorithms are available to overcome this situation. When the solution space is very large (or possibly combinatorial in nature) and may have multiple minima but no distinct global minimum, a stochastic algorithm which is able to effectively *sample* the solution space is the preferred option. This does not imply that a stochastic method cannot be used in the former case. Genetic algorithms have been used to optimize the weights and biases in neural networks.[64,65] In the various applications discussed in this work parameter optimization has been carried out using deterministic methods whereas descriptor selection has been carried out using stochastic methods. The following sections describe the principles underlying the genetic algorithm and simulated annealing and details of their implementations.

2.4.1 Genetic Algorithms

A genetic algorithm is a member of the class of optimization algorithms known as evolutionary algorithms, which utilize the concepts of biological evolution to develop efficient optimization strategies.[8] GA's have been used widely in the field of QSAR modeling,[66–68] cheminformatics[59,69,70] and chemometrics.[71–74] The application of genetic algorithms in this work are focused on their use as efficient tools to search large dimensional spaces. More specifically, one application of GA's in QSAR modeling is to search a descriptor space to find optimal subsets of descriptors that can be used to build predictive models. Fig. 2.11 shows a flowchart of the generic genetic algorithm and this section describes the steps in detail.

As mentioned above, a GA is based on the principles of evolution. As a result much of the terminology from the field of biological evolution has been adapted for

use in the field of genetic algorithms. Thus we define an individual as consisting of a chromosome and an associated fitness value. When using a GA for descriptor selection, the chromosome is simply a subset of descriptors (of user specified length) chosen from the descriptor pool that is being searched. A population is defined as a collection of individuals. The first step of the GA is to initialize the population. This is achieved by randomly generating a user specified number (usually 40 to 50) of descriptor subsets of user specified size. Each descriptor subset is used to build a model (which can be a linear regression model or a CNN model). The root mean square error (RMSE) for each model is used to determine the fitness of the individual. The implementation used in this work does not use the raw RMSE value but instead uses a linearly scaled form. The actual form of the fitness function depends on the nature of the model to be developed. For linear models the fitness for the i^{th} individual in the population is defined as

$$\text{Fitness}_i = \left(2 - \frac{\text{RMSE}_i}{\text{RMSE}_{\text{avg}}}\right)^{-1} \tag{2.19}$$

where RMSE_i is the RMSE for the i^{th} individual and RMSE_{avg} is the average RMSE for the whole population. In the case of CNN models, the fitness function is defined by the cost function described in Section 2.2.1. Once the fitness for each individual has been evaluated, the population is ranked.

The next step is to create a child population. First a mating list is created, which is of the same size as the current population. Those individuals with fitness greater than the population average (which from Eq. 2.19 is greater than 1.0) are automatically placed in the mating list. By definition, this will fill up half of the available slots. The remaining slots in the mating list are filled by using a roulette wheel selection procedure[6] to select individuals from the current population. Once the mating list is created a child population is then generated by successively selecting two individuals from the mating list at random and applying genetic operations.

The first operation is termed crossover, and involves the the swapping of portions of the chromosomes of a pair of individuals. The GA literature describes a number of variations of the crossover operation.[6] The current implementation restricts itself to the single point crossover. In this type of crossover a split point is chosen in the descriptor subset. Then the descriptors from one side of the split point in the two individuals are swapped to give rise to two new individuals. This operation is shown graphically in Fig. 2.12. The figure represents a crossover performed on two individuals having a chromosome (descriptor subset) of length 5. The split point is chosen at the fourth

descriptor and the descriptors on the left of the split point are swapped resulting in two new individuals. The goal of crossover is to generate new individuals that will have the good features of the parent individuals. That is, if two individuals have a high fitness this implies that certain parts of their chromosomes (i.e., certain descriptors) are responsible for their fitness. By combining a portion of the chromosomes of two fit individuals, we expect that the children will exhibit equal if not better fitness.

The second genetic operation is termed mutation and is performed on a single child individual. It should be noted that mutation is not performed on all individuals in a population but is carried out only 5% of the time, mirroring the low frequency of mutation in biological evolution. In a genetic algorithm the mutation operation is performed by randomly changing a part of the chromosome of an individual. That is, a random descriptor within an individual is replaced with a randomly chosen descriptor from the descriptor pool. This is shown schematically in Fig. 2.13. The goal of the mutation operation is two-fold. First, random mutations prevent the algorithm from getting stuck in a local minimum and second, mutations prevent the phenonemon of premature convergence. This occurs when the algorithm creates very similar (or even identical) individuals whose fitness is high, but not necessarily optimal. The mutation operation can also be viewed as a method to maintain diversity within a population, though this does not entirely solve the problem of premature convergence as noted by Goldberg.[6]

With the application of these two operations we end up with a second, child, population. The fitness of the individuals in this population are evaluated and the individuals ranked. The second generation population is then created by randomly selecting individuals from the the top 50% of the previous population and the child population. Finally, if the best model in the child population is of lower fitness than the best model from the previous population, the best model from the previous population is kept in the second generation. With the formation of the second generation population, the whole process is repeated. This continues for a user specified number of cycles (usually 1000) and at the end the top ranked individuals (i.e., the top ranked descriptor subsets and associated RMSE values) are reported to the user.

2.4.2 Simulated Annealing

Simulated annealing is a generalization of the Metropolis Monte Carlo method[75] to optimization problems. The original method was devised as an efficient way to evaluate the Boltzmann average of a given atomic or molecular property. The method was extended by Kirkpatrick et al.[9] to determine the most stable state of a system. This modification was based on the physical phenonemon of annealing in which a melt (such as a glass or metal) is initially at a high temperature and then allowed to cool slowly, such that at any time, the melt is in thermal equilibrium. As the temperature is decreased the atoms in the melt will achieve increasingly ordered states and when the final temperature (say, room temperature) is reached the configuration of the atoms should be that of the most stable state.

This modification of the Metropolis method is easily extendable to combinatorial optimization problems and more specifically for QSAR, feature selection. In terms of an optimization problem, the temperature term in the simulated annealing algorithm effectively controls the size of the solution space and the cooling schedule narrows the space over time, allowing the algorithm to reach the global minimum of the solution space.

The original algorithm described by Metropolis et al. considered an initial thermodynamic configuration of a system with energy E at a temperature T. This configuration was perturbed and the change in energy, ΔE, was evaluated. If the change in energy was negative the new configuration was accepted and if positive, the configuration was accepted with a probability equal to the Boltzmann factor, $\exp(-\Delta E/kT)$. This procedure was repeated a number of times to obtain sampling statistics and then the temperature was reduced by a small amount. The whole procedure was then repeated till the final temperature was achieved. In terms of a feature selection problem, the thermodynamic configuration is replaced by a set of descriptors and the energy is replaced by a cost function which in the case of the studies presented in this work is either a linear regression routine or CNN routine. Thus, the algorithm starts out with a random descriptor subset (configuration), say x_0, selected from the descriptor pool and the value of the cost function (which is the RMSE for linear models and Eq. 2.6 for CNN models) for this descriptor subset is calculated, say $C(x_0)$. Next, the descriptor subset is perturbed by randomly replacing a single descriptor by a randomly selected descriptor from the pool. The value of the cost function for this new configuration is then determined, say $C(x)$. If $C(x) < C(x_0)$ the new configuration replaces the previous one and we repeat

the perturbation process. If $C(x) \geq C(x_0)$ then a detrimental step has been taken. The new configuration is accepted with a probability equal to the Boltzmann acceptance probability, P. If the new configuration is still not accepted, the algorithm replaces the new configuration with the old one and returns to the perturbation step. The working of the algorithm is shown schematically in Fig. 2.14. The whole procedure results in a single, low cost descriptor subset. The algorithm is then repeated with a different random descriptor subset to build up a pool of low cost descriptor subsets which can then be investigated in more detail.

The simulated annealing algorithm used in this work is an implementation of generalized simulated annealing described by Bohachevsky.[76,77] This method differs from the classical simulated annealing algorithm by introducing a step size Δr and a normalized n-D vector v. The vector v corresponds to a set of random perturbations of the current configuration x. v is obtained by generating a set of random numbers from $N(0,1)$ denoted by $u_i, i = 1 \ldots n$ and evaluating

$$v_i = \frac{u_i}{\sum_{i=1}^{n} u_i^2} \tag{2.20}$$

The new configuration y is then given by

$$y = x + \Delta r v \tag{2.21}$$

which lies in the neighborhood of x. The second modification is that the acceptance probability P, approaches zero as the configuration approaches the global optimum. This gives P the form

$$P = \exp\left(-\beta \frac{C(y) - C(x)}{C(x) - C_{est}}\right) \tag{2.22}$$

where $C(x)$ and $C(y)$ are the values of the cost function for the configurations x and y and C_{est} is the estimated global optimum. β is a parameter that controls the cooling schedule and corresponds to the effective temperature term, kT, in the Boltzmann factor. The choice of β is important as too small a value will result in a random walk and too large a value will cause the algorithm to converge to a local minimum. β begins with a small value which allows detrimental steps to be taken relatively frequently. This allows the algorithm to explore a larger space. As the algorithm progresses, β is increased making the probability of acceptance of a detrimental step lower, thus shrinking the search space. To determine the initial value of β the algorithm is run for a set number of iterations and β is adjusted until the relation $0.5 < \bar{P} < 0.9$ (where $\bar{P}$ is the mean

probability of a detrimental step) is achieved. In the case of the current implementation, the algorithm is allowed to run for 1000 iterations and for each detrimental step, the equation

$$P = \exp\left(-\beta \Delta C\right) \tag{2.23}$$

is solved assuming $P = 0.8$. Here ΔC is the difference in the value of the cost function for the detrimental configuration and the previous configuration. At the end of the iterations the average value of β is taken as the starting value for the actual run. To prevent premature convergence, β is multiplied by 2 every 100 iterations for a maximum of 50000 iterations. If detrimental steps occur more than 900 times in a row β is reset to the starting value and if this occurs twice in a row the algorithm exits.

2.5 Conclusions

In this section I have presented the underlying details of the modeling and optimization algorithms used in this work. For each class of modeling technique or optimization technique described here, there is a number of alternatives that have not been discussed. These include variants of the fundamental neural network model such as the probabilistic neural network and various types of linear modeling techniques such as ridge regression, linear discriminant analysis and so on. The QSAR literature has numerous applications of these and other modeling techniques. The field of optimization is certainly much more detailed than has been described in this chapter and, the literature describes numerous algorithms and variants that are suited for both general use as well as for special cases. However, the focus of this work is not on the modeling or optimization techniques themselves. Rather, the goal of this work is to develop and implement techniques that allow us to obtain meaningful knowledge from data using predictive or descriptive models. The methods described in this chapter allow us to achieve the first step, namely, the development of the model itself. Subsequent chapters describe various methods that have been developed to ensure validity and provide interpretability.

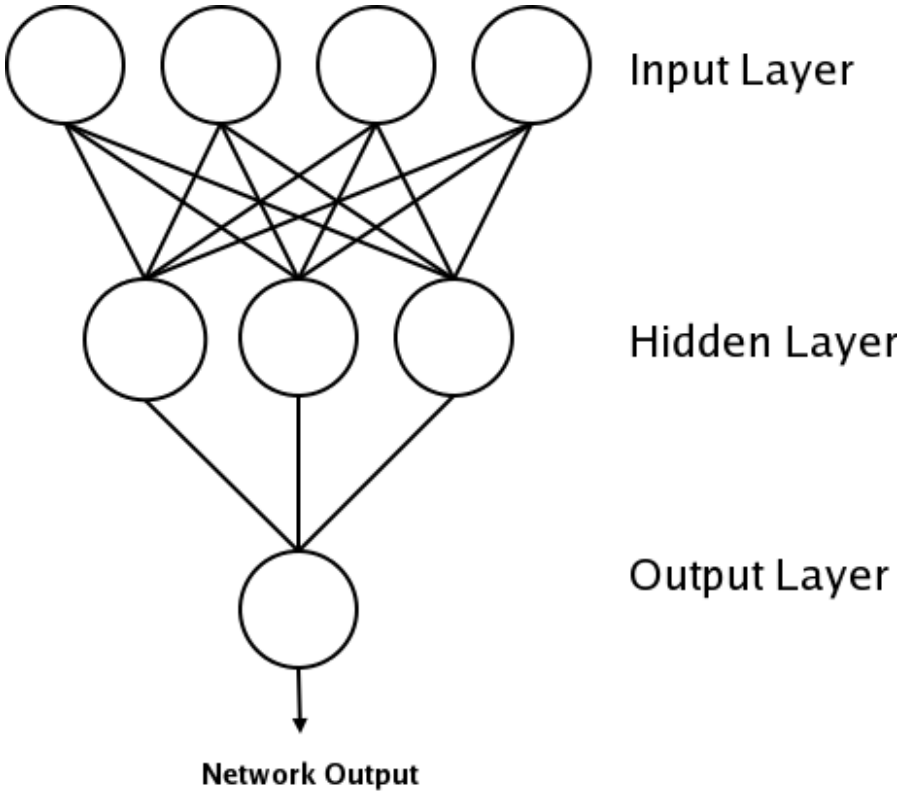


Fig. 2.1. A schematic diagram of a 3-layer, fully connected feed-forward neural network

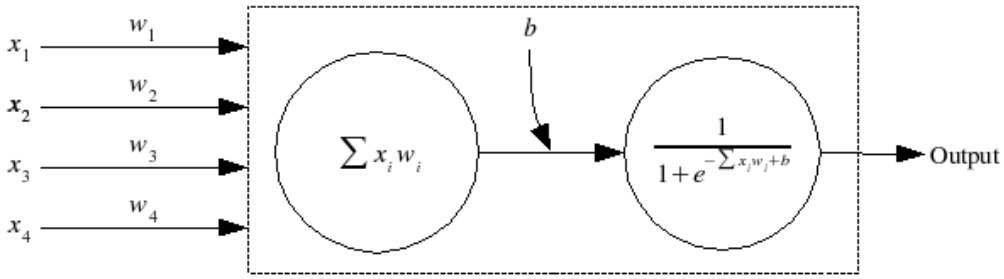


Fig. 2.2. A more detailed view of a single hidden layer neuron. The x_i's represent the output value of the neurons in the preceding layer and w_i's correspond to the weights for the connections between this neuron and those in the preceding layer. b represents the bias term for this neuron.

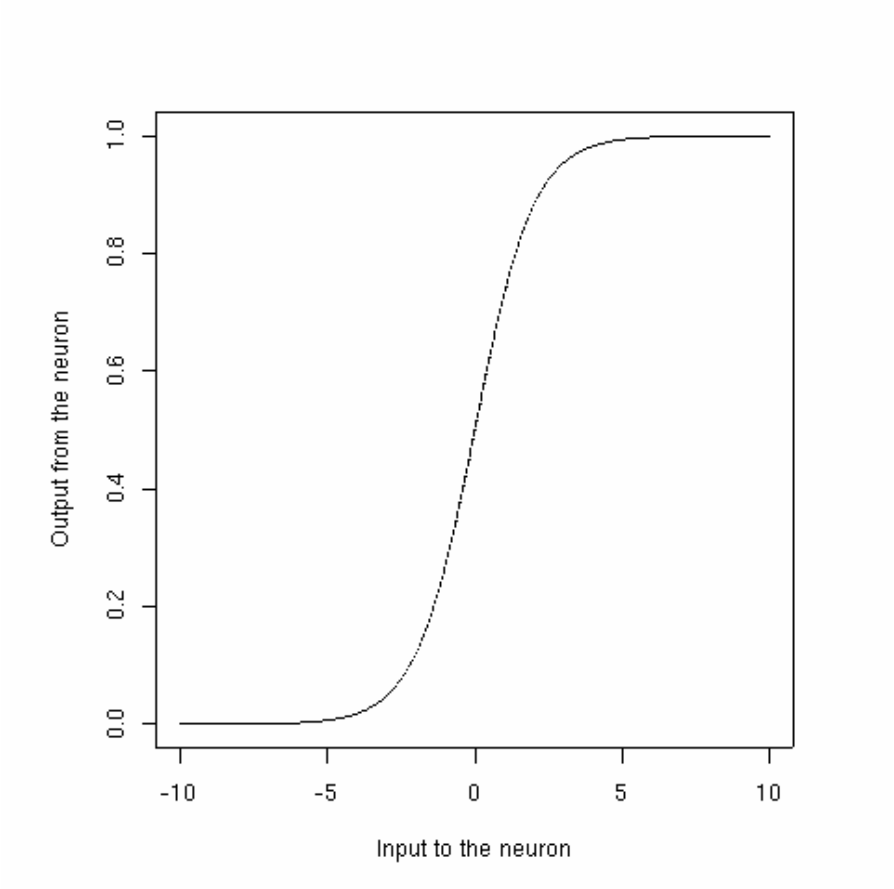


Fig. 2.3. A plot of the signmoidal transfer function used in the implementation of the neural network algorithm in this work

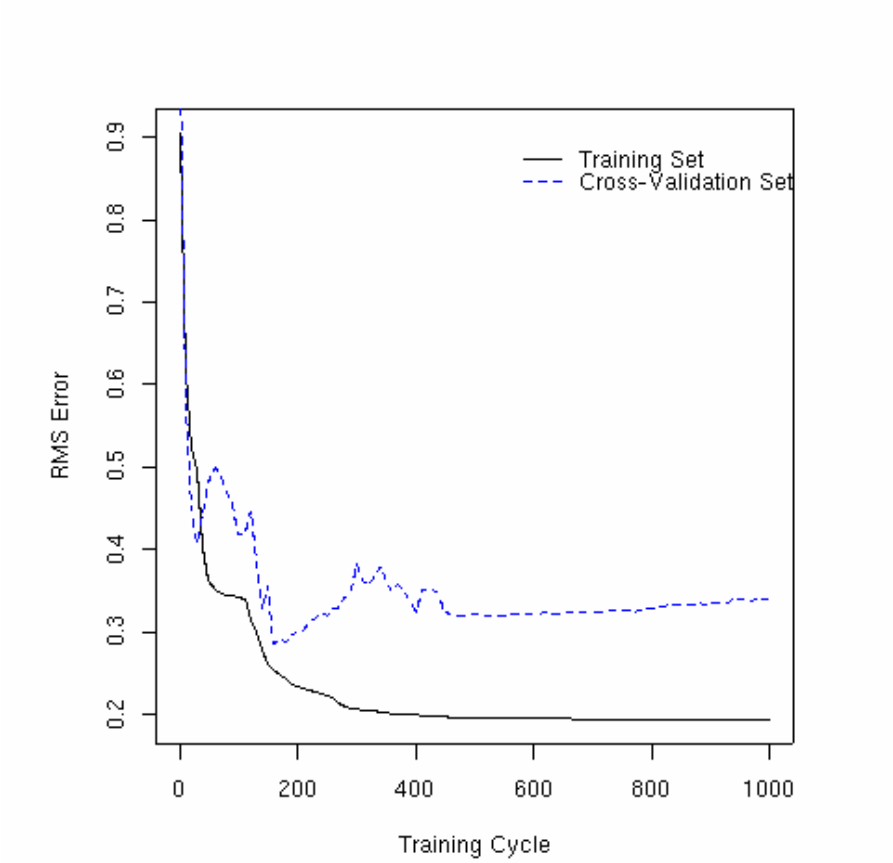


Fig. 2.4. A plot showing the variation of training set and cross-validation set RMSE with training cycle. The global minimum of the cross-validation curve indicates the training cycle at which the optimal weights and biases occur

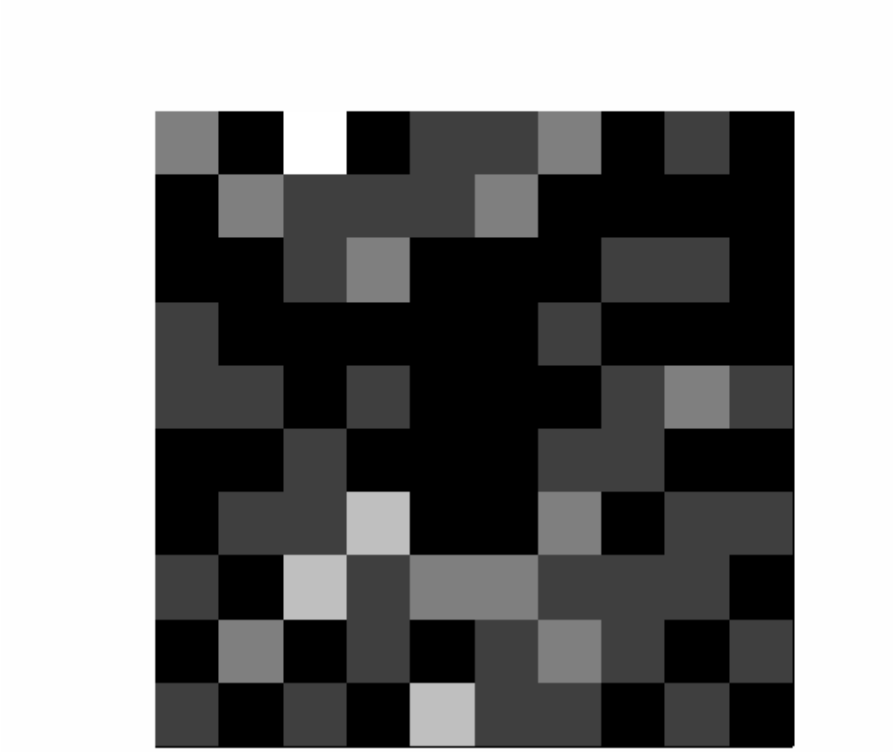


Fig. 2.5. A clustering of the first 5000 compounds from the NCI AIDS test dataset[31] obtained using a SOM. The grid dimensions are 10×10 and the neurons are color coded based on the number of compounds that map to them. Thus black neurons have no members from the training set mapped to them whereas the white neurons have the maximum number of observations mapped to them.

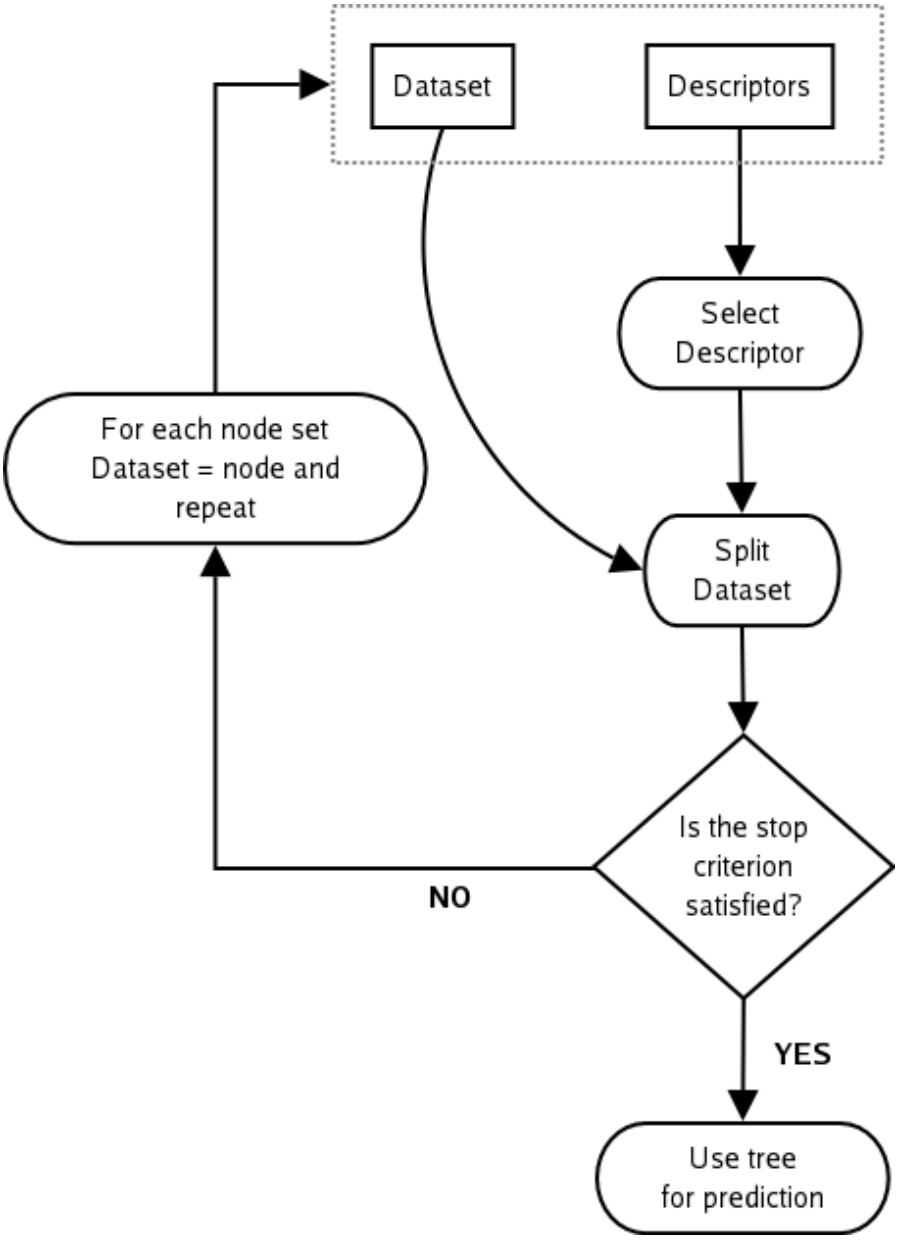


Fig. 2.6. A flowchart illustrating the recursive partitioning algorithm used to generate a decision tree

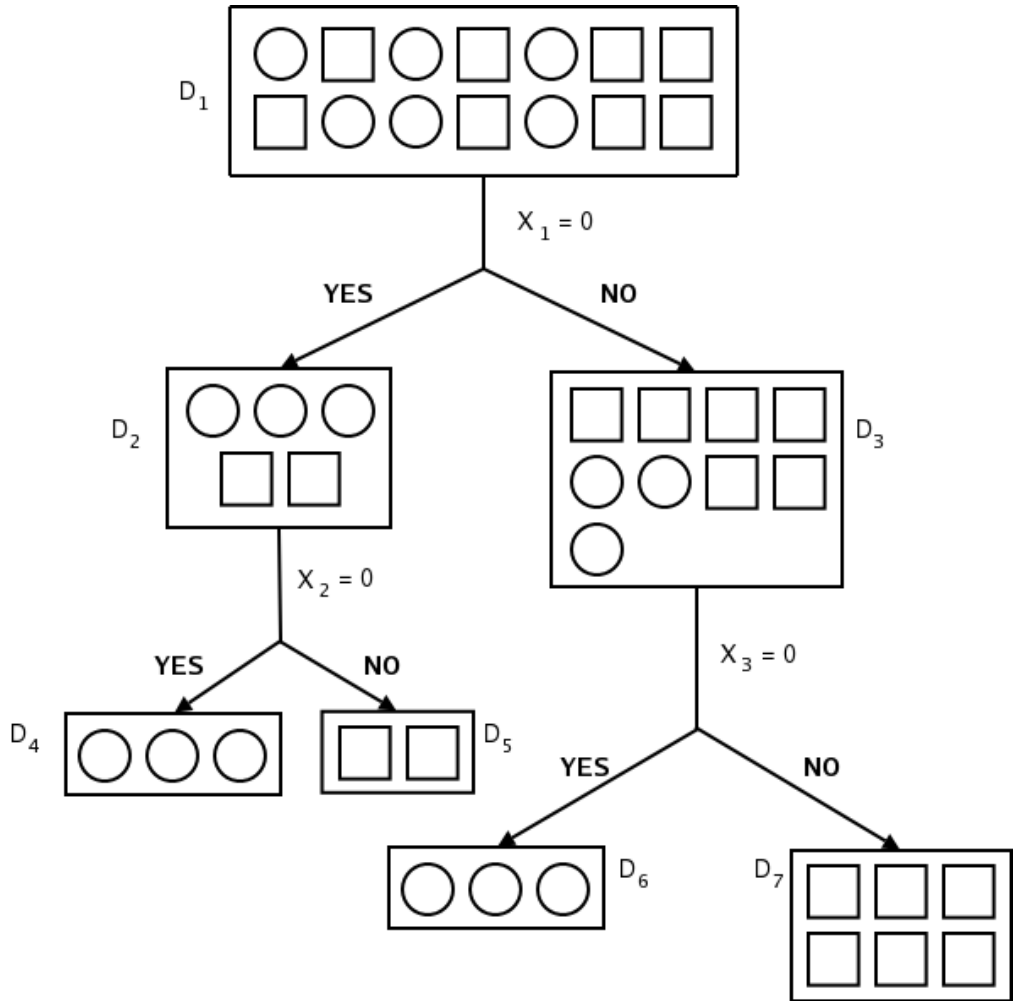


Fig. 2.7. A schematic diagram of a decision tree for a hypothetical clasification problem. The purity measure used to grow the tree was defined to be the fraction of a single class in the nodes created by a prospective split. Three binary descriptors, X_1, X_2 and X_3 were available for splitting and the D_i's correspond to each node. Nodes D_4, D_5, D_6 and D_7 represent leaf nodes.

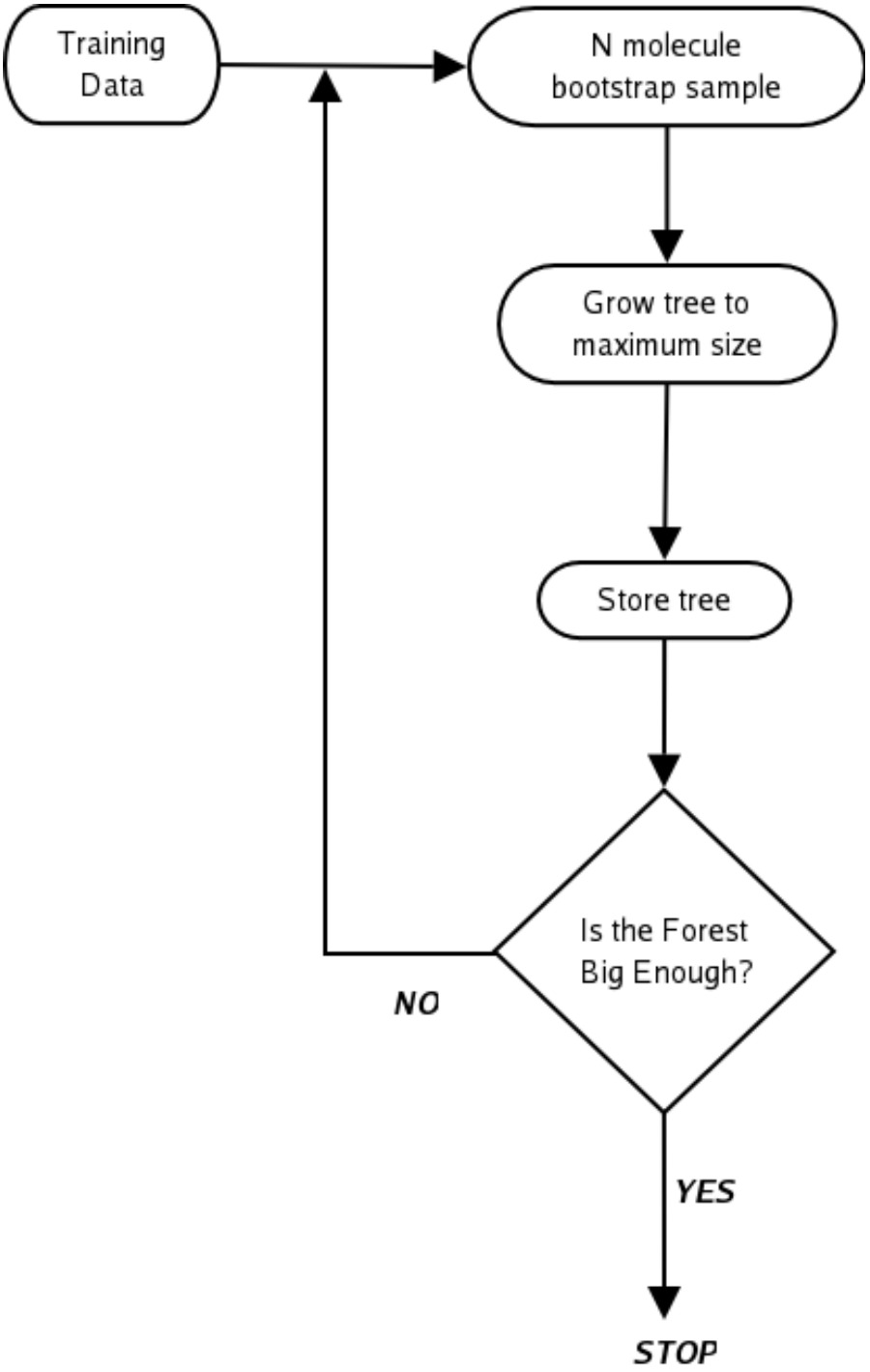


Fig. 2.8. A flow chart describing the working of the random forest algorithm

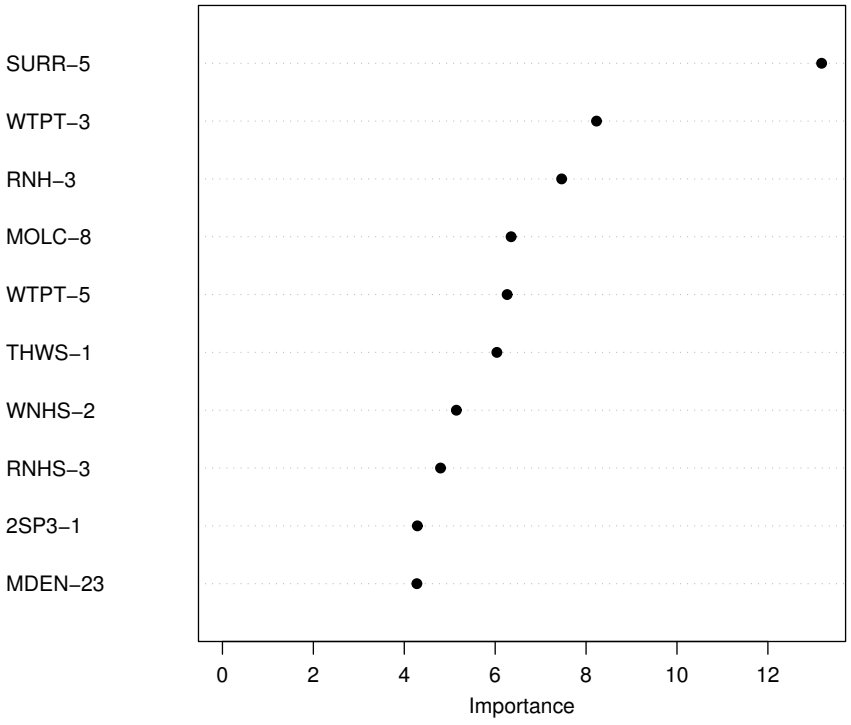


Fig. 2.9. A random forest descriptor importance plot

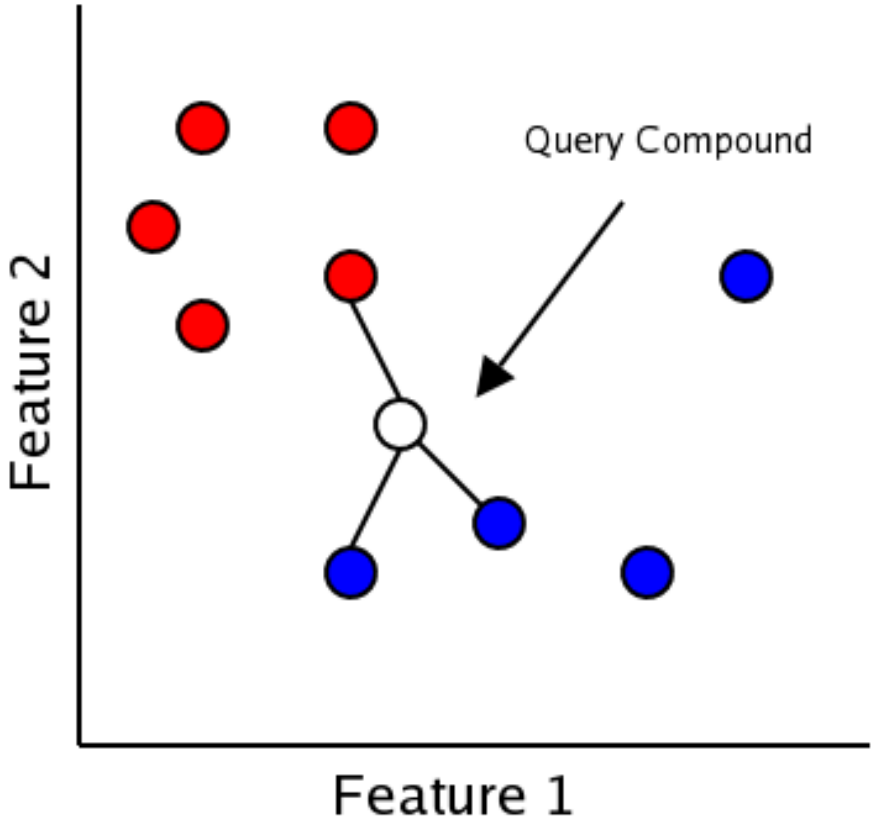


Fig. 2.10. A schematic diagram of the kNN algorithm

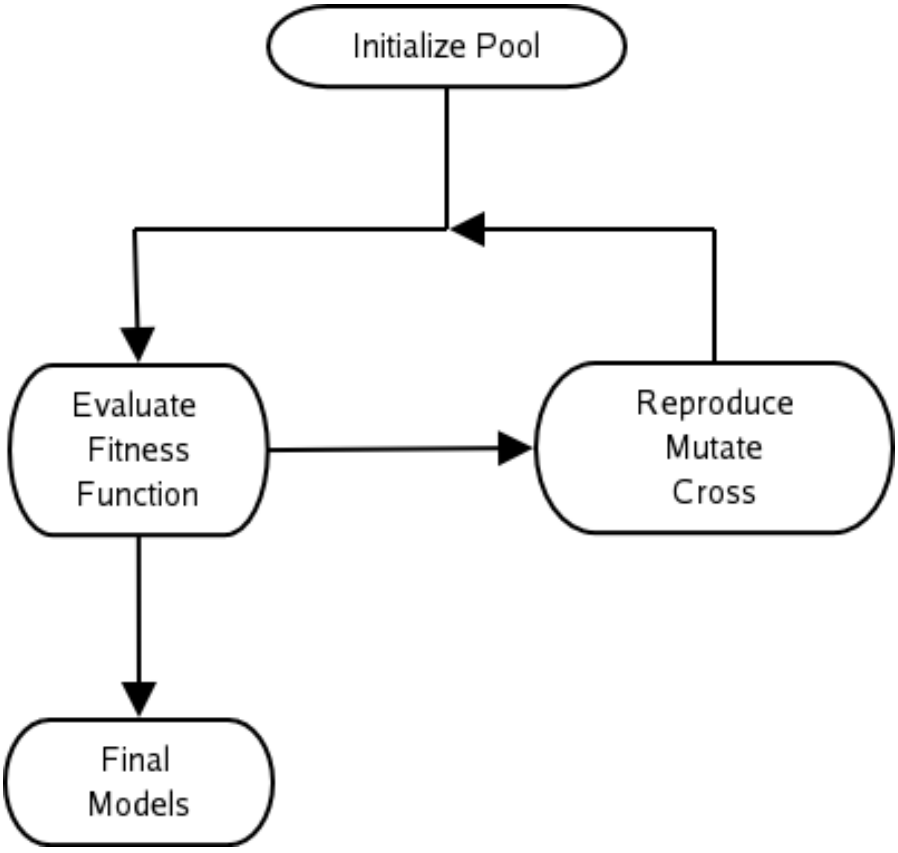


Fig. 2.11. A flow chart describing the working of a genetic algorithm

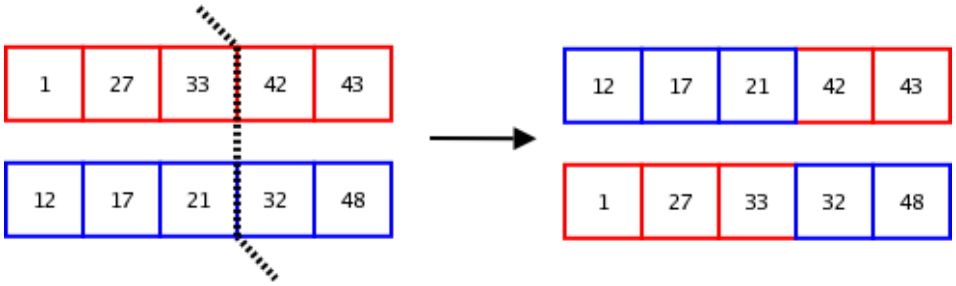


Fig. 2.12. A schematic diagram of the single point crossover operation. The grids on the left represent the parents and the grids on the right represent the children formed after crossover. The portion of the chromosomes to the left of the split point are swapped.

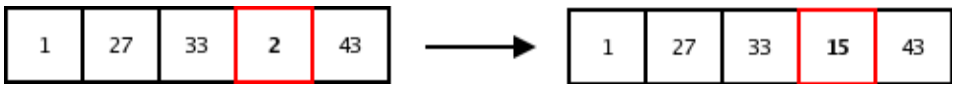


Fig. 2.13. A schematic diagram of the mutation operation

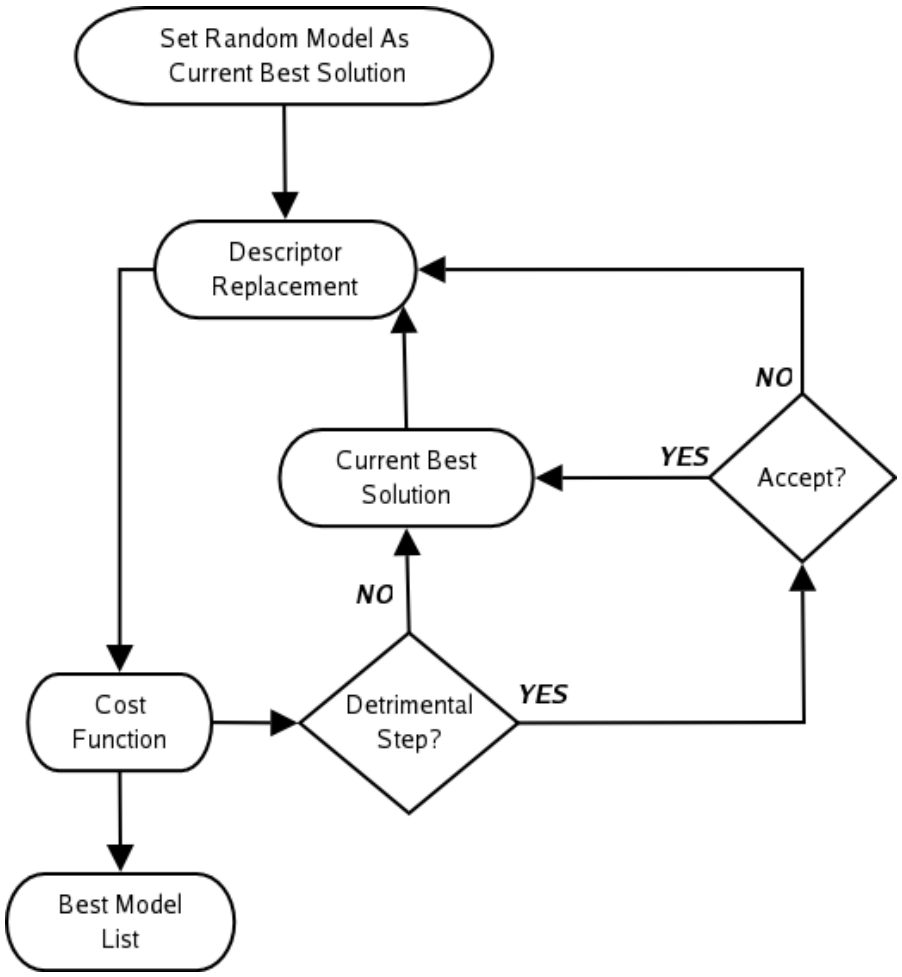


Fig. 2.14. A flow chart describing the working of a simulated annealing algorithm

References

[1] Broyden, C. The Convergence of a Class of Double-Rank Minimization Algorithms 2, The New Algorithm. *J. Inst. Math. Applic.* **1970,** *6,* 222–231.

[2] Fletcher, R. A New Approach to Variable-Metric Algorithms. *Comput. J.* **1970,** *13,* 317–322.

[3] Goldfarb, D. A Family of Variable-Metric Algorithms Derived by Variational Means. *Math. Comput.* **1970,** *24,* 23–26.

[4] Shanno, D. Conditioning of Quasi-Newton Methods for Function Minimization. *Math. Comput.* **1970,** *24,* 647–656.

[5] Nelder, J.; Mead, R. A Simplex Method for Function Minimization. *Comput. J.* **1965,** *7,* 308–315.

[6] Goldberg, D. *Genetic Algorithms in Search, Optimization & Machine Learning;* Addison-Wesley: Reading, MA, 2000.

[7] Holland, J. Genetic Algorithms. *Sci. Am.* **1992,** *267,* 66–72.

[8] Forrest, S. Genetic Algorithms: Principles of Natural Selection Applied to Computation. *Science* **1993,** *261,* 872–878.

[9] Kirkpatrick, S.; Gelatt, J. C.; Vecchi, M. Optimization by Simulated Annealing. *Science* **1983,** *220,* 671–680.

[10] Norinder, U. Single and Domain Mode Variable Selection in 3D QSAR Applications. *J. Chemom.* **1996,** *10,* 95–105.

[11] Golbraikh, A.; Tropsha, A. Beware of q^2. *J. Mol. Graph. Model.* **2002,** *20,* 269–276.

[12] Neter, J. *Applied Statistics;* Allyn and Bacon Inc.: Boston, MA, 1988.

[13] Barnett, V. Probability Plotting Methods and Order Statistics. *Appl. Stat.* **1975,** *24,* 95–108.

[14] Rousseeuw, P.; Leroy, A. *Robust Regression and Outlier Detection;* Wiley Series in Probability and Mathematical Statistics John Wiley & Sons: Hertfordshire, England, 1987.

[15] Gupta, A.; Park, S.; Lam, S. Generalized Analytic Rule Extraction for Feedforward Neural Networks. *IEEE Transactions on Knowledge and Data Engineering* **1999,** *11,* 985–991.

[16] Haykin, S. *Neural Networks;* Pearson Education: Singapore, 2001.

[17] Ripley, B. *Pattern Recognition and Neural Networks;* Cambridge University Press: Oxford, 1996.

[18] Kohonen, T. *Self Organizing Maps;* volume 30 of *Springer Series in Information Sciences* Springer: Espoo, Finland, 1994.

[19] Hornik, K.; Stinchcombe, M.; White, H. Universal Approximation of an Unknown Mapping and its Derivation Using Multilayer Feedforward Networks. *Neural Networks* **1990,** *3,* 551–556.

[20] Hornik, K.; Stinchcombe, M.; White, H. Multilayer Feedforward Networks are Universal Approximators. *Neural Networks* **1989,** *2,* 35–366.

[21] Livingstone, D.; Manallack, D. Statistics Using Neural Networks: Chance Effects. *J. Med. Chem.* **1993,** *36,* 1295–1297.

[22] Wessel, M. *Computer Assisted Development of Quantitative Structure Property Relationships and Design of Feature Selection Routines,* Thesis, Department of Chemistry, Pennsylvania State University, 1997.

[23] Kalelkar, S.; Dow, E. R.; Grimes, J.; Clapham, M.; Hu, H. Automated Analysis of Proton NMR Spectra from Combinatorial Rapid Parallel Synthesis Using Self-Organizing Maps. *J. Comb. Chem.* **2002,** *4,* 622–629.

[24] Hoehn, F.; Lindner, E.; Mayer, H. A.; Hermle, T.; Rosenstiel, W. Neural Networks Evaluating NMR Data: An Approach To Visualize Similarities and Relationships of Sol-Gel Derived Inorganic-Organic and Organometallic Hybrid Polymers. *J. Chem. Inf. Comput. Sci.* **2002,** *42,* 36–45.

[25] Chen, L.; Gasteiger, J. Knowledge Discovery in Reaction Databases: Landscaping Organic Reactions by a Self Organizing Neural Network. *J. Am. Chem. Soc.* **1997,** *119,* 4033–4042.

[26] Bienfait, B. Applications of High Reolution Self Organizing Maps to Retrosynthetic and QSAR Analysis. *J. Chem. Inf. Comput. Sci.* **1994,** *34,* 890–898.

[27] Rose, V.; Croall, I.; Macfie, H. An Application of Unsupervised Neural Network Methodology Kohonen Topology-Preserving Mapping to QSAR Analysis. *Quant. Struct.-Act. Relat.* **1991,** *10,* 6–15.

[28] Anzali, S.; Barnickel, G.; Krug, M.; Sadowski, J.; Wagener, M.; Gasteiger, J.; Polanski, J. 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. Comp. Aid. Molec. Des.* **1996,** *10,* 521–534.

[29] Chen, L.; Gasteiger, J. Knowledge Discovery in Reaction Databases: Landscaping Organic Reactions by a Self Organizing Neural Network. *J. Am. Chem. Soc.* **1997,** *119,* 4033–4042.

[30] Espinosa, G.; Arenas, A.; Giralt, F. An Integrated SOM Fuzzy ARTMAP Neural System for the Evaluation of Toxicity. *J. Chem. Inf. Comput. Sci.* **2002,** *42,* 343–359.

[31] Nicklaus, M. "`http://cactus.nci.nih.gov/DownLoad/AID2DA99.sdz`", 1999.

[32] Breiman, L. Statiscal Modeling: Two Cultures. *Statistical Science* **2001,** *16,* 199–231.

[33] Sutton, R.; Barto, A. *Reinforcement Learning;* MIT Press: Cambridge, 1999.

[34] Witten, W.; Frank, E. *Data Mining;* Morgan Kaufman: San Francisco, 2000.

[35] Christianini, N.; Shawn-Taylor, J. *An Introduction to Support Vector Machines;* Cambridge University Press: Cambridge, 2002.

[36] Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning;* Springer: New York, 2003.

[37] Kohonen, T. *Self-Organization and Associative Memory;* Springer-Verlag: Berlin, 1989.

[38] Breiman, L.; Friedman, J.; Olshen, R.; Stone, C. *Classification and Regresion Trees;* CRC Press: Boca Raton, FL, 1984.

[39] Zhang, X.; Singer, B. *Recursive Partitioning in the Health Sciences;* Springer: New York, 1999.

[40] Shaha, A. Implications Of Prognostic Factors And Risk Groups In The Management Of Differentiated Thyroid Cancer. *Laryngoscope* **2004,** *114,* 393–402.

[41] Schuurmann, G.; Aptula, A. O.; Kuhne, R.; Ebert, R. Stepwise Discrimination between Four Modes of Toxic Action of Phenols in the *Tetrahymena pyriformis* Assay. *Chem. Res. Tox.* **2003,** *16,* 974–987.

[42] Butina, D.; Gola, J. M. R. Modeling Aqueous Solubility. *J. Chem. Inf. Comput. Sci.* **2003,** *43,* 837–841.

[43] Bos, P.; Baars, B.; van Raaij, M. Risk Assessment Of Peak Exposure To Genotoxic Carcinogens: A Pragmatic Approach. *Tox. Lett.* **2004,** *151,* 43–50.

[44] Ebert, M. P. A.; Meuer, J.; Wiemer, J. C.; Schulz, H.-U.; Reymond, M. A.; Traugott, U.; Malfertheiner, P.; Rocken, C. Identification of Gastric Cancer Patients by Serum Protein Profiling. *J. Proteome Res.* **2004,** *3,* 1261–1266.

[45] Tong, W.; Xie, W.; Hong, H.; Fang, H.; Shi, L.; Perkins, R.; Petricoin, E. Using Decision Forest To Classify Prostate Cancer Samples On The Basis Of SELDI-TOF MS Data: Assessing Chance Correlation And Prediction Confidence. *Evironmental Health Perspectives* **2004,** *112,* 1622–1627.

[46] Therneau, T.; Atkinson, E. "An Introduction to Recursive Partitioning Using the RPART Routines", Technical Report, Department of Health Science Research, Mayo Clinic, Rochester, Minnesota, 1997.

[47] Venables, W.; Ripley, B. *Modern Applied Statistics with S;* Springer: New York, 2002.

[48] Breiman, L. Bagging Predictors. *Machine Learning* **1996,** *26,* 123–140.

[49] Breiman, L. Randomizing Outputs to Increase Prediction Accuracy. *Machine Learning* **2000,** *40,* 229–242.

[50] Dietterich, T. An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting and Randomization. *Machine Learning* **2000,** *40,* 139–167.

[51] Svetnik, V.; Liaw, A.; Tong, C.; Culberson, C.; Sheridan, R.; Feuston, B. Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling. *J. Chem. Inf. Comput. Sci.* **2003,** *42,* 1947–1958.

[52] Breiman, L. "Out-of-bag estimation", Technical Report, Department of Statistics, University of California, Berkeley, 1996.

[53] Guha, R. Unpublished data.

[54] Cover, T.; Hart, P. Nearest Neighbor Pattern Classification. *Proc. IEEE Trans. Inform. Theory* **1967,** *IT-11,* 21–27.

[55] Friedman, J. "Flexible Metric Nearest Neighbor Classification", Technical Report, Stanford University, 1994.

[56] Hastie, T.; Tibshirani, R. Discriminant Adaptive Nearest Neighbor Classification. *IEEE Pattern Recognition and Machine Intelligence* **1996,** *18,* 607–616.

[57] Shen, M.; Xiao, Y.; Golbraikh, A.; Gombar, V.; Tropsha, A. Development and Validation of k-Nearest Neighbor QSPR Models of Metabolic Stability of Drug Candidates. *J. Med. Chem.* **2003,** *46,* 3013–3020.

[58] Gillet, V.; Khatib, W.; Willet, P.; Gleming, P.; Green, D. Combinatorial Library Design Using a Multiobjective Genetic Algorithm. *J. Chem. Inf. Comput. Sci.* **2002,** *42,* 375–385.

[59] Le Bailly de Tilleghem, C.; Beck, B.; Boulanger, B.; Govaerts, B. A Fast Exchange Algorithm for Designing Focused Libraries in Lead Optimization. *J. Chem. Inf. Comput. Sci.* **2005,** *ASAP,* XXX.

[60] Agrafiotis, D. K.; Rassokhin, D. N. Design and Prioritization of Plates for High-Throughput Screening. *J. Chem. Inf. Comput. Sci.* **2001,** *41,* 798–805.

[61] Altman, D. G.; Andersen, P. K. Bootstrap investigation of the stability of a Cox regression model. *Statistics in Medicine* **1989,** *8,* 771–783.

[62] Tibshirani, R. Regression Shrinkage and Selection via the Lasso. *J. Royal Stat. Soc. B* **1996,** *58,* 267–288.

[63] Mantel, N. Why Stepdown Procedures in Variable Selection. *Technometrics* **1970,** *12,* 621–625.

[64] Hinnela, J.; Saxen, H.; Pettersson, F. Modeling of the Blast Furnace Burden Distribution by Evolving Neural Networks. *Ind. Eng. Chem. Res.* **2003,** *42,* 2314–2323.

[65] Moisa, T.; Ontanu, D.; Dediu, A. *Speech Synthesis Using Neural Networks Trained by an Evolutionary Algorithm;* volume 2074 of *Lecture Notes in Computer Science* Springer-Verlag GmbH: New York, 2001.

[66] Mattioni, B. E.; Jurs, P. C. Prediction of Glass Transition Temperatures from Monomer and Repeat Unit Structure Using Computational Neural Networks. *J. Chem. Inf. Comput. Sci.* **2002,** *42,* 232–240.

[67] Guha, R.; Jurs, P. C. Development of Linear, Ensemble, and Nonlinear Models for the Prediction and Interpretation of the Biological Activity of a Set of PDGFR Inhibitors. *J. Chem. Inf. Comput. Sci.* **2004,** *44,* 2179–2189.

[68] Guha, R.; Jurs, P. The Development of QSAR Models To Predict and Interpret the Biological Activity of Artemisinin Analogues. *J. Chem. Inf. Comp. Sci.* **2004,** *44,* 1440–1449.

[69] Venkatraman, V.; Dalby, A. R.; Yang, Z. R. Evaluation of Mutual Information and Genetic Programming for Feature Selection in QSAR. *J. Chem. Inf. Comput. Sci.* **2004,** *44,* 1686–1692.

[70] Wright, T.; Gillet, V. J.; Green, D. V. S.; Pickett, S. D. Optimizing the Size and Configuration of Combinatorial Libraries. *J. Chem. Inf. Comput. Sci.* **2003,** *43,* 381–390.

[71] Goicoechea, H. C.; Olivieri, A. C. Wavelength Selection for Multivariate Calibration Using a Genetic Algorithm: A Novel Initialization Strategy. *J. Chem. Inf. Comput. Sci.* **2002,** *42,* 1146–1153.

[72] Hervás, C.; Silva, M.; Serrano, J. M.; Orejuela, E. Heuristic Extraction of Rules in Pruned Artificial Neural Networks Models Used for Quantifying Highly Overlapping Chromatographic Peaks. *J. Chem. Inf. Comput. Sci.* **2004,** *44,* 1576–1584.

[73] Leardi, R. Genetic Algorithms in Chemometrics and Chemistry. *J. Chemo.* **2001,** *15,* 559–569.

[74] Levine, B.; Moores, A. Genetic Algorithm in Analytical Chemistry. *Anal. Lett.* **1999,** *32,* 433–445.

[75] Metropolis, N.; Rosenbluth, A.; Rosenbluth, M.; Teller, A.; Teller, E. Equation of State Calculations by Fast Computing Machines. *J. Chem. Phys.* **1953,** *21,* 1087–1092.

[76] Bohachevsky, I.; Johnson, M.; Stein, M. Generalized Simulated Annealing for Function Optimization. *Technometrics* **1986,** *28,* 209–217.

[77] Sutter, J.; Kalivas, J. Comparison of Forward Selection, Backward Elimination and Generalized Simulated Annealing for Variable Selection. *Microchemical J.* **1993,** *47,* 60–66.

Chapter 3

QSAR Methodology and ADAPT

The previous chapter described a number of modeling techniques with which QSAR models can be built. Based on the nature of the method used, QSAR models are classified as linear or nonlinear. However, the modeling process does not simply consist of passing data through an algorithm. As described in Chapter 1, the fact that we cannot directly calculate physical properties or biological activities requires us to take an indirect route. As a result, QSAR modeling is a stepwise process consisting of five main steps:

1. Structure entry and optimization
2. Descriptor calculations
3. Objective and subjective feature selection
4. Model development
5. Prediction

This is a very broad overview and certain steps, such as set generation and interpretation have been skipped over, though these will be discussed in more detail in subsequent chapters. Another important step in the QSAR model development process is the consideration of the validity of models. This aspect has many facets and one of them involves deciding as to whether a model will be applicable to a set of unseen query compounds. This topic is discussed in more detail in Chapter 5.

The utility of a QSAR model depends on its intended future use. If a model is to be used as a screen in a high throughput pipeline, the predictive ability of the model is paramount. In these cases the high predictive ability of CNN's and random forests make models based on these methods attractive. If a QSAR model is to be used as a guide to possible modifications of molecules to improve their activities, the interpretability of the model assumes a major role. In this case the simple PLS interpretation scheme that can be applied to linear models make them good candidates for QSAR modeling inspite of their lower predictive performance for biological properties. Chapters 8 and 9 describe

two approaches to providing interpretations for CNN models. As a result, one gets the best of both worlds - high predictive ability and a measure of interpretability.

This chapter discusses in detail the various steps in the model development process and is oriented towards the use of the ADAPT[1,2] software package for 2-D QSAR model development and testing.

3.1 Structure Entry and Optimization

The model building process using ADAPT begins with the creation of a work area which initializes the various files and related storage requirements for a QSAR study. The molecules to be used in the study can be available as 2-D or 3-D structures. In general, the data are in the former format and as a result, 3-D structures are required. 3-D structures are usually generated using Hyperchem. The resultant structures are crudely optimized using a molecular mechanics method within Hyperchem. Once the dataset has been converted to 3-D structures, they are rigorously optimized with Mopac 7.01. This program employs a semi-empirical method using the PM3[3,4] Hamiltonian. This Hamiltonian is reported to be well suited to the purpose of geometry optimization. Since some molecular descriptors also require information about the electronic environment of the molecule, the molecules are also optimized for electronic properties. In this case the AM1[5–7] Hamiltonian is used. Once the molecules have been optimized for geometry and electronics, they are stored in the ADAPT work area.

3.2 Molecular Descriptor Calculations

As mentioned before, the fundamental assumption of QSAR modeling is that molecular structure can be correlated to physical or biological properties. Thus the fundamental requirement is some method to encode various structural features in a molecule. Molecular descriptors fulfill this requirement. Descriptors are (in general) numerical representations of specific molecular features. Such features can range from very simple ones such as the number of carbons or number of halogen atoms to more complex and abstract features such as graph invariants of the molecular graph or the information content of a molecule as characterized by entropy. Several packages are available to calculate a wide variety of descriptors. Examples include Dragon,[8] JOELib[9] and ADAPT. Owing to the large variety of descriptors that can be calculated we restrict ourselves to a discussion of the main types of descriptors that are calculated by ADAPT and refer the reader to the literature[10] for additional information. The descriptors calculated by ADAPT can be

classified into four types: geometrical, topological, electronic and hybrid. The following sections describe the nature of each descriptor class in detail.

3.2.1 Geometric Descriptors

Geometric descriptors characterize the shape and extent of the molecule in terms of its 3-D coordinates. As a result accurate coordinates are required and so the structure must be geometry optimized before these descriptors can be calculated. Examples include moment of inertia,[11] molecular surface area and volumes,[12] and shadow descriptors.[13,14] The surface area and volume descriptors are usually used in combination with atomic properties (such as partial charges or hydrophobicities) and are useful in characterizing the distribution of these properties. The shadow area descriptors align the first two moments of inertia of the molecule along the X and Y axes and then calculate the area of the projection of the molecule on the XY, XZ and YZ planes. In general these types of descriptors capture features related to molecular size and shape and thus are generally physically interpretable. The drawback to these descriptors is that they require accurate molecular geometries and thus for large sets of molecules the optimization step can become time consuming. Furthermore, the ADAPT implementation of these descriptors do not take conformational features into account and work with the lowest energy conformer. Situations where conformation details play an important role (such as ligand binding) will not be accurately characterized by these descriptors.

3.2.2 Topological

As the name suggests, topological descriptors consider the topology of a molecule. That is, in the most general case, only the connections between the atoms in a hydrogen suppressed molecule, effectively converting it into a mathematical graph. Certain topological descriptors consider the type or certain properties of atoms involved in the connections as weights. Topological descriptors characterize features such as path lengths and connectivity. Examples include connectivity indices,[15–17] distance edge vectors[18] and eccentricity indices.[19] Since topological descriptors consider the molecule as a mathematical graph, a number of these descriptors are simply various graph invariants or other functions of the molecular graph. Examples include eigenvalues of the adjacency matrix and descriptors based on the molecular influence matrix[20] etc. However, there are other

descriptors that consider features such as paths and vertex degrees. Topological descriptors are able to provide a more detailed description of molecular shape features such as branching and crowdedness.

As an example consider the calculation of some connectivity indices. These were first described by Randic[21] and later extended by Kier and Hall[15] with the assumption that

> there resides in the structural formula sufficient information so that an index, based upon non-empirical counts of atoms, can be calculated

Numerous connectivity indices have been defined. A well-studied example is the χ index. These indices consider the vertex degree of each atom in various subgraphs of the molecular graph. Thus the ${}^1\chi$ index is defined as

$$ {}^1\chi = \sum_{i \neq j} \frac{1}{\sqrt{\delta_i \delta_j}} \tag{3.1} $$

where δ_i and δ_j are the vertex degrees of two bonded atoms, i and j. However, the ${}^1\chi$ descriptor is simplistic since it only considers atoms connected by a single bond. To characterize a molecular structure on a larger scale, extended versions of the χ descriptor were defined. Thus the second order connectivity index, ${}^2\chi$, is computed by *dissecting*[16] a structure into 2-bond (i.e., 3 atoms) fragments. The value of the descriptor is then calculated by

$$ {}^2\chi = \sum_{i \neq j \neq k} \frac{1}{\sqrt{\delta_i \delta_j \delta_k}} \tag{3.2} $$

where δ_i, δ_j and δ_k are the vertex degrees of the atoms in a given fragment. Higher order χ indices can be calculated in a similar manner. An important feature of these descriptors is that they restrict themselves to linear paths. However, structures exhibit branching and cyclic paths and the connectivity indices were extended to take these features into account, resulting in the ${}^m\chi_f$ descriptors where m denotes the number of edges in a fragment and f denotes the type of fragment that may be p (path), c (cluster), pc (path–cluster) or ch (chain). The structures of these fragments are summarized in Fig. 3.1.

It should be noted that the original definition only considered saturated carbon atoms. To take into account unsaturation and heteroatoms the δ value for an atom was modified such that

$$ \delta^v = Z^v - h \tag{3.3} $$

where Z^v is the number of valence electrons and h is the number of hydrogens. Eq. 3.3 was also extended to take into account core electrons for higher row elements. The use of the δ^v values lead to the calculation of the valence corrected χ indices denoted by ${}^m\chi^v_f$, where m and f have been previously defined.

Fig. 3.2 shows an example of how a molecular structure is decomposed into a variety of fragments. The original molecule (1-methyl 3-ethyl benzene) is decomposed into a 6^{th} order chain (top), two 3^{rd} order clusters (right) and three 4^{th} order path–clusters (bottom). The numbers in the central structure correspond to the vertex degree for each atom. Thus the value of ${}^6\chi_{ch}$ would be obtained by

$$\begin{aligned} {}^6\chi_{ch} &= 1 \cdot \frac{1}{\sqrt{2 \times 2 \times 2 \times 2 \times 2 \times 2}} \\ &= 0.125 \end{aligned}$$

Similarly the fourth order path–cluster χ index, ${}^4\chi_{pc}$ would be calculated as

$$\begin{aligned} {}^4\chi_{pc} &= 3 \cdot \frac{1}{\sqrt{(1 \times 3 \times 2 \times 2 \times 2)}} \\ &= 0.612 \end{aligned}$$

Finally the value of the third order cluster χ index, ${}^3\chi_c$ would be calculated as

$$\begin{aligned} {}^3\chi_c &= 2 \cdot \frac{1}{\sqrt{(1 \times 3 \times 2 \times 2)}} \\ &= 0.576 \end{aligned}$$

Another type of topological descriptors are the BCUT's developed by Pearlman et al.[22] These descriptors are based on the Burden matrix[23] which is an adjacency matrix in which the non-diagonal elements are weighted based on the nature of the connectivity of the atoms involved. Thus for a molecule with n atoms and an $n \times n$ adjacency matrix, A, the Burden matrix, B is defined as

$$B_{ij} = \begin{cases} \pi \times 0.1 & \text{if } i \neq j \text{ and } A_{ij} = 1 \\ 0.001 & \text{if } i \neq j \text{ and } A_{ij} = 0 \\ Z & \text{if } i = j \end{cases}$$

where π represents the conventional bond order and Z represents the atomic number. Furthermore, all off-diagonal elements of B are augmented by 0.01.

The fundamental modification made by Pearlman was to place atomic properties along the diagonal of the Burden matrix. This leads to a variety of weighted Burden matrices where the weights include atomic weight, polarizability, electronegativity and hydrogen bonding ability. The actual descriptors are obtained by performing an eigenvalue decomposition of the Burden matrix and taking the lowest and highest eigenvalues. It has been shown that the extreme eigenvalues of the Burden matrix encode global information[24] regarding the molecule. Thus by combining atomic properties with the Burden matrix, the resultant eigenvalues encode global structure-property characteristics of a molecule, leading to BCUT descriptors being termed *holistic*. The holistic nature of these descriptors have led to their frequent use in studies of chemical diversity,[25–27] library design[28,29] and hit selection in high throughput screens.[30–32]

Topological descriptors have been widely used and have been shown to be very useful in building predictive models. Since they only require connectivity information for a molecule, the process of drawing and optimization of 3-D structures can be avoided. This results in the rapid calculation of this class of descriptors. The downside to topological descriptors is the lack of physical interpretability. Many of these types of descriptors are quite abstract in nature and though a number of reports have described correlations between certain descriptors and physical properties,[33–35] these are not easily generalized.

3.2.3 Electronic Descriptors

Electronic descriptors consider various features of the molecules' electronic environment. These include the HOMO and LUMO energies, electronegativity and various atom-centered partial charge descriptors. The ADAPT system is able to calculate atomic charges using empirical data to fit the dipole moments of molecules[36,37] or by using pK_a values.[38] These approaches are attractive since they do not require any optimization to be carried out and only consider molecular connectivity. The downside of these methods is that they are based on a predefined set of parameters and thus will not necessarily be accurate for a number of molecules. An alternative approach is to use an *ab initio* or semi-empirical technique to calculate charges. Owing to the time intensive nature of the former method, the semi-empirical approach is preferred and ADAPT is able to import partial charges calculated using the AM1[5] Hamiltonian with the MOPAC package. Though ADAPT focuses on charge based descriptors derived semi-empirically, there are a number of studies describing the development and application of *ab inito* quantum mechanical molecular descriptors[39–42] that calculate properties such as electron density,

Fukui functions and so on. Owing to the computationally intensive nature of the calculation of these descriptors, a method has been developed that works with *atomic fragment values*, termed Transferable Atom Equivalent's (TAE)[43] which allow for the calculation of quantum mechanical descriptors for whole molecules using atomic fragment values. Together with a wavelet based encoding and a hybrid shape-property descriptor, TAE's have been used to build predictive QSAR models of high quality.[44]

3.2.4 Hybrid Descriptors

Hybrid descriptors are generally combinations of electronic or topological descriptors and geometric descriptors and in general characterize the distribution of a molecular feature over the whole molecule. Examples include the charged partial surface area (CPSA),[45] hydrophobic surface area (HPSA)[46] and hydrogen bonding[47,48] descriptors. The important characteristic is that they provide localized information regarding molecular features. Thus, in the case of the HPSA descriptors, one is able to obtain specific values of the hydrophobicity for different regions of the molecule as well as a global value for the whole molecule. For example, consider Fig. 3.3. In the upper figure, the atomwise hydrophobicity values are displayed. These hydrophobicity values are then color coded and mapped to the molecular surface to provide a visual representation of the information in the lower figure. The atomwise hydrophobicity values can be combined with surface area information for the individual atoms, as shown in Table 3.2, to obtain a wide variety of descriptors (25 in the ADAPT implementation). Examples include the atomic constant weighted hydrophobic and hydrophilic surface areas, total hydrophobic constant weighted hydrophobic surface area and the relative hydrophobicity. The functional forms of these descriptors are given below.

$$\begin{aligned}
\text{PPHS-2} &= \sum(+SA_i)(+\log P_i) \\
\text{PNHS-2} &= \sum(-SA_i)(-\log P_i) \\
\text{THWS} &= \sum(\log P_i)(SA_i) \\
\text{RPH-1} &= \frac{\text{Most hydrophobic atom constant}}{\sum \log P_i}
\end{aligned}$$

where SA_i is the surface area for the i^{th} atom and $\log P_i$ is the hydrophobic constant for the i^{th} atom and the $+$ and $-$ symbols indicate a hydrophobic or hydrophilic atom respectively.

The CPSA descriptors are similar in concept to the HPSA descriptors. In this case, the surface area values are combined with partial charges leading to 25 descriptors. In addition, a number of CPSA descriptors specific to certain atoms (such as N and O) are also calculated. These descriptors are similar in concept to the Polar Surface Area (PSA) descriptors[49,50] which have been shown to be very useful in studies of intestinal absorption[51] and blood brain barrier crossing.[52] The development of the Topological Polar Surface Area (TPSA) method by Ertl et al.[53] allows the rapid evaluation of polar surface areas using only connectivity information (SMILES strings) and a library of fragment contributions.

By combining molecular surfaces with atomic properties, these descriptors are useful both in 2-D as well as 3-D QSAR methods. In addition, surface-property descriptor types usually have simple physical interpretations and have been shown to be quite information rich.[54,55]

3.3 QSAR Set Generation

An important step in the modeling process is the creation of QSAR sets. Given a dataset of molecules, three mutually exclusive sets are created. The first, termed the training set, is used during the model building process. The learning algorithm used to build the model uses this set to characterize the dataset based on features present in the training set. The next set is the cross-validation set and is used in the case of CNN models. This set is used periodically during the training of the CNN and allows for the monitoring of the error rate during training. In the case of linear models the training set and cross-validation set are combined together. Finally the prediction set is a subset of the dataset that is not used at all during model building. Its purpose is to validate the final model and ascertain its predictive ability. These three sets are collectively termed QSAR sets.

The most common technique to generate these sets is random selection. The technique used in ADAPT is termed activity-weighted binning. In this procedure the dataset is binned based on activity values and then molecules are selected based on a probability, weighted by the bin populations.

An important point to note is that the learning algorithms are attempting to capture features of the dataset from a smaller subset of the overall dataset. When a model has been built it is tested on a another subset of the dataset. Clearly if the features that are present in the training set are not sufficiently represented in the prediction set, the

models' predictive ability will be poor. Thus we must consider the idea of *representative* QSAR sets. In general, the QSAR sets should be created such that the various features present in the dataset should be proportionally represented in each individual QSAR set. One approach to this problem is to classify the dataset based on features described by a set of global molecular descriptors. The aim of such an approach is that these descriptors should be able to represent the main features of the dataset. The QSAR sets are then created such that molecules from the classes are represented in the same proportion that was found in the overall dataset. This approach is discussed in more detail in Chapter 4 where a Kohonen self organizing map is used to classify the dataset and subsequently create the QSAR sets. Alternative methods include the use of statistical molecular,[56] D-Optimal[57] or Kennard-Stone[58] design methods.

3.4 Feature Selection

Though only four types of descriptors has been mentioned above, these classes account for the nearly 300 descriptors calculated by ADAPT. Other programs such as DRAGON are able to evaluate nearly 1200 descriptors covering a wide variety of descriptor classes. It is apparent that in such a large descriptor pool a number of descriptors will be highly correlated with other descriptors or else may have the same value for all the molecules (such as number of aromatic rings, when the dataset has no aromatic rings) and will thus contain no relevant information. Thus before descriptors can be used for model building, the original descriptor pool must be reduced in size by selecting only feature rich and relevant descriptors. This selection step is termed objective feature selection. Once a reduced pool of descriptors has been created, suitable subsets of the descriptors must be selected to build QSAR models with. This step is termed subjective feature selection.

3.4.1 Objective Feature Selection

The original descriptor pool obtained from the evaluation of all available descriptors is reduced in size by two main methods. First, an identical test is carried out. This procedure removes descriptors that have a constant value for a user specified percentage of the dataset. In general the percentage ranges from 80% to 90%. The next step is to calculate the correlation coefficient between all the pairs of descriptors. If a pair of descriptors exhibit a R^2 value greater than or equal to a user specified cutoff, one member of the descriptor pair is discarded. Which pair is discarded is in general random; however,

if one member of the pair is a topological descriptor, it is kept in preference to the other member. The reason for this behavior is that topological descriptors generally provide a global description of a molecule. Though the same is true of geometric descriptors, the larger number of topological descriptors available warrant their preferred inclusion in the reduced descriptor pool. Another technique that is used to create a reduced pool of descriptors is *vector space descriptor analysis*, which is based on the Gram-Schmidt orthogonalization procedure.[59] This technique considers descriptors as vectors and attempts to create a descriptor pool as a spanning linear vector space. Essentially, it starts by placing the descriptor that is most correlated to the dependent variable in the reduced pool. The next step is to find the descriptor from the original pool that is most orthogonal to the current descriptor. This step is repeated, each time selecting the descriptor from the original pool that is most orthogonal to the subspace spanned by the previously selected descriptors. The procedure is repeated until the number of descriptors in the reduced pool reaches a user-defined limit.

By varying the cutoffs for the identical and correlation tests and the size limit for the vector space technique, the size of the reduced descriptor pool can be varied by the user. In general a rule of thumb is used to decide on the size of the final reduced pool and is given by

$$\frac{n_{\text{mol}}}{n_{\text{reduced}}} = .6 \tag{3.4}$$

where n_{mol} is the number of molecules in the dataset and n_{reduced} is the number of descriptors in the reduced pool. This rule is derived from work carried out by Topliss et al.,[60] which quantitatively measured the relationship among the number of variables, the number of observations and the probability of chance correlations in linear regression models based on simulated data. The value of 0.6 represents a tradeoff between the numbers of variables and observations to minimize the probability of chance correlations.

3.4.2 Subjective Feature Selection

This stage of feature selection refers to methods by which descriptor subsets are selected from the reduced descriptor pool for model building purposes. The problem is combinatorial in nature; for reduced pools of moderate size a brute force approach to subset selection is unwieldy and for larger pools, computationally unfeasible. As a result, for reduced pools containing more than 20 descriptors stochastic search methods are preferred. Such methods include genetic algorithms[61,62] (GA), simulated annealing[63]

(SA), particle swarms[64] and ant colony algorithms.[65] ADAPT implements GA and SA methods for subjective feature selection.

The details of the GA and SA methods have been described in Chapter 2. The implementation of the GA in ADAPT involves the use of an objective function, which depends on the type of model being built. In the case of linear models the genetic algorithm is coupled with a linear regression routine. The fitness of a given descriptor subset is a function of the root mean square error (RMSE) of the model based on that subset. Another constraint that is sometimes applied is that models with values of the t-statistic less than 4.0 are rejected. However, it has been seen in practice that this sometimes leads to the rejection of models that have good predictive ability. Hence, this constraint is not strictly applied and models with lower values of the t-statistic are considered. In the case of descriptor subset selection for neural network models, the objective function is a 3-layer, fully-connected, feed-forward CNN as described in the previous chapter. The fitness for a given descriptor subset is defined using a cost function based on the RMS errors of the training and cross validation sets used in the CNN model. This cost function is defined as

$$\text{Cost} = \text{RMSE}_{TSET} + 0.5 \times |\text{RMSE}_{TSET} - \text{RMSE}_{CVSET}| \tag{3.5}$$

where RMSE_{TSET} and RMSE_{CVSET} are the RMSE values for the training and cross-validation sets, respectively. This cost function is designed to take into account model performance based on the training set as well as the extent of overfitting. As described in Chapter 2, care must be taken to prevent overfitting in a neural network model. This is controlled by the use of the cross-validation set. By considering the RMSE for the cross-validation set, the cost function penalizes models that cannot generalize as exhibited by having poor cross-validation performance. The constant factor of 0.5 is an empirically chosen value and has been observed to provide a balance between the RMSE values of the training and cross-validation sets.

In the case of the simulated annealing algorithm, the above discussion holds, except that the *energy* of a given configuration (i.e., descriptor subset) is now given by the RMSE (for linear models) or the value of the cost function (CNN models).

It should be noted that in the case of CNN models, the use of the genetic or simulated annealing algorithms results in models having optimal descriptor subsets for the specified architecture. To fully investigate the performance of a selected descriptor

subset, a variety of CNN architectures must be considered. This is carried out by developing models with the same set of input descriptors but varying architectures (i.e., varying numbers of hidden layer neurons). The final model for a given descriptor subset is that which exhibits the lowest cost function.

3.5 Model Development

Once we have calculated the descriptors, reduced the original pool to a more manageable size and then selected a number of optimal descriptor subsets we can then proceed to build a set of models and choose the best one. The ADAPT methodology for model development involves three steps. First a set of linear models are developed using the top five to ten descriptor subsets selected by the GA or SA, coupled to the linear regression routine as the cost function. These models are termed Type I models. The best model is selected based on R^2 and RMSE value. In many cases, such as for biological properties, a simple linear relationship will not result in good predictive performance. Thus, the next step is to investigate whether the selected descriptor subset will show enhanced perfomance when used in a nonlinear relationship. Thus, we use the descriptor subset from the linear model and build a nonlinear CNN model. For the given descriptor subset (i.e., input neurons) a number of CNN models are developed by varying the number of hidden neurons, subject to the constraint specified by Eq. 2.5. Out of this set of models the final model is the one that exhibits the lowest cost function defined by Eq. 3.5. This model is termed a Type II model. The problem with this type of model is that it uses a descriptor subset that was selected by the GA (or SA), based on its performance in a linear model. That is, the descriptor subset was optimal for linear models but not necessarily for nonlinear models. As a result, the final step of model building consists of using the GA (or SA) coupled to the CNN routine to search for descriptor subsets that show good performance in CNN models. Once a number of descriptor subsets have been obtained, the final architecture is obtained as described above. Nonlinear models that are obtained by linking the feature selection routines to a nonlinear cost function are termed Type III models. The model development procedure described here is summarized graphically in Fig. 3.4. The result of this procedure is to create a set of linear and nonlinear models. In many cases, both types of models can be used in combination to investigate different aspects of the structure-property relationship being modeled and in other cases one type of model may be sufficient to

understand the trends present in the dataset as well as provide good predictive ability for new observations.

3.6 Prediction, Validation and Interpretation

After a QSAR model has been developed the next step is to investigate its predictive ability. The simplest method is to test the model on a subset of the dataset that has not been used during the model development process (the prediction set). The statistics obtained from the results of the prediction set can give us some indication of the model's predictive ability. The most common statistics for linear models are R^2 and RMSE, though the former is not always a very reliable indicator of the goodness of fit as shown in Fig. 3.5. The figure plots the predicted versus observed values obtained from a linear regression model based on a simulated dataset. The dataset consisted of two well-seperated Gaussian clusters. Clearly, the relationship between the independent variables and the dependent variable is not linear. However, the R^2 value of 0.91 misleadingly indicates that the regression model fits the data well.

Another aspect closely related to predictive ability is *generalizability*. The main problem with the use of a single prediction set as a test of a model's predictive ability is that it is a limited indicator of the model's ability to handle new data. Generalizability is a more general term than predictive ability and essentially describes how the model behaves when faced with new data. The question of generalizability arises owing to the fact that a testing methodology based on a subset of the original dataset is inherently biased since the prediction set will, to some extent, share distribution characteristics of the training set. Obviously this may not always be true and is dependent on the manner in which the training and prediction sets are generated. But in general one can assume that new datasets will share the characteristics of the data to differing extents. Clearly a new dataset that differs greatly from the training data (say a dataset of linear molecules versus a dataset of cyclic molecules) will not give rise to good predictions from the model. On the other hand a new dataset containing molecules that are similar to the training data can be expected to lead to good predictions.

How can we measure generalizability? The answer to this question is not clear cut. One possible indicator of model generalizability is the relative performance of the model on cross-validation and prediction sets. This possibility is discussed in more detail in Chapter 4. An important point to note regarding this approach is that this requires the use of a cross-validation set and consequently cannot be applied directly to linear

models built using multiple linear regression. An alternative approach to the question of generalizability, alluded to above, is to try and quantify how well a new dataset will be predicted by a model. Essentially, this method tries to link some aspect of model quality to the structures of the molecules being considered. One approach is to link model quality to some similarity measure between the training dataset and a new dataset. Yet another possibility is to predict the performance of a model on a new dataset directly, using information from the model and the new structures. These approaches are discussed in more detail in Chapter 5.

Validation of a QSAR model is very similar in nature to the ideas discussed above. However, whereas the above discussion focuses on validation of a final QSAR model, validation also plays an important role during model development and is generally termed cross-validation. More specifically, algorithms such as neural networks and random forests all benefit from a validation mechanism during model development. In the case of a neural network, cross-validation is required to prevent over-training as described in Section 2.2.1. Similarly, the random forest algorithm uses a built-in cross-validation scheme to provide an internal measure of accuracy.

The ADAPT CNN methodology uses two forms of validation. One option is to use a fixed cross-validation set through all validation iterations. The second option is to use a leave n% out validation scheme. The latter method works by randomly selecting n% of the training set at each validation iteration and evaluating a cross-validation RMSE. A wrapper is also available which carries out a *round robin* leave n% out validation scheme (though this is probably more correctly termed as an ensemble method). In contrast to the above method it generates multiple training, cross-validation and prediction sets such that each member of the dataset is present in one of the prediction sets (and correspondingly one of the cross-validation sets). Though this is more rigorous than than a neural network algorithm with a fixed cross-validation set it is probably not as useful as the leave n% out method using randomly selected cross-validation sets. The reasons are twofold. First, the procedure whereby each member of the dataset is predicted once is extremely time consuming. Second, this procedure results in an ensemble of neural network models (for each training, cross-validation and prediction set combination) rather than a single model. One possible justification for ignoring the latter drawback is that neural network models are in general not considered interpretable and are usually developed for their predictive ability. Thus the fact that an ensemble of models is generated rather than a single model may be justified to some extent if the predictive ability of the ensemble is significantly better than the single model. However,

as noted by Agrafiotis et al.,[66] the "benefits of aggregation methods are clear but not overwhelming."

Validation, in the sense of neural networks and random forests, is not directly applicable to the case of linear model's developed using multiple linear regression. However, one method that can be used to gain an idea of a linear model's predictive ability is to use a leave-one-out (LOO) procedure resulting in a prediction for each member of the dataset. This method results in a cross-validated R^2, usually denoted by Q^2. However, the utility of this statistic is debatable and numerous discussions are available in the literature.[67–71]

An important component of the validation process is testing for chance correlations. That is, we would like to know whether the results generated by the model were due to chance correlations rather than the model actually capturing a specific structure activity relationship (SAR). This is important in the context of the ADAPT methodology as the algorithms used during subjective feature selection are stochastic in nature. Thus it is possible that the results of a model developed on the basis of a descriptor subset selected by the GA or SA are simply due to luck rather than an any real relationship between the dependent variable and the independent variables. The simplest strategy to test for chance correlations is to scramble the dependent variable and estimate R^2 and RMSE values for the model using the scrambled dependent variable. Since the fundamental assumption of QSAR modeling is that descriptor values correlate with the observed activity (or property) one would expect that the R^2 for the scrambled dependent variable would decrease and that the RMSE would increase. Graphically, a plot of the observed versus predicted property should appear random as illustrated in Fig. 3.6. If the results of the scrambled runs are similar to those produced by a model using the true dependent variable then one must conclude that the model has not captured a real structure activity relationship. Topliss et al.[60] discuss the role of chance correlations in the context of linear regression and their simulations provide a guide to the probability of observing a given value of R^2 (for the case of random variables). The simulation only considered a small set of possible variable combinations and thus is not an exhaustive study. However it does indicate the importance of checking for chance correlations. The method of scrambling the dependent variable can be applied to both linear and nonlinear models. In either case this technique tests the resultant model for chance correlations.

Another possibility is to test the feature selection algorithms themselves for chance correlations. That is, are the best descriptor subsets arising due to chance or are they

really minima in the descriptor space searched by the GA or SA? A simple way to investigate this type of chance correlations is to evaluate the statistics of models built from randomly selected descriptors. Similar results as described above would be expected. In this case the difference should not be as large since the descriptors will still be correlated to the dependent variable, but owing to random selection the descriptor combinations may not be optimal and hence should result in poorer statistics compared to a model built with an optimal subset of descriptors.

At this point we have in hand a validated model with (it is hoped) good predictive ability. The important feature of the model is that it should have incorporated one or more structure activity relationships. The final task of a QSAR modeling methodology is to interpret the model to describe these relationships. The ADAPT methodology leads to both linear and non-linear models and currently both types of models can be interpreted. The interpretation of linear models utilizes the PLS technique described by Stanton.[72] Its ability to dissect the effects of individual descriptors on the dataset allows a very detailed description of any structure activity relationship captured by the model. A brief description of the PLS technique is provided below.

The first requirement of this technique is to have a statistically valid linear regression model - generally characterized by high absolute values of individual t-statistics and a high value of the overall F-statistic. The next step is to build a PLS model using the selected descriptors. An important observation at this point is that the PLS algorithm employed in this work used a leave-one-out cross-validation scheme to determine the optimal number of PLS components. If the optimal number indicated by cross-validation does not equal the number of descriptors, the initial linear model was overfit and thus cannot be usefully analyzed by the PLS technique.[72] Given a validated model we extract the X and Y scores and the X weights from the PLS analysis. The X weights give an $m \times n$ matrix, where m is the number of descriptors and n is the number of PLS components, which are simply linear combinations of the descriptors used in the original original linear models. Essentially each column can be interpreted as the contributions of individual descriptors to a given component. The X and Y scores will be also be matrices with the PLS components in the columns and the observations in the rows. The Y score vector for a given component is analogous to a predicted value made by the original linear model, except that now it models the transformed variable denoted by the X score vector. For each component we create scoreplots by plotting the X score vector against the Y score vector. The next stage involves a simultaneous analysis of the components and their corresponding scoreplots. Ideally we would see that there are one

or two descriptors in each component that have high weight values - indicating that they are the main contributors to the component. We start with the first PLS component and obtain the most weighted descriptor. We then consider the score plot for that component. Compounds in the upper right and lower left are properly predicted whereas compounds lying in the other quadrants are either over-predicted (upper left) or under-predicted (lower right). Compounds that are correctly predicted as active will tend to lie in the upper right quadrant and those that are correctly predicted as inactive will occupy the lower left quadrant of the scoreplot. One can thus conclude that compounds with high values of the most weighted descriptor (assuming the weight is positive) will be more active than compounds with low values. This argument is reversed if the weight for the descriptor is negative. The under- or over-predicted compounds are not explained by the current component. Thus we must consider the next component and its most weighted descriptor. One would expect that compounds that were poorly predicted by the first component will be well predicted by the second one and the most weighted descriptor for this component will be able to account for the good predictions. Once again, for the poorly predicted cases, we move to the next component and proceed as before.

At the end of this procedure the role of the individual descriptors in determining activity (or lack of it) will have been extracted from the model. In the words of Stanton,[73] "it's like reading a book". This technique has been used in the interpretation of biological activity of artemisinin analogous[74] and the inhibitory activity of PDGFR inhibitors.[55]

In the case of a neural network model two forms of interpretation can be generated. First, a measure of the importance of the input descriptors can be generated using a technique analogous to the measure of variable importance in random forests.[75,76] In addition, we can also provide a more detailed interpretation of a CNN model based on a method inspired by the PLS technique described above. The development of the CNN interpretation methodologies and examples of applications are described in Chapters 8 and 9.

3.7 Conclusions

The development of QSAR models proceeds in a stepwise fashion as described in this chapter. The first step is the entry of the molecular structures and optimizations for geometry and electronic properties. Next, molecular descriptors are calculated for the dataset and objective feature selection is carried out to reduce the number of descriptors to a manageable pool. The next step is to select subsets of descriptors to build models. As

has been shown, descriptor selection and model building are interlinked, using stochastic algorithms to search for descriptor subsets that lead to low cost (in terms of RMS error for linear and cost function for nonlinear) models.

The model building process generally proceeds in three stages. In the first stage a set of linear models are built. In the second stage, the descriptor subsets used in the best linear models are then used to build neural network models, the assumption being, that, if a nonlinear structure-activity relationship is present, the CNN should be able to better capture it. In the third stage, the GA or SA feature selection method is coupled with the CNN routine to search for descriptor subsets that perform optimally in a nonlinear model. In both the second and first phases, the final architecture of the CNN model, for a selected descriptor subset, is decided upon by rigorously investigating all possible architectures subject to the constraint on the number of adjustable parameters.

Finally, after a number of models have been generated, they are validated and then investigated for predictive and interpretive ability. The former is usually good for the selected models. The resultant models can then be interpreted. Depending on the type of model different degrees of interpretation are possible. Linear regression and CNN models can be interpreted in a detailed manner. In addition, broad measures of descriptor importance can also be obtained for CNN models and ensemble models (such as random forest models) though such interpretations are necessarily not as informative.

The following chapters discuss applications of the QSAR methodology described here as well as investigations of specific steps in the QSAR methodology.

Table 3.1: A list of the descriptors and their associated class available in ADAPT

Type	Name	Function	Reference
Topological	DKAPPA	κ shape indices	77–79
	DMALP	All self avoiding paths of length upto the longest path in the structure	80, 81
	DMCHI	χ molecular connectivity indices	15–17
	DMCON	Molecular connectivity indices, similar to DMCHI but corrects for heteroatoms in rings and aromatic rings	82, 83
	DMFRAG	Counts for a variety of substructures	
	DMWP	Weighted paths based on Randic's molecular ID	21
	DEDGE	Molecular distance edge descriptor, λ	18
	CTYPES	Hybridization of carbon atoms based on connectivity only	
	DESTAT	Electrotopological state	84, 85
	DPEND	Superpendentic index	86
Geometric	DSYM	Structural symmetry index, equal to ratio of the number of unique atoms to the total number of atoms in a hydrogen suppressed structure	
	ECCEN	Eccentric connectivity index	19
	DMOMI	Moments of inertia along X, Y and Z axes	11
	SAVOL	Molecular surface area and volume	12
	SHADOW	Shadow areas obtained by projecting a 3-D structure onto the XY, XZ or YZ planes	13, 14
	DGRAV	Gravitational index	87
	LOVERB	Molecular length to breadth ratio	
Electronic	CHARGE	Dipole moment, charges on most negative and positive atoms and the sum of absolute values of all charges	
	HLEH	HOMO & LUMO energies and electronegativity and hardness	
	MRFRAC	Molecular refraction	88
	MPOLR	Molecular polarizability	89
Hybrid	CPSA	Charged partial surface areas	45

Table 3.1: (continued)

Type	Name	Function	Reference
	DATOM	CPSA descriptors for specific groups and atoms (carbonyl, O, N, S, and halogens)	
	HBSA	Hydrophobic surface areas	54, 90
	HBMIX	Intermolecular hydrogen bonding ability	47, 48
	HBPURE	Intramolecular hydrogen bonding ability	47, 48

Table 3.2. The hydrophobicity and solvent accessible surface area values calculated for glycine. These values are combined to generate the 25 HPSA[54] descriptors.

Serial No.	Atom Label	Hydrophobicity	Surface Area (Å^2)
1	C	-0.20	2.58
2	C	-0.28	7.36
3	O	-0.15	48.00
4	O	-0.29	26.00
5	N	-1.02	19.04
6	H	0.12	25.01
7	H	0.12	25.33
8	H	0.30	30.81
9	H	0.21	29.81
10	H	0.21	21.98

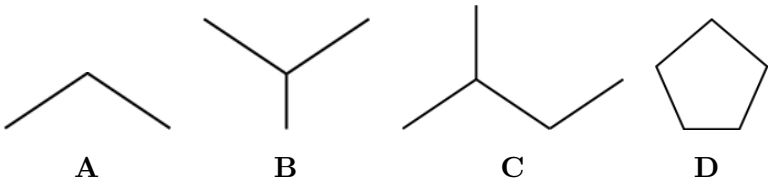


Fig. 3.1. The four types of fragments used to calculate χ descriptors. **A** – 2^{nd} order path. **B** – 3^{rd} order cluster. **C** – 4^{th} order path cluster. **D** – 5^{th} order chain. The order refers to the number of edges in each fragment.

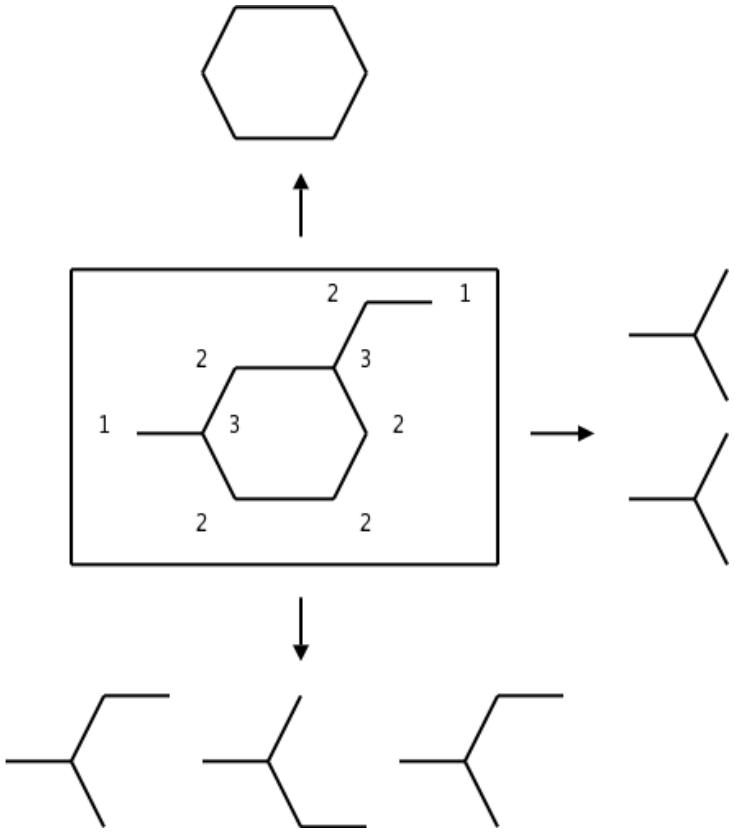
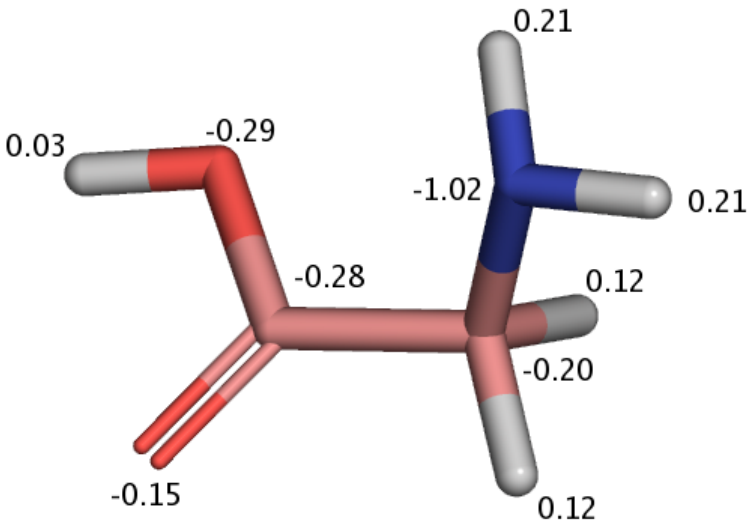
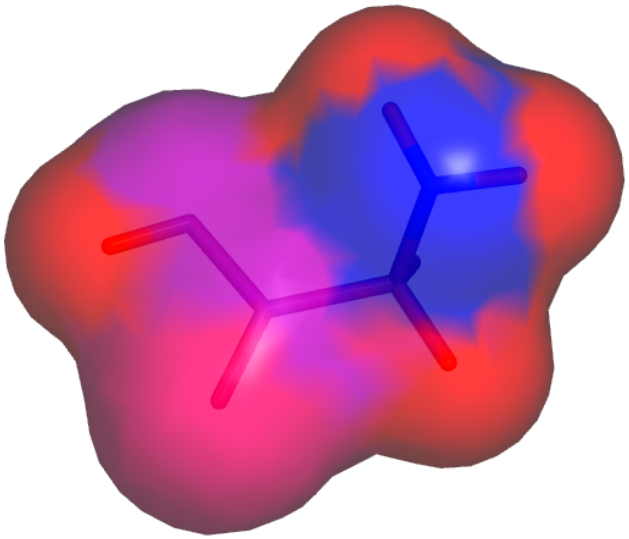


Fig. 3.2. A diagram illustrating the decomposition of 1-methyl 3-ethyl benzene into fragments for subsequent use in the calculation of χ descriptors. The annotations of the central structure correspond to the vertex degree of each atom.



A



B

Fig. 3.3. Graphical representations of hydrophobicity values for the glycine molecule. **A** shows the numerical hydrophobicity values and **B** displays the solvent accessible surface area color coded by the hydrophobicity values. Blue regions indicate the most hydrophilic groups and red corresponds to the most hydrophobic groups.

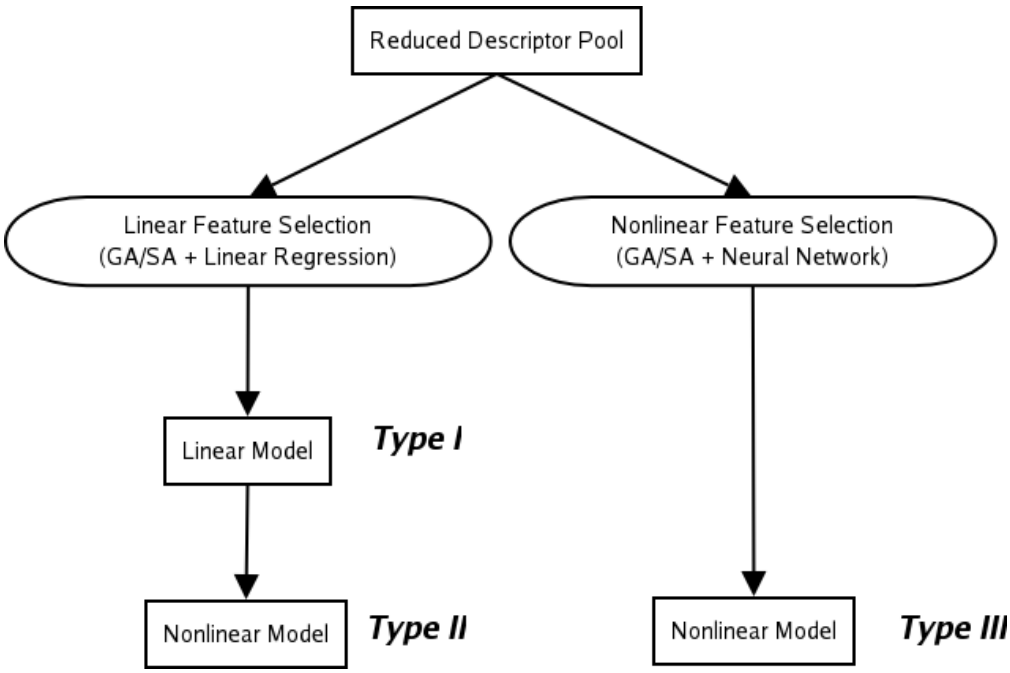


Fig. 3.4. The sequence of steps involved in model building using the ADAPT methodology. Here GA and SA refer to the genetic algorithm and simulated annealing feature selection methods respectively.

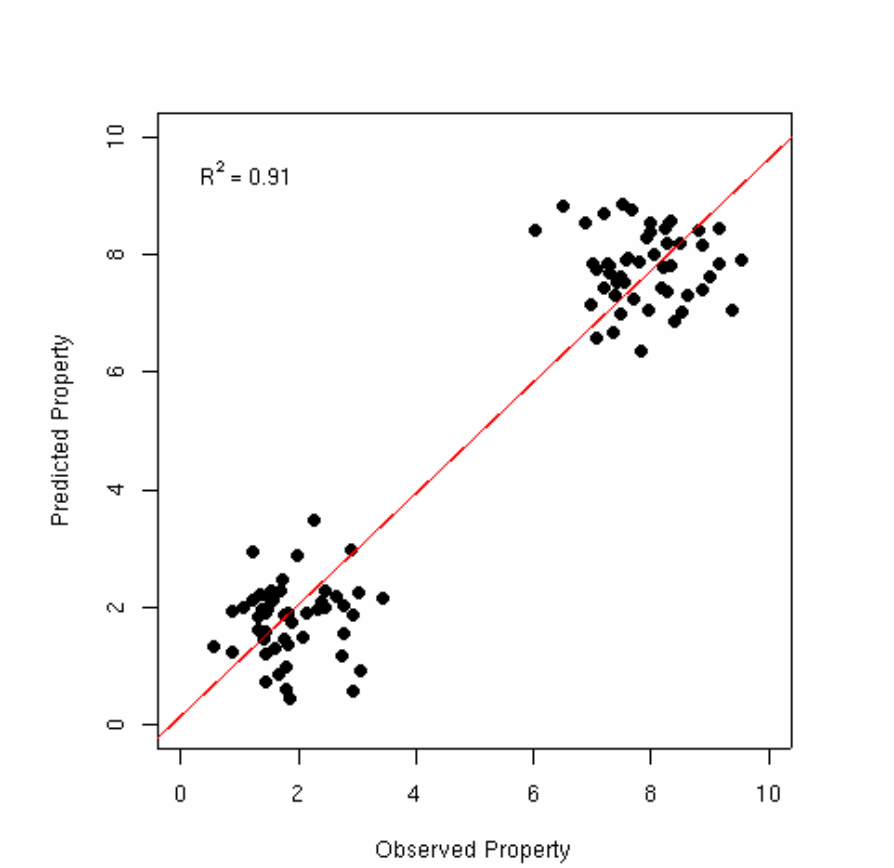


Fig. 3.5. A plot generated from a linear regression model, using simulated data, with a high value of R^2, but clearly unable to explain the variation in the dataset. The red line represents the fitted regression line.

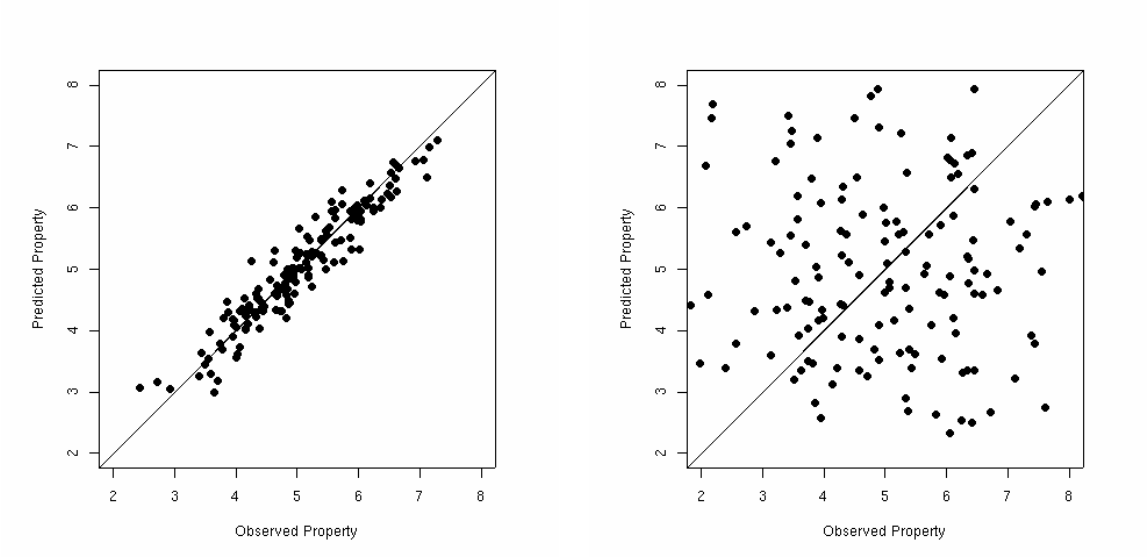


Fig. 3.6. Plots generated using simulated data, illustrating the results of testing chance correlations in a linear model by scrambling the dependent variable. Plot **A** represent the original linear model. Plot **B** represents the linear model rebuilt after scrambling the independent variable.

References

[1] Jurs, P.; Chou, J.; Yuan, M. Computer Assisted Drug Design. In ; American Chemical Society: Washington D.C., 1979; Chapter Studies of Chemical Structure Biological Activity Relations Using Pattern Recognition.

[2] Stuper, A.; Brugger, W.; Jurs, P. *Computer Assisted Studies of Chemical Structure and Biological Function;* Wiley: New York, 1979.

[3] Stewart, J. Optimization of Parameters for Semi-Empirical Methods I – Method. *J. Comp. Chem.* **1989,** *10,* 209–220.

[4] Stewart, J. Optimization of Parameters for Semi-Empirical Methods II – Applications. *J. Comp. Chem.* **1989,** *10,* 221–64.

[5] Dewar, M.; Zoebisch, E.; Healy, E.; Stewart, J. AM1 – A New General Purpose Quantum Mechanical Molecular Model. *J. Am. Chem. Soc.* **1985,** *107,* 5348–5348.

[6] Davis, L.; Burggraf, L.; Storch, D. Hydration of Small Anions - Calculations by the AM1 Semi-Empirical Method. *J. Comp. Chem.* **1991,** *12,* 350–358.

[7] Dewar, M.; McKee, M.; Rzepa, H. MNDO Parameters for Third Period Elements. *J. Am. Chem. Soc.* **1978,** *100,* 3607.

[8] Todeschini, R.; Consonni, V.; Pavan, M. "DRAGON", .

[9] Wegner, J. "JOELib", `http://joelib.sf.net`, 2005.

[10] Todeschini, R.; Consonni, V. *Handbook of Molecular Descriptors;* Wiley-VCH: Berlin, 2002.

[11] Goldstein, H. *Classical Mechanics;* Addison Wesley: Reading, MA, 1950.

[12] Pearlman, R. Physical Chemical Properties of Drugs. In ; Marcel Drekker, Inc.: New York, 1980; Chapter Molecular Surface Area and Volumes and their Use in Structure-Activity Relationships.

[13] Stouch, T.; Jurs, P. A Simple Method for the Representation, Quantification, and Comparison of the Volumes and Shapes of Chemical Compounds. *J. Chem. Inf. Comput. Sci.* **1986,** *26,* 4–12.

[14] Rohrbaugh, R.; Jurs, P. Molecular Shape and Prediction of High Performace Liquid Chromatographic Retention Indices of Polycyclic Aromatic Hydrocarbons. *Anal. Chem.* **1987,** *59,* 1048–1054.

[15] Kier, L.; Hall, L.; Murray, W. Molecular Connectivity I: Relationship to local anasthesia. *J. Pharm. Sci.* **1975,** *64,* 1971–1974.

[16] Kier, L.; Hall, L. *Molecular Connectivity in Structure Activity Analysis;* John Wiley & Sons: Hertfordshire, England, 1986.

[17] Kier, L.; Hall, L. Molecular Connectivity VII: Specific Treatment to Heteroatoms. *J. Pharm. Sci.* **1976,** *65,* 1806–1809.

[18] Liu, S.; Cao, C.; Li, Z. Approach to Estimation and Prediction for Normal Boiling Point (NBP) of Alkanes Based on a Novel Molecular Distance Edge (MDE) Vector, λ. *J. Chem. Inf. Comput. Sci.* **1998,** *38,* 387–394.

[19] Sharma, V.; Goswami, A.; Madan, A. Eccentric Connectivity Index: A Novel Highly Discriminating Topological Descriptor for Structure-Property and Structure-Activity Studies. *J. Chem. Inf. Comput. Sci.* **1998,** *37,* 273–282.

[20] Consonni, V.; Todeschini, R.; Pavan, M. Structure-Response Correlations and Similarity/Diversity Analysis By GETAWAY Descriptors. Part 1. Theory of the Novel 3D Molecular Descriptors. *J. Chem. Inf. Comput. Sci.* **2002,** *42,* 682–692.

[21] Randic, M. On Molecular Idenitification Numbers. *J. Chem. Inf. Comput. Sci.* **1984,** *24,* 164–175.

[22] Pearlman, R.; Smith, K. 3D-QSAR in Drug Design. In ; Kubinyi, H. e. a., Ed.; Kluwer/Escom: Dordrecht, The Netherlands, 1998.

[23] Burden, F. Molecular Identification Number for Substructure Searches. *J. Chem. Inf. Comput. Sci.* **1989,** *29,* 225–227.

[24] Burden, F. A Chemically Intuitive Molecular Index Based on the Eigenvalues of a Modified Adjacency Matrix. *Quant. Struct. Act. Relat.* **1997,** *16,* 309–314.

[25] Pearlman, R.; Smith, K. Novel Software Tools for Chemical Diversity. *Persp. Drug Discov. Design* **1998,** *9,* 339–353.

[26] Stanton, D. Evaluation and Use of BCUT Descriptors in QSAR and QSPR Analysis.. *J. Chem. Inf. Comput. Sci.* **1999,** *39,* 11–20.

[27] Martin, Y. Diverse Viewpoints on Computational Aspects of Molecular Diversity. *J. Comb. Chem.* **2001,** *3,* 231–250.

[28] Schnur, D. Design and Diversity Analysis of Large Compound Libraries Using Cell-Based Methods. *J. Chem. Inf. Comput. Sci.* **1999,** *39,* 36–45.

[29] Young, S. S.; Wang, M.; Gu, F. Design of Diverse and Focused Combinatorial Libraries Using an Alternating Algorithm. *J. Chem. Inf. Comput. Sci.* **2003,** *43,* 1916–1921.

[30] Shanmugasundaram, V.; Maggiora, G.; Lajiness, M. Hit Directed Nearest-Neighbor Searching. *J. Med. Chem.* **2005,** *48,* 240–248.

[31] Stanton, D. T.; Morris, T. W.; Roychoudhury, S.; Parker, C. N. Application of Nearest-Neighbor and Cluster Analysis in Pharmaceutical Lead Discovery. *J. Chem. Inf. Comput. Sci.* **1999,** *39,* 21–27.

[32] Manallack, D. T.; Pitt, W. R.; Gancia, E.; Montana, J. G.; Livingstone, D. J.; Ford, M. G.; Whitley, D. C. Selecting Screening Candidates for Kinase and G Protein-Coupled Receptor Targets Using Neural Networks. *J. Chem. Inf. Comput. Sci.* **2002,** *42,* 1256–1262.

[33] Todeschini, R.; Cazar, R.; Collina, E. The Chemical Meaning of Topological Indices. *Chemom. Intell. Lab. Sys.* **1975,** *97,* 6609–6615.

[34] Randic, M.; Zupan, J. On Interpretation of Well Known Topological Indices. *J. Chem. Inf. Comput. Sci.* **2001,** *41,* 550–560.

[35] Randic, M.; Balaban, A.; Basak, S. On Structural Interpretation of Several Distance Related Topological Indices. *J. Chem. Inf. Comput. Sci.* **2001,** *41,* 593–601.

[36] Dixon, S.; Jurs, P. Atomic Charge Calculations for Quantitative Structure-Property Relationships. *J. Comp. Chem.* **1992,** *13,* 492–504.

[37] Abraham, R.; Griffiths, L.; Loftus, J. Approaches to Charge Calculations in Molecular Mechanics. *J. Comp. Chem.* **1982,** *3,* 407–416.

[38] Dixon, S. L. *Development of Computational Tools for Use in Quantitative Structure-Activity and Structure-Property Relationships,* PhD thesis, Department of Chemistry, Pennsylvania State University, 1994.

[39] Estrada, E.; Perdomo-Lopez, I.; Torres-Labandeira, J. J. Combination of 2D-, 3D-Connectivity and Quantum Chemical Descriptors in QSPR. Complexation of α- and β-Cyclodextrin with Benzene Derivatives. *J. Chem. Inf. Comput. Sci.* **2001,** *41,* 1561–1568.

[40] Netzeva, T. I.; Aptula, A. O.; Benfenati, E.; Cronin, M. T. D.; Gini, G.; Lessigiarska, I.; Maran, U.; Vracko, M.; Schuurmann, G. Description of the Electronic Structure of Organic Chemicals Using Semiempirical and Ab Initio Methods for Development of Toxicological QSARs. *J. Chem. Inf. Model.* **2005,** *45,* 105–114.

[41] Karelson, M.; Lobanov, V.; Katritzky, A. Quantum-Chemical Descriptors in QSAR/QSPR Studies. *Chem. Rev.* **1996,** *105,* 7512–7516.

[42] Thanikaivelan, P.; Subramnian, V.; Rao, J.; Nair, B. Application of Quantum Chemical Descriptors in Quantitative Structure Activity and Structure Property Relationships. *Chem. Phys. Lett.* **2000,** *323,* 59–70.

[43] Whitehead, C.; Sukumar, N.; Breneman, C. Transferable Atom Equivalent Multi-Centered Multipole Expansion Method. *J. Comp. Chem.* **2003,** *24,* 512–529.

[44] Breneman, C.; Sundling, C.; Sukumar, N.; Shen, L.; Katt, W. New Developments in PEST Shape/Property Hybrid Descriptors. *J. .Comp. Aided Mol. Des.* **2003,** *17,* 213–240.

[45] Stanton, D.; Jurs, P. Development and Use of Charged Partial Surface Area Structural Descriptors in Computer Assissted Quantitative Structure Property Relationship Studies. *Anal. Chem.* **1990,** *62,* 2323–2329.

[46] Stanton, D. T.; Mattioni, B. E.; Knittel, J. J.; Jurs, P. C. Development and Use of Hydrophobic Surface Area (HSA) Descriptors for Computer-Assisted Quantitative Structure-Activity and Structure-Property Relationship Studies. *J. Chem. Inf. Comput. Sci.* **2004,** *44,* 1010–1023.

[47] Pimentel, G.; McClellan, A. *The Hydrogen Bond;* Reinhold Pub. Corp.: New York, 1960.

[48] Vinogradov, S.; Linnell, R. *Hydrogen Bonding;* Van Nostrand Reinhold: New York, 1971.

[49] Palm, K.; Luthman, K.; Ungell, A.-L.; Strandlund, G.; Beigi, F.; Lundahl, P.; P., A. Evaluation of Dynamic Polar Molecular Surface Area as Predictor of Drug Absorption: Comparison with Other Computational and Experimental Predictors. *J. Med. Chem.* **1998,** *41,* 5382–5392.

[50] Stenberg, P.; Luthman, K.; Artursson, P. Prediction of Membrane Permeability to Peptides from Calculated Dynamic Molecular Surface Properties. *Pharm. Res.* **1999,** *16,* 972–978.

[51] Palm, K.; Stenberg, P.; Luthman, K.; Artursson, P. Polar Molecular Surface Properties Predict the Intestinal Absorption of Drugs in Humans. *Pharm. Res.* **1997,** *14,* 568–571.

[52] Clark, D. Rapid Calculation of Polar Molecular Surface Area and its Application to the Prediction of Transport Phenomena. 2. Prediction of Blood-Brain Barrier Penetration. *J. Med. Chem.* **1999,** *88,* 815–821.

[53] Ertl, P.; Rohde, B.; Selzer, P. Fast Calculation of Molecular Polar Surface Area as a Sum of Fragment-based Contributions and its Application to the Prediction of Drug Transport Properties. *J. Med. Chem.* **2000,** *43,* 3714–3717.

[54] Stanton, D.; Mattioni, B. E.; Knittel, J.; Jurs, P. Development and Use of Hydrophobic Surface Area (HSA) Descriptors for Computer Assisted Quantitative Structure-Activity and Structure-Property Relationships. *J. Chem. Inf. Comp. Sci.* **2004,** *44,* 1010–1023.

[55] Guha, R.; Jurs, P. C. The Development of Linear, Ensemble and Non-Linear Models for the Prediction And Interpretation of the Biological Activity of a Set of PDGFR Inhibitors. *J. Chem. Inf. Comput. Sci.* **2004,** *44,* 2179–2189.

[56] Andersson, P.; Lundstedt, T. Hierarchical Experimental Design Exemplified By QSAR Evaluation of a Chemical Library Directed Towards the Melanocortin–4 Receptor. *J. Med. Chem.* **2002,** *16,* 490–496.

[57] Wu, W.; Walczak, B.; Massart, D.; Heuerding, S.; Erni, F.; Last, I.; Prebble, K. Artificial Neural Networks In Classification of NIR Spectral Data: Design of the Training Set. *Chemom. Intell. Lab. Sys.* **1996,** *33,* 35–46.

[58] Kocjancic, R.; Zupan, J. Modelling of the River Flowrate: The Influence of the Training Set Selection.. *Chemom. Intell. Lab. Sys.* **2000,** *54,* 21–34.

[59] Arfken, G.; Weber, H. *Mathematical Methods for Physicists;* Harcourt/Academic Press: San Diego, CA, 2000.

[60] Topliss, J.; Edwards, R. Chance Factors in Studies of Quantitative Structure-Activity Relationships. *J. Med. Chem.* **1979,** *22,* 1238–1244.

[61] Cramer, N. A Representation for the Adaptive Generation of Simple Sequential Programs. In *Proc. of the International Conference on Genetic Algorithms and their Applications*; Lawrence Erlbaum Associates: Pittsburgh, PA, 1985.

[62] Goldberg, D. *Genetic Algorithms in Search, Optimization & Machine Learning;* Addison-Wesley: Reading, MA, 2000.

[63] Metropolis, N.; Rosenbluth, A.; Rosenbluth, M.; Teller, A.; Teller, E. Equation of State Calculations by Fast Computing Machines. *J. Chem. Phys.* **1953,** *21,* 1087–1092.

[64] Agrafiotis, D. K.; Cedeño, W. Feature Selection for Structure-Activity Correlation Using Binary Particle Swarms. *J. Med. Chem.* **2002,** *45,* 1098–1107.

[65] Izrailev, S.; Agrafiotis, D. A Novel Method for Building Regression Tree Models for QSAR Based on Artificial Ant Colony Systems. *J. Chem. Inf. Comput. Sci.* **2001,** *41,* 176–180.

[66] Agrafiotis, D.; Cedeño, W.; Lobanov, V. On the Use of Neural Network Ensembles in QSAR and QSPR. *J. Chem. Inf. Comput. Sci.* **2002,** *42,* 903–911.

[67] Golbraikh, A.; Tropsha, A. Beware of q^2. *J. Mol. Graph. Model.* **2002,** *20,* 269–276.

[68] Golbraikh, A.; Shen, M.; Xiao, Z.; Xiao, Y.; Lee, K.; Tropsha, A. Rational Selection of Training and Test Sets for the Development of Validated QSAR Models. *J. Comput. Aid. Mol. Des.* **2003,** *17,* 241–253.

[69] Kubinyi, H.; Hamprecht, F.; Mietzner, T. Three-Dimensional Quantitative Similarity-Activity Relationships (3D QSAR) from SEAL Similarity Matrices. *J. Med. Chem.* **1998,** *41,* 2553–2564.

[70] Novellino, E.; Fattorusso, C.; Greco, G. Use of Comparative Molecular Field Analysis and Cluster Analysis in Series Design. *Pharm. Acta. Helv.* **1995,** *70,* 149–154.

[71] Norinder, U. Single and Domain Made Variable Selection in 3D QSAR Applications. *J. Chemom.* **1996,** *10,* 95–105.

[72] Stanton, D. On the Physical Interpretation of QSAR Models. *J. Chem. Inf. Comput Sci.* **2003,** *43,* 1423–1433.

[73] Stanton, D. "personal communication", 2004.

[74] Guha, R.; Jurs, P. C. The Development of QSAR Models To Predict and Interpret the Biological Activity of Artemisinin Analogues. *J. Chem. Inf. Comput. Sci.* **2004,** *44,* 1440–1449.

[75] Svetnik, V.; Liaw, A.; Tong, C.; Culberson, C.; Sheridan, R.; Feuston, B. Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling. *J. Chem. Inf. Comput. Sci.* **2003,** *42,* 1947–1958.

[76] Breiman, L.; Friedman, J.; Olshen, R.; Stone, C. *Classification and Regresion Trees;* CRC Press: Boca Raton, FL, 1984.

[77] Kier, L. A Shape Index From Molecular Graphs. *Quant. Struct.-Act. Relat. Pharmacol.,Chem. Biol.* **1985,** *4,* 109–116.

[78] Kier, L. Shape Indexes for Orders One and Three From Molecular Graphs. *Quant. Struct.-Act. Relat. Pharmacol.,Chem. Biol.* **1986,** *5,* 1–7.

[79] Kier, L. Distinguishing Atom Differences in a Molecular Graph Index. *Quant. Struct.-Act. Relat. Pharmacol.,Chem. Biol.* **1986,** *5,* 7–12.

[80] Randic, M.; Brissey, G.; Spencer, R.; Wilkins, C. Search for All Self-Avoiding Paths Graphs for Molecular Graphs. *Comput. Chem.* **1979,** *3,* 5–14.

[81] Wiener, H. Structural Determination of Paraffin Boiling Points. *J. Am. Chem. Soc.* **1947,** *69,*.

[82] Balaban, A. Higly Discriminating Distance Based Topological Index. *Chem. Phys. Lett.* **1982,** *89,* 399–404.

[83] Kier, L.; Hall, L. *Molecular Connectivity in Chemistry and Drug Research;* Academic Press: New York, 1976.

[84] Kier, L.; Hall, L. *Molecular Structure Description. The Electrotopological State;* Academic Press: London, 1999.

[85] Kier, L.; Hall, L. An Electrotopological-State Index for Atoms In Molecules. *Pharm. Res.* **1990,** *7,* 801–807.

[86] Gupta, S.; Singh, M.; Madan, A. Superpendentic Index: A Novel Topological Descriptor for Predicting Biological Activity. *J. Chem. Inf. Comput. Sci.* **1999,** *39,* 272–277.

[87] Katritzky, A.; Mu, L.; Lobanov, V.; Karelson, M. Correlation of Boiling Points with Molecular Structure. 1. A Training Set of 298 Diverse Organics and a Test Set of 9 Simple Inorganics. *J. Phys. Chem.* **1996,** *100,* 10400–10407.

[88] Vogel, A. *Textbook of Organic Chemistry;* Chaucer Press: London, 1970.

[89] Miller, K.; Savchik, J. A New Empirical Method to Calculate Average Emiprical Polarizabilities. *J. Am. Chem. Soc.* **1979,** *101,* 7206–7213.

[90] Mattioni, B. E. *The Development of Quantitative Structure-Activity Relationship Mode Physical Property and Biological Activity Prediction of Organic Compounds,* PhD thesis, Department of Chemistry, Pennsylvania State University, 2003.

Chapter 4

Generation of QSAR Sets Using a Self-Organizing Map

4.1 Introduction

As mentioned in Chapter 2, self-organizing maps[1] (SOM) are a class of unsupervised neural networks whose characteristic feature is their ability to map nonlinear relations in multi-dimensional datasets into easily visualizable two-dimensional grids of neurons. SOM's are also referred to as self-organized topological feature maps since the basic function of a SOM is to display the topology of a dataset, that is, the relationships between members of the set. SOM's were first developed by Kohonen in the 1980's, and since then they have been used as pattern recognition and classification tools in various fields including robotics,[2] astronomy,[3] and chemistry.

Neural networks, in general, have been used extensively in chemistry[4] and chemometrics and examples of applications in chemistry include spectroscopy,[5–8] prediction of NMR properties[9] and prediction of reaction products[10,11]

SOM's have also been applied to studies in the field of QSAR/QSPR.[12] The fundamental premise of QSAR studies is that structurally related (similar) compounds will have similar properties. Determining similarity is a complex task, and many methods exist such as principal components analysis and hierarchical cluster analysis. The fact that a SOM is able to extract topological information from a dataset makes it a valuable tool for detecting similarities in a dataset. Thus, it is to be expected that neighboring neurons in a two-dimensional SOM grid will be similar to each other. If each neuron in such a SOM grid can be assigned a molecule, groups of similar molecules can be identified.

Many studies have used a SOM to perform the actual QSAR[13–16] analysis by detecting relationships between structures and activities of interest. Other applications use SOM's at different stages of the QSAR study, for example, the use of a SOM to choose the best subset of molecular descriptors[17,18] to perform a QSAR analysis. However, another important step in QSAR study is the generation of training, cross-validation,

---

This work was published as Guha, R.; Serra, J.; Jurs, P.C., "Generation of QSAR Sets with a Self-Organizing Map", *J. Mol. Model. Graph.*, **2004**, *23*, 1–14.

and prediction sets. A number of methods exist, including random selection, activity-ranked binning, and sphere exclusion algorithms.[19] A number of studies have focused on approaches to select the training set. These approaches include classical statistical design methods such as Kennard-Stone,[8,20] and D-Optimal[8] as well as Kohonen neural networks.[8,20–22] Set selection is also an important step in QSAR modeling of chemical libraries. Most strategies for this are based on a combination of principal components analysis (PCA) for dimensionality reduction followed by statistical molecular design[23–25] (SMD).

The goal of this study was to implement a set generation technique, utilizing a SOM, together with whole molecule descriptors, to initially classify the dataset and subsequently use this classification to generate training, cross-validation, and prediction sets for QSAR studies, whose composition would mirror the overall composition of the entire dataset. The expectation was, that this technique should lead to the generation of QSAR models that exhibit equal or higher validity than models generated from subsets developed with random selection or activity-ranked binning. The distribution of the members of the training and prediction sets (with respect to each other in *descriptor space*) was also studied by calculating a molecular diversity index.[26] In addition, the results from SOM generated QSAR sets were compared to results obtained using QSAR sets created using traditional activity binning, as well as sets created using a sphere exclusion algorithm described by Golbraikh.[19]

4.2 Implementation of an SOM

A Kohonen self-organizing map (SOM) is an unsupervised neural network that uses as its inputs only the independent variables of the dataset here, molecular structure descriptors. The theoretical details of the SOM can be found in Section 2.2.2.

The implementation for this study consisted of a 13x13 grid. A Gaussian kernel was used, and the learning factor α was set to an initial value of 1.0 and was decremented with a constant decrement of 0.1 per training iteration. The dataset we used to test this method consisted of 333 molecules. According to Chen[11] the grid should contain 333 to 999 neurons. This translates to grid sizes ranging from 18x18 up to 31x31. However, we noted that for grids larger than 15x15 the SOM converged to a configuration in which the training set was mapped relatively evenly over the grid with little apparent clustering. In addition the use of larger grids increased the running times significantly. The method of choosing a grid size does appear to be arbitrary. However, given the fact that following

Chen's rule of thumb produced grids with hardly any clustering observable, we felt that examining smaller grid sizes was justified. Using a 13x13 grid of neurons the SOM usually required 80 to 90 training iterations for the grid neurons to converge to their final values. Depending on the number of descriptors used to represent each compound, this took approximately 3 to 6 minutes on an AMD 750MHz Duron processor running RedHat Linux 7.3.

After the SOM was trained, the results were analyzed to detect clusters of neurons. In this context, a cluster refers to neurons that have similar Euclidean distances from each other. As mentioned in Satoh,[10] "recognition of boundaries of clusters in a Kohonen network is a difficult task". This was implemented by considering two neurons, having a distance less than a user specified value, to be a part of the same cluster. Starting with an arbitrary neuron, we assigned an arbitrary class label. Next, we considered the distances to all the nearest neighbor neurons. Using the rule mentioned above, the neighboring neurons were assigned classes; either the same class as the initial neuron or the opposite class. This procedure was then repeated with all the neurons in the grid. An example of the grid layout after cluster detection (using three different threshold values) is shown in Fig. 4.1. The diagrams are based on the grid generated using the BCUT & 2D Autocorrelation descriptor combination, which are described in a subsequent section.

The final step in this procedure was to assign classes to the actual dataset members by submitting each dataset vector to the trained grid. The class of the closest grid neuron (in terms of Euclidean distance) was assigned to the dataset member.

The result of the cluster detection procedure was to divide the dataset into two classes. Fig. 4.2 shows how the classified dataset is distributed over the SOM. As mentioned before, the arbitrariness of cluster detection lies in the fact that the user must specify a distance threshold value. Too small a value or too large a value results in all the dataset members being assigned to the same class. As the threshold value progresses from zero to larger values the SOM generates a bulk class containing the majority of the dataset members and a minor class. At one point, the populations of both classes will be approximately equal, and then with further increase of the threshold value the populations once again get skewed. It is thus clear that the threshold value must be chosen carefully. Below we describe the method that we employed to arrive at a threshold value.

It should be noted that the classification of the dataset by the SOM is not intended to correspond to a classification based on any structure-activity relationship. The aim of the classification is to simply divide the dataset into two sets differing in structural features, as characterized by whole molecule descriptors.

4.3 Using the SOM to Create Sets

In this study, the SOM was used to generate training, prediction and cross-validation sets (hereafter referred to collectively as QSAR sets) for QSAR studies using the ADAPT[27,28] methodology. Previously, these sets had been generated by randomly selecting the requisite number of molecules from the binned (based on activity) dataset. However, owing to the random selection process, the binning procedure does not necessarily create sets that represent the composition of the whole dataset. Yan and Gasteiger[22] used a SOM to select QSAR sets, in which sets were created by simple selection of grid points. As a result thier method is similar to the sphere exclusion technique, in that there is a correspondence between the training and prediction set points in descriptor space. However the technique described by Yan and Gasteiger[22] does not necessarily maintain a correspondence between the composition of the QSAR sets and the overall dataset. Our method emphasizes the use of characteristic features of the dataset to create sets whose composition would mirror the overall dataset. This is achieved by using the SOM to divide the dataset into two classes, based on the molecular structure descriptors representing the compounds of the dataset. These two classes thus represent the SOM classification of the whole dataset into a major and minor class (say, Class I and Class II, respectively).

As described above, the threshold value controls the population of the two classes. We initially ran the SOM with the threshold value set to zero. The output of this run reported the distances between all the neurons in the grid. This distance information was used to determine the range of threshold values to be considered in subsequent runs of the SOM. The next step was to run the SOM several times in succession, with threshold values ranging from about 5% to 90% of the maximum distance reported in the initial run. Each run generated a set of class assignments. We considered those runs that generated a bulk class having approximately 80% of the entire dataset. The difference between the populations of the bulk and minor class for each of these runs, D, was noted. A large jump in the value of D was usually seen at one point in the series. This can be seen in Fig. 4.3, which plots D versus the threshold value (represented as a percentage of the maximum distance in the grid when the threshold value is set to zero). The descriptor subset supplied to these SOM runs was the MoRSE-WHIM subset. The classification results from the run that generated the lower value of D for the jump were used for the subsequent creation of QSAR sets. From Fig. 4.3 it is apparent that there is a large jump from 23% to 24% as well from 4% to 5%. However, we did not consider

these jumps since the number of molecules in the bulk class for these jumps was not close to 80% of the whole dataset. Instead, the grid configuration that corresponds to the jump from 9% to 11% had a bulk class that contained 80.1% of the whole dataset. Thus the grid results from the run using a threshold value of 11% were used subsequently. After the dataset had been classified, the information produced was used to create the actual QSAR sets. At this point the SOM had classified the dataset into two classes (Class I and Class II), members of each class being similar to each other but dissimilar to members of the other class.

Now, for example, say that Class I contains 75% of the whole dataset and Class II contains the other 25%. Our premise is that QSAR sets that contain Class I and Class II molecules distributed according to their percentages in the overall dataset will be more representative of the overall dataset and thus should lead to good predictive models. Continuing with the example, let us assume that we have a dataset of 100 molecules and the SOM classifier splits this dataset in to 75 molecules in class I and 25 molesules in class II. We also assume that for the QSAR sets, the training set should contain 80% of the dataset and the cross-validation and prediction sets should each contain 10%. To make the training set composition similar to that of the overall dataset it will have 80 compounds, of which 75% (60 compounds) will be from class I and 25% (20 compounds) will be from class II. Similarly the cross-validation and prediction sets will each have 10 compounds, of which 75% (8 compounds) will be from class I and 25% (2 compounds) will be from class II. Due to rounding, the final QSAR sets may not have the exact number of compounds described, but can differ by 1. The breakup of the QSAR sets among the SOM classes discussed above is represented diagrammatically in Fig. 4.4 with the exact numbers of compounds rounded appropriately.

Unlike methods such as the sphere exclusion method, discussed below, there is no guarantee that the QSAR sets generated cover the entire descriptor space. Though it is possible that a specific QSAR set is generated by sampling points from a small region of the grid, while still covering both classes, it appears that this does not occur. Fig. 4.5 shows the distribution of the QSAR sets over the grid. As can be seen, the members of each set seem to be relatively evenly distributed over the grid. The diagrams in Fig. 4.5 are based on the BCUT & 2D-Autocorrelation descriptor combination. The other QSAR sets generated from other Dragon[29] descriptor combinations investigated generated similar plots.

4.4 Sphere Exclusion

This method, described by Golbraikh,[19] uses the concept of molecular diversity[26] coupled with a sphere exclusion algorithm to generate training and prediction sets which satisfy the following criteria: points in the training and prediction sets should be close (in terms of descriptor space) to each other, and the training set should be diverse, as measured by the value of its diversity index.[26]

Golbraikh describes three types of sphere exclusion algorithms. A brief summary of the general sphere exclusion algorithm follows. For a training set with N compounds and described by K descriptors, the compound with the highest activity is first selected and placed in the training set. Next, a radius, R, is calculated. R is given by the formula

$$R = c\left(\frac{V}{N}\right)^{1/K} \tag{4.1}$$

where V is the volume of the space occupied by the points of the dataset in the descriptor space and c is a user defined constant termed the Dissimilarity Level (DL)[26] and essentially controls the number of molecules placed in the training and prediction sets. To simplify calculations, the descriptor space is normalized using the formula

$$X_{ij}^{n} = \frac{X_{ij} - X_{j,\min}}{X_{j,\max} - X_{j,\min}} \tag{4.2}$$

where X_{ij} is the non-normalized j'th descriptor for the i'th molecule and X_{ij}^{n} is the normalized value of the descriptor. Thus after normalization, $V = 1$ and the equation for the radius simplifies to

$$R = c\left(\frac{1}{N}\right)^{1/K} \tag{4.3}$$

After a value of R is obtained, a sphere with this radius is centered at the point chosen above, and all compounds that lie within this sphere (except the center point) are included in the prediction set and removed from the dataset so as not to be considered later. At this point if there are no more points left to consider the algorithm halts, otherwise the distances from the remaining points to the centers of all the spheres considered so far are calculated. The distance is given by

$$d_{ij} = \sqrt{\sum_{a=1}^{K}(X_{ia} - X_{ja})^2} \tag{4.4}$$

where X_i and X_j are the descriptor vectors for the i'th and j'th molecules respectively and K is the number of descriptors. One of the points is chosen to be the center of the next sphere and this process is repeated. The manner of choosing the next point gives rise to 3 variations of the sphere exclusion algorithm: the point that had the smallest d_{ij} , the point that had the largest d_{ij}, or randomly choosing a point. In this study we implemented the first option. The result of this algorithm is to generate a training and prediction set. Since the ADAPT methodology requires the use of a cross-validation set, we randomly selected the required number of molecules out of the training set to create the cross-validation set.

4.5 Descriptors for the SOM

The SOM requires that each compound be represented by a set of molecular structure descriptors. We used an external set of descriptors (from the Dragon[29] program), as opposed to the ADAPT descriptors since we wanted to classify the dataset in terms of global features, rather than specific structural trends. As a result, various subsets of Dragon descriptors which are holistic in nature were used, rather than ADAPT descriptors, many of which concentrate on specific structural features. Another reason for not using ADAPT descriptors is that the resultant QSAR sets would indirectly contain the information generated by the ADAPT descriptors and thus using same descriptors again during model development would lead to the possibility of biased models (in that the same information that was used to arrange the molecules would be used again when predicting their activity).

This technique, thus, proceeds in two stages and requires two sets of descriptors, preferably orthogonal in nature. In the first stage, one set of descriptors is used to classify the dataset with the SOM leading to creation of training, cross-validation and prediction sets. The second stage involves the generation of the actual QSAR model using the second set of ADAPT descriptors and the training, cross-validation and prediction sets created in the first stage.

As mentioned above the, descriptors for the first stage were taken from the Dragon program. Several combinations of the Dragon descriptors were selected to see if they could provide a holistic description of the molecules. The number of descriptors in each combination was reduced using correlation and identical testing before using them in the SOM algorithm. A brief description of the descriptors used for the SOM clustering follows. The size of each reduced Dragon descriptor set is shown in Table 4.1.

The BCUT metrics[30–32] are hybrid descriptors derived from the Burden parameters[30] which originally combined the atomic number of an atom and the bond types for adjacent and non-adjacent atoms. The BCUT metrics improve upon the number and type of atomic features that can be encoded. This descriptor has shown significant utility in the measurement of molecular diversity.[33]

Autocorrelation descriptors are based on the autocorrelation function defined[34] as

$$AC_l = \int_a^b f(x)f(x+l)\ dx \tag{4.5}$$

where $f(x)$ is a function of x, l is an interval of x and a and b are the limits of the interval under consideration. $f(x)$ is generally a time-dependent function or in the case of molecular descriptors a spatially-dependent function, in which the atoms of a molecule define the points in space and $f(x_i)$ represents some atomic property for the i^{th} atom. Both 2-D and 3-D autocorrelation descriptors can be calculated and for this study we restricted ourselves to three types of 2-D autocorrelation descriptors - Moreau-Broto,[35–37] Moran,[38] and Geary.[39] These descriptors sum products of atom properties of terminal atoms of all paths of a specific path length.

The Galvez topological charge indices[40–42] use the distance matrix to evaluate charge terms which characterize the charge transfer between individual atoms in the molecule.

The GETAWAY[43–45] descriptors are based on the information contained within the molecular influence matrix.[44] They combine the geometrical information in the influence matrix and topological information in the molecular graph weighted by various atomic properties. As a result, there are two sets of GETAWAY descriptors, the H & R GETAWAY, both of which we chose to use in the classification stage.

3D-MoRSE[46,47] descriptors are fixed length representations of 3D molecular structure and are based on the algorithm used for the analysis of electron diffraction data. Individual descriptors are obtained by considering different weighting functions as described in the literature.

WHIM[48–53] descriptors describe a molecule in terms of size, shape, symmetry and atom distribution and are based on a principal components analysis on the centered molecular coordinates with different weighting schemes.[52]

Our studies used both individual sets as well as combinations of the above descriptors. The sphere exclusion method also used the same combinations of external Dragon descriptors for the generation of the QSAR sets.

4.6 Results and Discussion

To test this method we generated QSAR models using the 333-compound pcDHFR dataset that was studied by Mattioni.[54] The structures and activity values for all the molecules are contained in above-mentioned reference. To generate the QSAR sets we fed combinations of Dragon descriptors to the SOM, and its output was used to generate the sets. For each Dragon descriptor subset the sizes of the training, cross-validation and prediction sets were the same. To ensure a large enough training set, 80% of the dataset was placed in the training set and the remaining 20% was divided equally amongst the CV and prediction sets. The actual number of molecules in each set is summarized in Table 4.2 . After the QSAR sets were generated, we calculated ADAPT descriptors for the entire dataset of 333 molecules. This generated 248 descriptors for each molecule. The number of descriptors was then reduced via objective feature selection to generate a reduced pool of 74 descriptors. The reduced pool of ADAPT descriptors was then used with the QSAR sets created from each of the Dragon descriptor combination, to build nonlinear computational neural network (CNN) models using the ADAPT methodology. In total we used six combinations of Dragon descriptors to generate six nonlinear CNN QSAR models (Table 4.1).

4.6.1 Nonlinear CNN Models

To generate nonlinear models, the descriptor subsets selected by a genetic algorithm were fed to a 3-layer, fully-connected, feed-forward neural network to test fitness. The best neural network models were those that minimized the cost function defined as

$$\text{Cost} = \text{RMSE}_{TSET} + 0.5 \times |\text{RMSE}_{TSET} - \text{RMSE}_{CVSET}|$$

where RMSE_{TSET} and RMSE_{CVSET} are the RMS errors for the training and cross validation sets respectively. Futher details of the development of CNN models using a genetic algorithm for model selection can be found in Section 3.4.2.

After several of the best (low cost) models were obtained a more rigorous analysis was performed on each model to identify the optimal neural network parameters. The results for the nonlinear models are summarized in Table 4.3. Though the number of descriptors in the two best models (see Table 4.4) is significantly lower than the number in the published model, they are of similar types, the majority being simple structural counts and topological path descriptors.

The MoRSE-2D Autocorrelation Dragon descriptor combination generated QSAR sets which produced a CNN model whose prediction set RMSE value was slightly larger than the original value whereas the prediction set error for the models that were generated from QSAR sets produced by using the GETAWAY and the MoRSE-WHIM Dragon descriptor sets match the predicted value. However, in all cases the RMSE's for the training and cross-validation set were significantly larger than those for the reported model. The higher cross-validation set error could indicate a loss of generalizability in these models. On the other hand the RMSE values for the training and cross-validation sets generated from the MoRSE-GETAWAY combination are much closer to those reported, though the prediction set error is now significantly larger. However, the attractive feature of the models generated from QSAR sets produced by MoRSE-2D Autocorrelation, GETAWAY and MoRSE-WHIM Dragon descriptor sets are that they are 5- or 6-descriptor models. Furthermore, the number of neurons in the hidden layers in these three models are all less than in the published model, indicating a simpler neural network.

Table 4.4 lists the descriptors present in the two best nonlinear models. The two best models have a similar set of ADAPT descriptors when compared to the published model, though none of them include a geometric descriptor. The R^2 values for the two best models (i.e., the models using QSAR sets generated using the MoRSE-2D Autocorrelation and MoRSE-WHIM Dragon descriptor combinations) are close to those reported for the best model. These are summarized in Table 4.5. The R^2 values for the training and cross-validation sets produced by the MoRSE-WHIM combination compare favorably to those published. The R^2 value for prediction set produced by the MoRSE-WHIM combination is a little higher than the reported value, but is not significantly larger. Considering the fact that the R^2 for the prediction set, produced by the MoRSE-2D Autocorrelation combination, is the same as that published this could indicate that a combination of MoRSE, 2D Autocorrelation and WHIM descriptor sets would lead to QSAR sets which would lead to a CNN model with better correlation coefficients overall.

However, it should be noted that though the R^2 value is a good test for evenly distributed data, it is not always reliable for an unevenly distributed dataset as the one used in this study. As a result we feel that the RMSE values provide a more reliable indication of the fitness of a model.

The plot for the predicted versus experimental values for the model generated using QSAR sets produced using the MoRSE-WHIM Dragon descriptor combination is shown in Fig. 4.6. Molecules in the prediction set were classified as outliers if their predicted value was two standard deviations away from the mean. This criterion led to

one outlier, whose structure is shown in Fig. 4.7. The best, published model also classified a single outlier. Ideally, we would like the same outliers to be detected by either method. Although this is not the case, it should be noted that there is a structural similarity in the outliers presented in Fig. 4.7. Although the original work does not provide an explanation of why that outlier is not predicted well, the fact that the SOM based technique predicts a structurally similar outlier indicates that that this technique is able to take into account similarity features of the dataset in the creation of the QSAR sets.

An important feature of the two best CNN models (using QSAR sets generated from the MoRSE-2D Autocorrelation and the MoRSE-WHIM Dragon descriptor combinations) is the consistency between the RMSE's for the training, cross-validation, and prediction sets. In many cases a low RMSE for the prediction set would be indicative of a good predictive model. However, at the same time, if the RMSE for the training and cross-validation sets are much lower than that of the prediction set it could indicate that the model lacks generalizability. Thus one would strive for models that have similar or consistent RMSE values for all the three QSAR sets. As can be seen, this does not occur for the original published results. However, for the best CNN models generated by the SOM-based method, though the RMSE's for the training and cross-validation sets are higher than those reported in the original model, the RMSE's are more consistent over all the three sets. The standard deviation of the RMSE's for the three QSAR sets in the original model is 0.11 whereas the standard deviations in the case of the two best models noted above are 0.02 in both cases. This suggests that the models generated by this method have both sufficient generalizability as well as predictive ability. However, apart from the conclusions regarding the nature of the models themselves, these results are indicative of the fact that the QSAR sets that are generated by the SOM are indeed similar to each other and representative of the data set as a whole thus leading to similar predictions made during training and after training (using the external prediction set).

We also reran the original, published 10–6–1 CNN model five times with different QSAR sets generated using activity binning. The results obtained are summarized in Table 4.6. As can be seen there is a large variation in the RMSE's for the three QSAR sets in each run. Furthermore, when compared to the RMSE's for the best CNN models generated using QSAR sets created by the SOM, we see that the SOM results in general lie midway between the RMSE values from the 10–6–1 models using QSAR sets from activity binning. We believe that this is a good indication for the consistency of results obtained using the SOM to generate representative QSAR sets.

It thus appears that the technique of using a group of external descriptors coupled with a SOM to generate sets for QSAR modeling do generate improved results. The ability of the SOM to detect similarities in the dataset allows us to generate sets that are more representative of the overall data set. As a result models with fewer parameters (i.e., descriptors) are able to produce results comparable to the original model that had nearly twice the number of parameters and in addition produce consistent RMSE's over the three QSAR sets.

4.6.2 Sphere Exclusion

For comparison, results of CNN models generated using different QSAR sets created by the sphere exclusion method are presented in Table 4.7. For each set of external descriptors used, the model with lowest cost is reported. None of the models seem to be significantly better than the published model. The architectures are not significantly simpler than the reported 10–6–1 architecture and the R^2 values and RMSE's are comparable, though none of the models seem to provide an improvement over the published statistics. In addition, there is not much of a difference in the RMSE values for models that are generated from QSAR sets that were created using different Dragon descriptor combinations. However, when comparing the results from the sphere exclusion method to those obtained from the SOM technique it appears that the SOM generated QSAR sets produce better models in terms of size (i.e., requiring fewer descriptors), with RMSE's being comparable. In addition the RMSE's for the three QSAR sets in the models generated by the sphere exclusion method do not show much consistency. The RMSE for the prediction set is usually higher than the RMSE's for the training and cross-validations sets by 0.1 to 0.3. This is similar to the nature of the RMSE's in the original model. Due to the nature of the sphere exclusion algorithm one would expect that the resultant QSAR sets would be similar to each other and thus lead to consistent RMSE's. The fact that it does not, is a possible indication that a simple Euclidean distance between individual molecular descriptor vectors is not sufficient to characterize similarity of molecules of in a dataset. Thus the sphere exclusion method does not appear to generate QSAR sets that can produce models with both generalizability as well as predictive ability for this dataset.

4.6.3 Randomization Studies

The best nonlinear model (i.e., the one generated using QSAR sets produced by the MoRSE-WHIM Dragon descriptor combination) was subjected to randomization tests. The first set of tests involved generating random training, cross-validation and prediction sets. These sets were then used to generate a nonlinear model with a 6–5–1 CNN architecture five times (each time using randomly generated sets) and noting the average RMSE's. In addition, the variance between the five individual runs for the random sets was also compared to the variance for five runs of the original QSAR sets that gave the best 6-descriptor CNN model. The correlation coefficient for each of the sets in each of the runs was also compared to the correlation coefficients for the best model. The results are summarized in Table 4.8. The average correlation coefficient (R^2) for the random training, cross-validation and prediction sets were 0.75, 0.73, and 0.56 respectively. These values would indicate that the KSOM technique is not much better than random set generation. As mentioned above, R^2 is not always reliable for an unevenly distributed dataset such as the one used in this study and as a result we feel that the RMSE values provide a better indicator of the goodness of a model. Though the RMSE's for the training and cross-validation sets are comparable, the prediction set RMSE is much larger for the random sets. In addition, comparing the standard deviation in the RMSE values for the five runs for the random and KSOM sets indicates that the KSOM technique is more consistent compared to the random approach. For the original best model the standard deviations for the three sets were 0.005, 0.01 and 0.02, respectively. For the random sets the standard deviations were 0.02, 0.03 and 0.13 respectively, indicating that predictions made using the random sets were not consistent over several runs. Once again, we believe that this is evidence for the KSOM's ability to generate good sets based on features of the dataset.

The next randomization test consisted of regenerating the best nonlinear model (using the ADAPT descriptors as reported in Table 4.4) but scrambling the dependent variable. With the scrambled dependent variable, the best CNN model was regenerated using the original QSAR sets. This process was repeated five times, each time scrambling the dependent variable and the average RMSE and R^2 values for the training, cross-validation and prediction sets for the five runs were noted. It would be expected that the resultant model would have relatively high RMSE's for the three sets, as well as low R^2 values. This was indeed the case with the training, cross-validation and prediction sets having RMSE values of 1.04, 1.00 and 0.97 respectively (Table 4.9). In addition

the R^2 values for the three sets were 0.17, 0.09 and 0.01, respectively. Compared to the RMSE and R^2 values for the best model, it appears that chance correlations played little (if any) part in the results for the best model.

Finally a randomization test was carried out to investigate the role of chance correlations in the genetic algorithm (i.e., the descriptor selection algorithm). This was carried out by generating one hundred CNN models using a 6–5–1 architecture and the QSAR sets generated by the SOM (using the MoRSE-WHIM Dragon descriptor subset). However in each run, six ADAPT descriptors were randomly selected from the reduced pool. One would assume that the RMSE and R^2 values for the models generated by randomly selecting descriptors would be worse than for the best reported model but not as poor compared to the runs using a scrambled dependent variable. This can be explained by noting that since the dependent variable is not scrambled there will be some correlation with the descriptors selected. However due to the fact that we randomly select descriptors this correlation will not be as significant compared to descriptor selection using a genetic algorithm, which looks for descriptor subsets that are well correlated with the dependent variable and hence produce models with low cost functions. Thus this test ensures that the specific set of descriptors selected by the genetic algorithm did not arise by chance alone. The results for this test are provided in Table 4.10. As can be seen, the average RMSE for all three sets are higher than those reported for the best model, though the differences are not as significant compared to the results from the scrambled dependent variable test. The R^2 values are also lower than for the best reported model but are not as poor when compared to the results from the scrambled dependent variable test.

The results from the randomization tests described above thus indicate that chance correlations played little (if any) role in both the descriptor selection algorithm as well in the final model itself.

4.7 Diversity Indices and SOM Generated Sets

The SOM was used to prepare training and prediction sets such that they would be heterogeneous in nature and representative of the whole dataset. The molecular dataset diversity index[26] has been developed to quantify the diversity of a dataset and the correspondence between training and prediction sets. This metric provides a quantitative estimate of the similarity between the training and prediction sets. Golbraikh describes three quantities - $M_{(\text{test, train})}$, $M_{(\text{train,test})}$ and I_{train}. The quantity of interest here is

$M_{(test,train)}$, which measures the diversity of the training set with respect to the prediction set. The value of $M_{(test,train)}$ depends on both the algorithm used to generate sets as well as the distribution of the data set in the descriptor space. In general lower values of $M_{(test,train)}$ indicate that the points in the prediction set are closer (or correspond better) to the points in the training set. However the evaluation of $M_{(test,\ train)}$ depends on the value of an arbitrary value termed the Dissimilarity Level (DL). Golbraikh does not go into detail regarding the choice of a dissimilarity level. Hence, we calculated $M_{(test,\ train)}$ values at increasing DL values for each Dragon descriptor combination, plotted them (Fig. 4.8), and correlated the behavior of the plots with the CNN model statistics. One would expect that for training and prediction sets which correspond well with each other (i.e., a prediction set point corresponds to some training set point) the $M_{(test,train)}$ should rapidly fall to zero with increasing DL values. However, another view would be to consider the training and prediction sets to be well distributed throughout descriptor space of the dataset. In such a case the correspondence between the two sets would not necessarily be very good and one would observe higher values of $M_{(test,train)}$ for a given DL value. This might lead one to conclude that such a situation would lead to bad model statistics. However, Fig. 4.8 indicates otherwise. From the plot we see that the curves for the MoRSE-2D Autocorrelation and the MoRSE-WHIM combinations remain constant at an $M_{(test,train)}$ value of 1 for all DL values up to approximately 2 and the MoRSE-GETAWAY combination remains at 1 up to nearly 2.5. From Table 4.3 we see that the MoRSE-GETAWAY combination has the best training and cross-validation set errors of all the sets tested, but its prediction set error is higher. At the same time, the training and prediction set errors for the MoRSE-WHIM and MoRSE-2D Autocorrelation combinations are larger than for the MoRSE-GETAWAY combination - but their order follows the trend in the graph. Sets that remain at a $M_{(train,test)}$ value of 1 for higher DL values appear to lead to lower RMSE's for the training and cross-validation sets.

If one considers the prediction set errors, a similar trend is seen. Sets whose $M_{(test,train)}$ vs. DL plots remain at a $M_{(test,train)}$ value of 1 for larger values of DL appear to lead to better prediction set errors. However this should not be considered as an absolute as the plot for the GETAWAY set does not follow this trend. In fact the prediction set error is equal to that for the MoRSE-WHIM set., but the $M_{(train,test)}$ value drops below 1 for DL values of 0.7 onwards. Thus the values of $M_{(test,train)}$ for the GETAWAY set are lower for a given DL value, indicating a better correspondence between the training and prediction sets. However, though this leads to a good prediction set error, the training and cross-validation set errors are quite large. This could imply

that lower $M_{(test,train)}$ values might lead to better prediction set errors but at the same time would lead to a loss of generalizability as evidenced by the training and cross-validation set errors.

Though the use of an arbitrary DL value in the evaluation of $M_{(test,train)}$ values does make interpretation of $M_{(test,train)}$ values slightly ambiguous, we feel that the technique we describe does provide some indication as to whether a training set might lead to good training and prediction set errors, based on diversity index information.

4.8 Conclusions

This study used a Kohonen self-organizing map to investigate whether a similarity based set generation method would lead to better QSAR models. Multiple runs using different sets of Dragon descriptors were used to generate training, cross-validation, and prediction sets, which were in turn used to create QSAR models. The best model obtained by this method did improve upon the previously published model in terms of model size. Although the actual RMSE values were not significantly better than those published, they were consistent and exhibited a lower standard deviation over the three QSAR sets compared to the original results. QSAR sets were also generated using a sphere exclusion[19] technique. Models generated using these QSAR sets did not show any significant improvement in terms of statistics or model size over the published results. When compared to the models generated using QSAR sets created by the SOM, we noted that there was no significant improvement in the statistics of the models generated by the sphere exclusion methods. Furthermore, the RMSE's of the three QSAR sets generated by the sphere exclusion method were not as consistent as those generated by the SOM and exhibited standard deviations similar to those of the original QSAR sets obtained by activity binning. However, the SOM did lead to models that were significantly simpler than those generated using the sphere exclusion method or activity binning (the published results). Randomization tests indicated that the models generated did not arise due to chance correlations. The use of the $M_{(test,train)}$ diversity index provided an indication of the Dragon descriptor set's ability to generate good QSAR sets which in turn lead to QSAR models.

Though the study did lead to a better model than that published, it involved a number of arbitrary decisions such as the choice of initial descriptors to submit to the SOM as well as choosing a specific SOM split out of several runs. The algorithm could be substantially improved by implementing a method to optimize the threshold value so

that classification of molecules in the SOM could be automated. Another improvement would be in the choice of initial descriptors. Since this study was exploratory in nature, we restricted ourselves to certain subsets of Dragon descriptors which we deemed to be holistic in nature. That is, we chose Dragon descriptor sets that appeared to characterize the whole molecule, rather than characterizing specific molecular features. In addition the choice of Dragon descriptors was also guided by the fact that we did not want to use ADAPT descriptors during the initial classification process. Clearly, there remains an element of arbitrariness in the selection of Dragon descriptor sets. This need not be the case and by including more or even all Dragon descriptors (followed by a PCA to obtain the main contributing components) the initial classification might be better. In addition though the evaluation of $M_{(\text{test,train})}$ does involve an arbitrary constant, it seems that looking at the trend rather than individual values (for fixed DL values) can be used to make a decision on which Dragon sets could be used for further study.

Table 4.1. Type and number of Dragon descriptors used by the SOM to generate training, cross-validation, and prediction sets for QSAR models.

Descriptor Name	No. of Descriptors	References
BCUT	123	30–32
BCUT & Galvez Topological Indices	63	30–32, 40–42
GETAWAY	128	43–45
MoRSE & 2D Auto Correlation	173	35–39, 46, 47
MoRSE & GETAWAY	223	43–47
MoRSE & WHIM	139	46–53

Table 4.2. Summary of the number of molecules present in the training, cross-validation and prediction sets. The sizes of these sets were the same for all the Dragon descriptor subsets investigated.

Set	Number of Molecules	Percentage of Molecules
Training	267	80.1
Cross-Validation	32	9.6
Prediction	34	10.3
Total	333	100

Table 4.3. Summary of the nonlinear CNN models using training, cross-validation, and prediction sets created by the SOM and Dragon descriptor combinations.

		RMSE			R^2		
Dragon Descriptor	CNN Arch.	TSET	CVSET	PSET	TSET	CVSET	PSET
BCUT - 2D Autocorrelation	5–3–1	0.63	0.68	0.79	0.68	0.60	0.67
BCUT - Galvez Topological Indices	5–3–1	0.62	0.62	0.71	0.69	0.66	0.64
GETAWAY	5–2–1	0.68	0.60	0.73	0.64	0.76	0.67
MoRSE - 2D Autocorrelation	5–3–1	0.63	0.63	0.68	0.68	0.60	0.74
MoRSE -GETAWAY	9–5–1	0.49	0.59	0.76	0.80	0.58	0.80
MoRSE - WHIM	6–5–1	0.60	0.61	0.65	0.75	0.78	0.64
Published Results[54]	10–6–1	0.45	0.49	0.66	0.84	0.78	0.64

Table 4.4. ADAPT descriptors present in the two best nonlinear CNN models.

MoRSE & 2D Autocorrelation			MoRSE & WHIM		
Descriptor	Type	Range	Descriptor	Type	Range
N7CH	Topo	7.0 - 28.0	V6P7	Topo	2.1 - 0.5
MOLC-8	Topo	0.6 - 2.8	WTPT-4	Topo	0.0 - 12.2
NDB-13	Topo	0.0 - 7.0	N7CH	Topo	7.0 - 28.0
NAB-15	Topo	6.0 - 23.0	NDB-13	Topo	0.0 - 7.0
WPSA-3	Hybrid	17 - 57.4	MDE-23	Topo	0.0 - 28.1
			RPCS	Hybrid	0.0 - 8.1

Topo indicates a topological descriptor. N7CH, number of seventh order chains index;[55–57] MOLC-8, average distance sum connectivity[58,59] (topological index J); NDB-13, number of double bonds; NAB-15, number of aromatic bonds; WPSA-3, partial positive surface area multiplied by the total molecular surface area divided by 1000;[60] RPCS, relative positive charged surface area;[60] MDE-23, molecular distance edge between primary and secondary carbons;[61] WTPT-4, sum of atom ID's for oxygens[62]

Table 4.5. Comparison of R^2 values for the training, cross-validation, and prediction sets created by the SOM using Dragon descriptors.*

	Training Set	Cross-validation Set	Prediction Set
MoRSE - 2D Autocorrelation	0.68	0.60	0.64
MoRSE - WHIM	0.75	0.78	0.67
Published[54]	0.83	0.78	0.64

* The models produced were CNN models. See Table 4.3 for the model architectures.

Table 4.6. A summary of the RMSE's for the 10–6–1 nonlinear CNN models using five QSAR sets generated by activity binning.

Serial No.	Training Set	Cross-validation Set	Prediction Set
1	0.45	0.59	0.81
2	0.45	0.52	0.73
3	0.44	0.63	0.95
4	0.64	0.64	1.00
5	0.67	0.61	0.95

Table 4.7. Summary of the best nonlinear CNN models generated from QSAR sets created using the sphere exclusion algorithm.

		RMSE			R^2		
Dragon Descriptor*	CNN Arch.	TSET	CVSET	PSET	TSET	CVSET	PSET
BCUT - 2D	9–3–1	0.55	0.54	0.87	0.74	0.78	0.33
Autocorrelation							
BCUT - Galvez	9–8–1	0.46	0.50	0.87	0.83	0.81	0.36
Topological Indices							
GETAWAY	8–5–1	0.56	0.56	0.63	0.75	0.80	0.67
MoRSE - 2D	9–8–1	0.49	0.53	0.68	0.81	0.82	0.68
Autocorrelation							
MoRSE -	8–6–1	0.52	0.58	0.64	0.79	0.84	0.67
GETAWAY							
MoRSE - WHIM	7–6–1	0.50	0.57	0.82	0.80	0.77	0.52
Published Results[54]	10–6–1	0.45	0.49	0.66	0.84	0.78	0.64

* The external descriptor set used by the sphere exclusion algorithm to create the training and prediction sets

Table 4.8. Comparison of statistics for training, cross-validation, and prediction sets generated randomly versus sets created by the SOM using the MoRSE-WHIM Dragon descriptor combination.*

	Random Sets			MoRSE - WHIM Sets		
	Mean RMSE	Std. dev.	Mean R^2	Mean RMSE	Std. dev	R^2
TSET	0.57	0.02	0.75	0.58	0.005	0.74
CVSET	0.59	0.03	0.73	0.57	0.010	0.76
PSET	0.80	0.13	0.56	0.63	0.020	0.63

* The statistics are from a nonlinear CNN model using a 6–5–1 architecture. The same descriptors were used in both models.

Table 4.9. RMSE values for a nonlinear CNN Model* using a scrambled dependent variable using training, cross-validation, and predictions sets created by the KSOM using the MoRSE-WHIM Dragon descriptor combination.

	Scrambled		Original	
	Mean RMSE	Mean R^2	Mean RMSE	R^2
TSET	1.04	0.17	0.58	0.74
CVSET	1.00	0.09	0.56	0.76
PSET	0.97	0.01	0.59	0.63

* The model was generated using a 6–5–1 CNN architecture and the ADAPT descriptors reported for the best nonlinear model.

Table 4.10. A summary of the RMS errors and R^2 values for one hundred runs of the best CNN architecture (6–5–1) using randomly selected ADAPT descriptors.*

	Mean RMSE	Std. Dev.	Mean R^2	Std. Dev.
Training Set	0.81	0.09	0.47	0.11
Cross-Validation Set	0.84	0.08	0.36	0.13
Prediction Set	0.84	0.09	0.28	0.13

* QSAR sets used in these models were created by the KSOM using the MoRSE-WHIM Dragon descriptor combination.

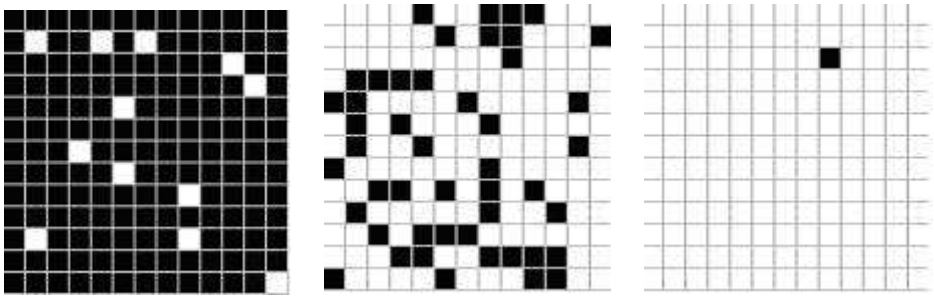


Fig. 4.1. A graphical representation of the SOM after the cluster detection step using the BCUT & 2D-Autocorrelation Dragon[29] descriptor subset. Black and white squares represent the individual classes. Grids A, B and C were obtained by setting the threshold value to 0, 1.2 and 3.6 respectively. Grid B was used to generate the final QSAR sets for this Dragon descriptor subset.

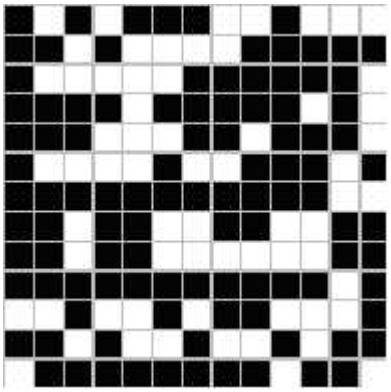


Fig. 4.2. A graphical representation of the distribution of whole dataset on the grid after it has been divided into two classes based on the BCUT & 2D-Autocorrelation Dragon[29] descriptor combination. Black and white squares represent the two different classes.

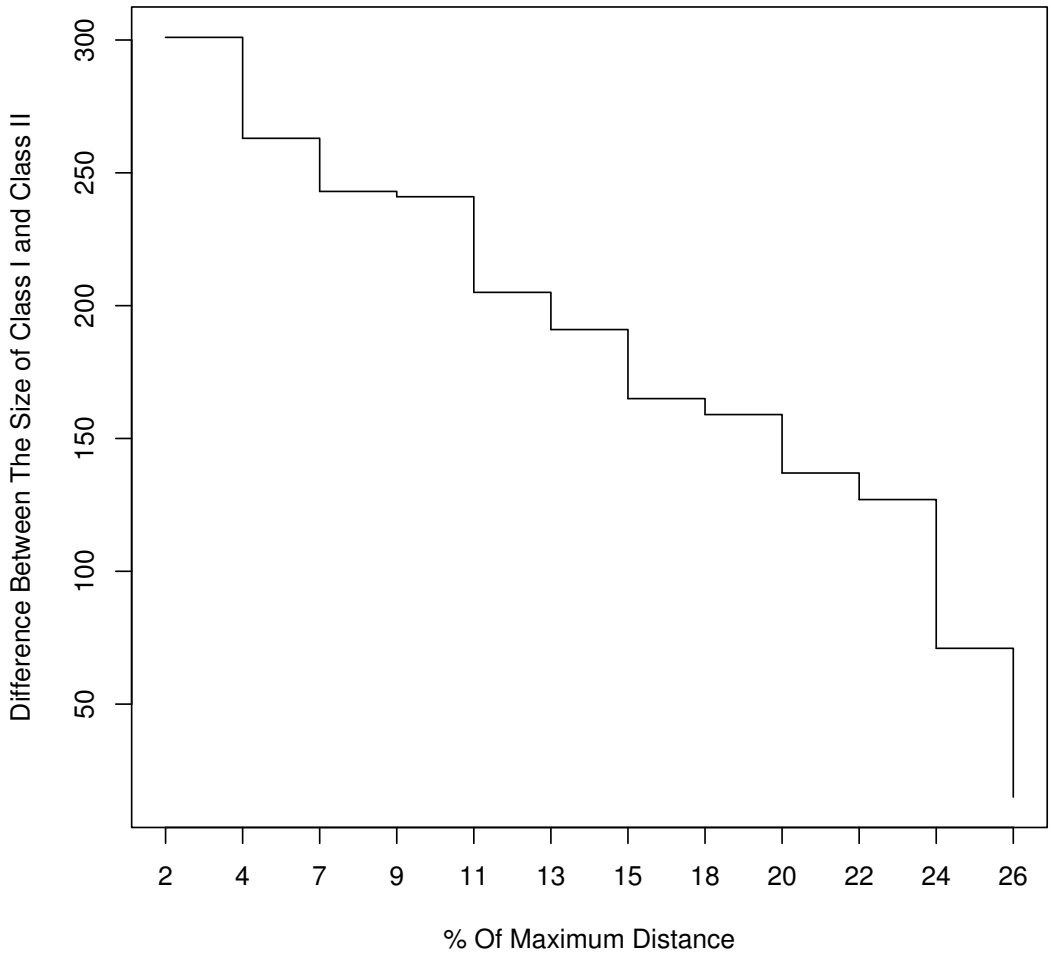


Fig. 4.3. A plot showing the variation of D (the difference in size between major and minor SOM classes) versus the threshold value for the SOM. In this plot the threshold value is represented as a percentage of the maximum distance in the grid for a SOM in which the threshold value was set to 0. The descriptor set used to generate the grids described in the plot was the MoRSE-WHIM Dragon[29] descriptor subset.

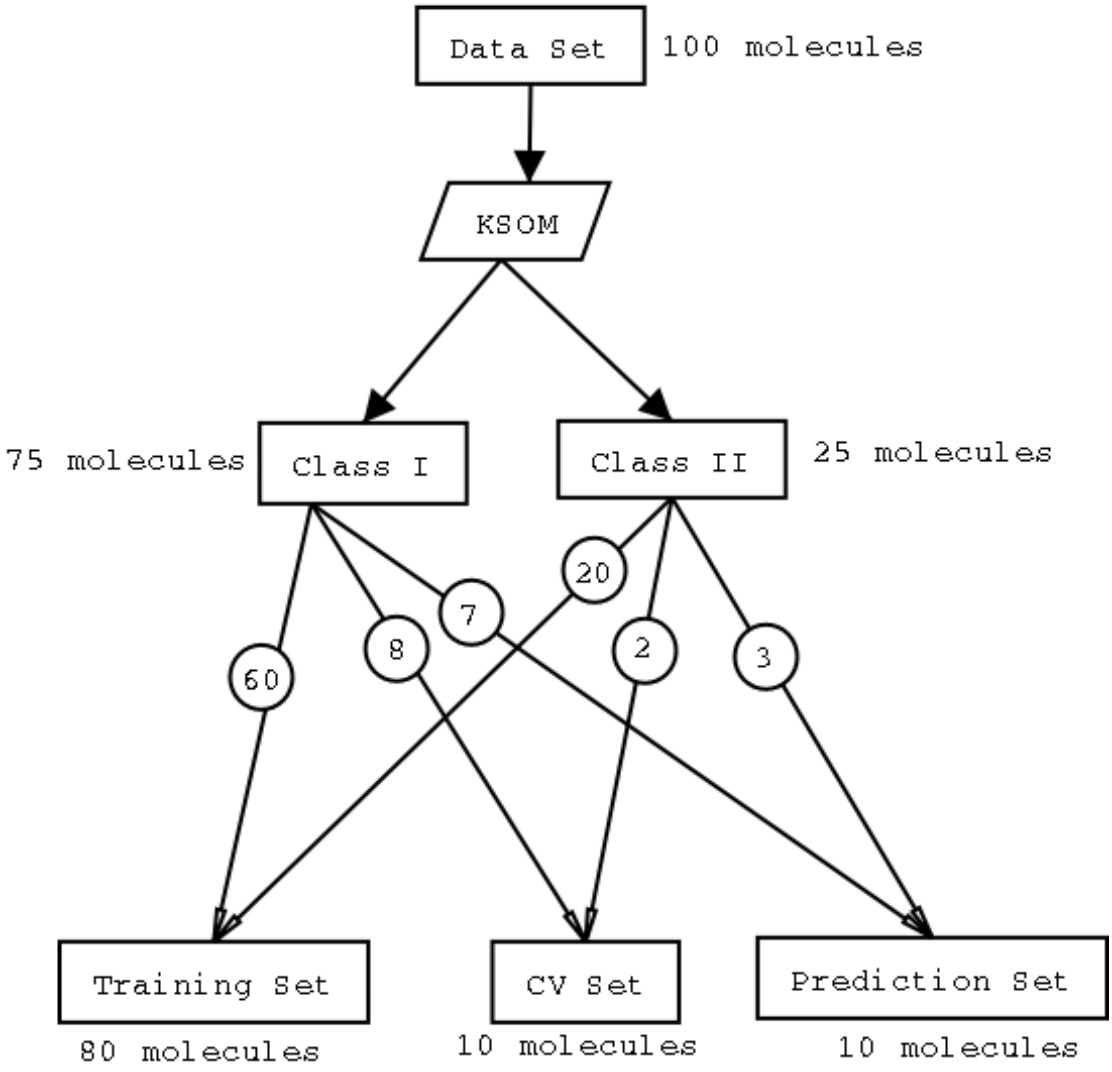


Fig. 4.4. A diagrammatic representation of the method we use to generate QSAR sets from the SOM classification of the whole dataset. The numbers within circles are the number of molecules from that class that present in the specific QSAR set.

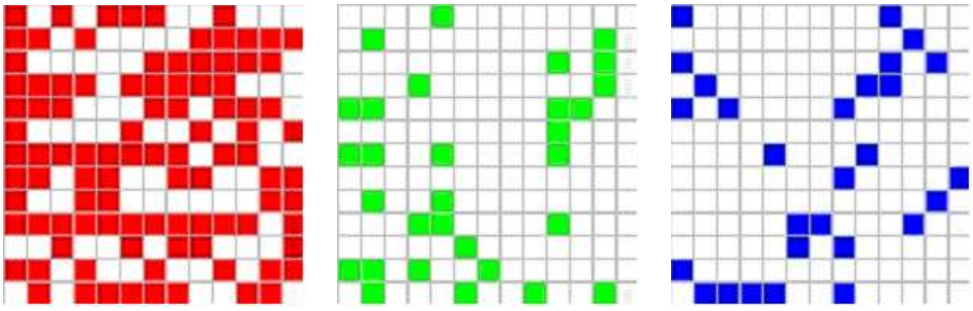


Fig. 4.5. The three diagrams represent the distribution of the QSAR sets over the surface of the SOM. The grid was trained with the BCUT & 2D-Autocorrelation Dragon[29] descriptor combination.

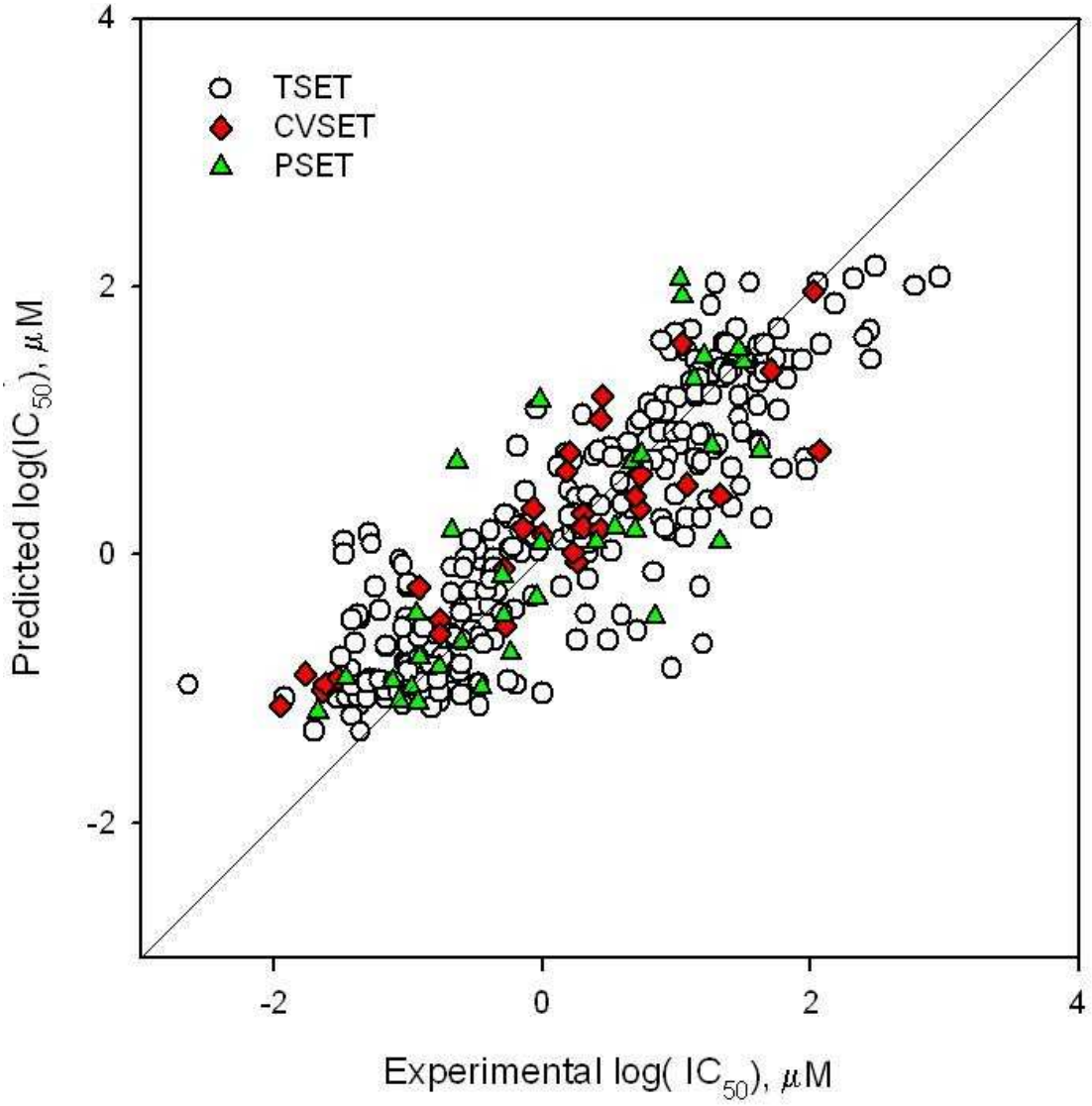
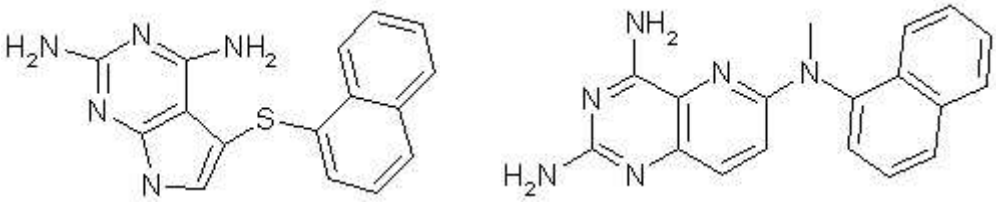


Fig. 4.6. Plot of experimental vs. predicted log IC_{50} for the 6–5–1 CNN model Generated Using Training, Cross-validation, and Prediction Sets Created Using the SOM and MoRSE - WHIM Dragon[29] Descriptor Combination.



Outlier detected by best CNN model in this study

Published outlier for the best pcDHFR model[54]

Fig. 4.7. Prediction set outliers.

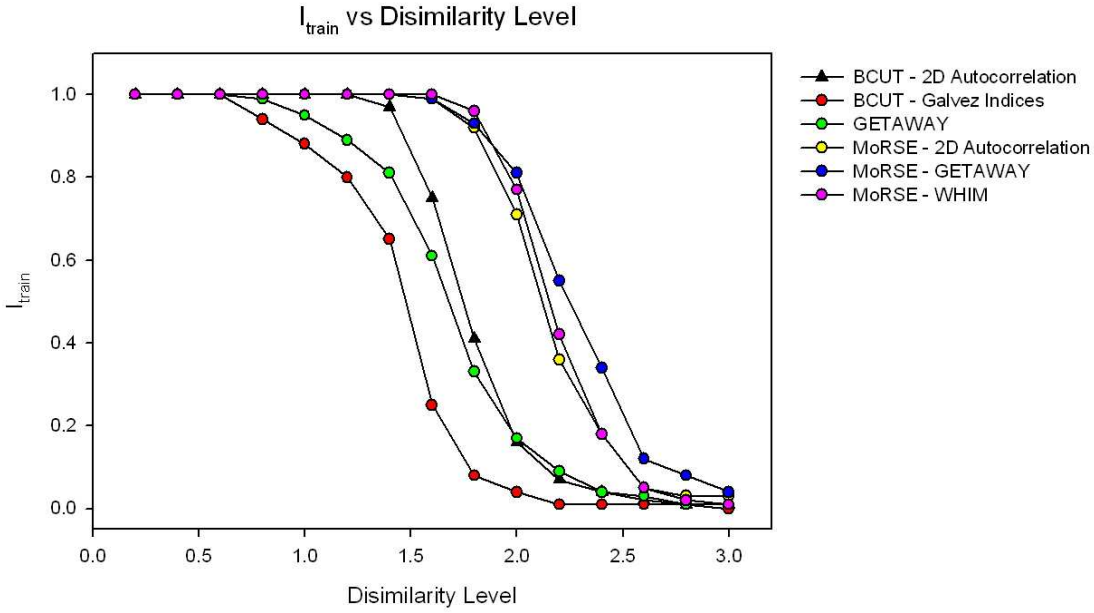


Fig. 4.8. Plot of dissimilarity level vs. $M_{(test,train)}$ for the various Dragon[29] sets studied.

References

[1] Kohonen, T. *Self Organizing Maps;* volume 30 of *Springer Series in Information Sciences* Springer: Berlin, 1994.

[2] Janet, J.; Gutierrez, R.; Chase, T.; White, M.; Sutton, J. Autonomous Mobile Robot Global Self Localization Using Kohonen and Region Feature Neural Networks. *Journal of Robotic Systems* **1997,** *14,* 263–282.

[3] Naim, A.; Ratnatunga, K.; Griffiths, R. Galaxy Morphology Without Classification: Self Organizing Maps. *Astrophysical Journal Supplement Series* **1997,** *111,* 357-367.

[4] Gasteiger, J.; Zupan, J. Neural Networks in Chemistry. *Angew. Chem. Int. Ed. Engl.* **1993,** *32,* 503–527.

[5] Daniel, N.; Lewis, I.; Griffiths, P. Interpretation of Raman Spectra of Nitro Containing Explosive Materials. Part II: the Implementation of Neural, Fuzzy and Statistical Models for Unsupervised Pattern Recognition.. *Appl. Spectr.* **1997,** *51,* 1854–1867.

[6] Vander Heyden, Y.; Vankeerbergghen, P.; Novic, M.; Zupan, J.; Massart, D. The Application of Kohonen Neural Netoworks to Diagnose Calibration Problems in Atomic Absorption Spectroscopy. *Talanta* **2000,** *51,* 455–466.

[7] Novic, M.; Zupan, J. Investigation of Infra-Red Spectra-Structure Correlation Using Kohonen and Counterpropagation Neural Networks. *J. Chem. Inf. Comput. Sci.* **1995,** *35,* 454-466.

[8] Wu, W.; Walczak, B.; Massart, D.; Heuerding, S.; Erni, F.; Last, I.; Prebble, K. Artificial Neural Networks in Classification of NIR Spectral Data: Design of the Training Set. *Chemometrics and Intelligent Laboratory Systems* **1996,** *33,* 35–46.

[9] Aires-de Sousa, J.; Hemmer, M. C.; Gasteiger, J. Prediction of 1H NMR Chemical Shifts Using Neural Networks. *Anal. Chem.* **2002,** *74,* 80–90.

[10] Satoh, H.; Sacher, O.; Nakata, T.; Chen, L.; Gasteiger, J.; Funatsu, K. Classification of Organic Reactions: Similarity of Reactions Based on the Electronic Features of Oxygen Atoms at the Reaction Sites. *J. Chem. Inf. Comput. Sci.* **1998,** *38,* 210–219.

[11] Chen, L.; Gasteiger, J. Knowledge Discovery in Reaction Databases: Landscaping Organic Reactions By a Self Organizing Neural Network. *J. Am. Chem. Soc.* **1997,** *119,* 4033–4042.

[12] Manallack, D.; Livingstone, D. Neural Networks in Drug Discovery: Have They Lived Upto Their Promise?. *Eur. J. Med. Chem.* **1999,** *34,* 195–208.

[13] Tetko, I.; Kovalishyn, V.; Livingstone, D. Volume Learning Algorithm Artificial Neural Networks for 3D QSAR Studies. *J. Med. Chem.* **2001,** *44,* 2411-2420.

[14] Bienfait, B. Applications of High Reolution Self Organizing Maps to Retrosynthetic and QSAR Analysis. *J. Chem. Inf. Comput. Sci.* **1994,** *34,* 890–898.

[15] Rose, V.; Croall, I.; Macfie, H. An Application of Unsupervised Neural Network Methodology Kohonen Topology-Preserving Mapping to QSAR Analysis. *Quant. Struct.-Act. Relat.* **1991,** *10,* 6–15.

[16] Anzali, S.; Barnickel, G.; Krug, M.; Sadowski, J.; Wagener, M.; Gasteiger, J.; Polanski, J. 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. Comp. Aid. Molec. Des.* **1996,** *10,* 521–534.

[17] Gramatica, P.; Consonni, V.; Todeschini, R. QSAR Study of the Tropospheric Degradation of Organic Compounds. *Chemosphere* **1999,** *38,* 1371–1378.

[18] Espinosa, G.; Arenas, A.; Giralt, F. An Integrated SOM Fuzzy ARTMAP Neural System for the Evaluation of Toxicity. *J. Chem. Inf. Comput. Sci.* **2002,** *42,* 343–359.

[19] Golbraikh, A.; Tropsha, A. Predicitve QSAR Modeling Based on Diversity Sampling of Experimental Datasets for the Test and Training Set Selection. *J. Comp. Aid. Molec. Des.* **2002,** *16,* 356–369.

[20] Kocjancic, R.; Zupan, J. Modelling of the River Flowrate: The Influence of Training Set Selection.. *Chemometrics and Intelligent Laboratory Systems* **2000,** *54,* 21–34.

[21] Kirew, D.; Chretien, J.; Bernard, P.; Ros, F. Application of Kohonen Neural Networks in Classification of Biologically Active Compounds. *SAR and QSAR in Environmental Research* **1998,** *8,* 93.

[22] Yan, A.; Gasteiger, J. Prediction of Aqueous Solubility of Organic Compounds Based on 3D Structure Representation. *J. Chem. Inf. Comput. Sci.* **2003,** *43,* 429-434.

[23] Andersson, P.; Sjostrom, M.; Wold, S.; Lundstedt, T. Strategies for Subset Selection of Parts of an In House Chemical Library. *J. Chemom.* **2001,** *15,* 353–369.

[24] Linusson, A.; Gottfries, J.; Olsson, T.; Ornskov, E.; Folestad, S.; Norden, B.; Wold, S. Statistcal Molecular Design, Parallel Synthesis and Biological Evaluation of a Library of Thrombin Inhibitors. *J. Med. Chem.* **2001,** *44,* 3424–3439.

[25] Linusson, A.; Gottfries, J.; Lindgren, F.; Wold, S. Statistical Molecular Design of Building Blocks For Combinatorial Chemistry. *J. Med. Chem.* **2000,** *43,* 1320–1328.

[26] Golbraikh, A. Molecular Dataset Diversity Indices and Their Applications to Comparison of Chemical Databases and QSAR Analysis. *J. Chem. Inf. Comput. Sci.* **2000,** *40,* 414–425.

[27] Jurs, P.; Chou, J.; Yuan, M. Computer Assisted Drug Design. In ; American Chemical Society: Washington D.C., 1979; Chapter Studies of Chemical Structure Biological Activity Relations Using Pattern Recognition.

[28] Stuper, A.; Brugger, W.; Jurs, P. *Computer Assisted Studies of Chemical Structure and Biological Function;* Wiley: New York, 1979.

[29] Todeschini, R.; Consonni, V.; Pavan, M. "DRAGON", 2005.

[30] Burden, F. Molecular Identification Number for Substructure Searches. *J. Chem. Inf. Comput. Sci.* **1989,** *29,* 225–227.

[31] Burden, F. A Chemically Intuitive Molecular Index Based on the Eigenvalues of a Modified Adjacency Matrix. *Quant. Struct.-Act. Relat.* **1997,** *16,* 309–314.

[32] Pearlman, R.; Smith, K. Metric Validation and the Receptor-Relevant Subspace Concept. *J. Chem. Inf. Comput. Sci.* **1999,** *39,* 28–35.

[33] Stanton, D. T. Evaluation and Use of BCUT Descriptors in QSAR and QSPR Studies. *J. Chem. Inf. Comput. Sci.* **1999,** *39,* 11–20.

[34] Todeschini, R.; Consonni, V. *Handbook of Molecular Descriptors;* Wiley-VCH: Berlin, 2002.

[35] Broto, P.; Moreau, G.; Vandycke, C. Molecular Structures: Perception, Autocorrelation Descriptor and SAR Studies. *Eur. J. Med. Chem.* **1984,** *19,* 66–70.

[36] Moreau, G.; Broto, P. Autocorrelation of Molecular Structures: Application to SAR Studies. *Nouv. J. Chim.* **1980,** *4,* 757-764.

[37] Moreau, G.; Broto, P.; Fortin, M.; Turpin, C. Computer Conducted Screening of Molecular Structures of Potentially Anxiolytic Substances Using an Autocorrelation Technique. *Eur. J. Med. Chem.* **1988,** *23,* 275–281.

[38] Moran, P. Notes on Continous Stochastic Phenonema. *Biometrika* **1950,** *37,* 17–23.

[39] Geary, R. The Contiguity Ratio and Statistical Mapping. *Incorp. Statist.* **1954,** *5,* 115–145.

[40] Galvez, J.; Garcia, R.; Salabert, M.; Soler, R. Charge Indexes. New Topological Descriptors. *J. Chem. Inf. Comput. Sci.* **1994,** *34,* 520–525.

[41] Galvez, J.; Garcia-Domenech, R.; de Gregorio Alapont, C.; De Julian Ortiz, V.; Popa, L. Pharmacological Distribution Diagrams: A Tool for De Novo Drug Design. *J. Mol. Graphics* **1996,** *14,* 272–276.

[42] Galvez, J.; Garcia-Domenech, R.; De Julian Ortiz, V.; Soler, R. Topological Approach to Drug Design. *J. Chem. Inf. Comput. Sci.* **1995,** *35,* 272–284.

[43] Consonni, V.; Todeschini, R. *Rational Approaches to Drug Design;* Prous Science: Barcelona, 2001.

[44] Consonni, V.; Todeschini, R.; Pavan, M. Structure-Response Correlations and Similarity/Diversity Analysis By GETAWAY Descriptors. Part 1. Theory of the Novel 3D Molecular Descriptors. *J. Chem. Inf. Comput. Sci.* **2002,** *42,* 682–692.

[45] Consonni, V.; Todeschini, R.; Pavan, M.; Gramatica, P. Structure/Response Correlations and Similarity/Diversity Analysis By GETAWAY Descriptors. Part 2. Application of the Novel 3D Molecular Descriptors to QSAR/QSPR Studies. *J. Chem. Inf. Comput. Sci.* **2002,** *42,* 693–705.

[46] Schuur, J. H.; Selzer, P.; Gasteiger, J. 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.* **1996,** *36,* 334–344.

[47] Gasteiger, J.; Sadowski, J.; Selzer, P.; Steinhauer, L.; Steinhauer, V. Chemical Information in 3D Space. *J. Chem. Inf. Comput. Sci.* **1996,** *36,* 1030–1037.

[48] Todeschini, R.; Lasagni, M.; Marengo, E. New Molecular Descriptors for 2D and 3D Structures - Theory. *J. Chemom.* **1994,** *8,* 263–273.

[49] Todeschini, R.; Gramatica, P. 3D Modelling and Prediction by WHIM Descriptors. Part 5. Theory Development and Chemical Meaning of WHIM Descriptors. *Quant. Struct.-Act. Relat.* **1997,** *16,* 113–119.

[50] Todeschini, R.; Gramatica, P. 3D-modelling and prediction by WHIM descriptors. Part 6. Applications of the WHIM descriptors in QSAR studies. *Quant. Struct.-Act. Relat.* **1997,** *16,* 120–125.

[51] Gramatica, P.; Corradi, M.; Consonni, V. Modelling and Prediction of Soil Sorption Coefficients of Non-Ionic Organic Pesticides by Different Sets of Molecular Descriptors. *Chemosphere* **2000,** *41,* 763–777.

[52] Todeschini, R.; Gramatica, P.; Marengo, E.; Provenzani, R. Modeling and Prediction by Using WHIM Descriptors in QSAR Studies: Submitochondrial Particles (SMP) as Toxicity Biosensors of Chlorophenols. *Chemosphere* **1995,** *33,* 71–79.

[53] Todeschini, R.; Vighi, M.; Provenzani, R.; Finzio, A.; Gramatica, P. Modeling and Prediction By Using WHIM Descriptors in QSAR Studies: Toxicity of Heterogeneous Chemicals on Daphnia Magna. *Chemosphere* **1996,** *32,* 1527–1545.

[54] Mattioni, B.; Jurs, P. Prediction of Dihydrofolate Reductase Inhibition and Selectivity Using Computational Neural Networks and Linear Discriminant Analysis. *J. Molec. Graph. Model.* **2003,** *21,* 391–419.

[55] Kier, L.; Hall, L. Molecular Connectivity VII: Specific Treatment to Heteroatoms. *J. Pharm. Sci.* **1976,** *65,* 1806–1809.

[56] Kier, L.; Hall, L. *Molecular Connectivity in Structure Activity Analysis;* Research Studies Press Ltd., John Wiley and Sons: Hertfordshire, England, 1986.

[57] Kier, L.; Hall, L. Molecular Connectivity I: Relationship to Local Anasthesia. *J. Pharm. Sci.* **1975,** *64,* 1971–1974.

[58] Kier, L.; Hall, L. *Molecular Connectivity in Chemistry and Drug Research;* Academic Press: New York, 1976.

[59] Balaban, A. Higly Discriminating Distance Based Topological Index. *Chem. Phys. Lett.* **1982,** *89,* 399–404.

[60] Stanton, D. T.; Jurs, P. C. Development and Use of Charged Partial Surface Area Structural Descriptors in Computer Assissted Quantitative Structure Property Relationship Studies. *Anal. Chem.* **1990,** *62,* 2323–2329.

[61] Liu, S.; Cao, C.; Li, Z. Approach to Estimation and Prediction for Normal Boiling Point (NBP) of Alkanes Based on a Novel Molecular Distance Edge (MDE) Vector, λ. *J. Chem. Inf. Comput. Sci.* **1998,** *38,* 387–394.

[62] Randic, M. On Molecular Identification Numbers. *J. Chem. Inf. Comput. Sci.* **1984,** *24,* 164–175.

Chapter 5

Determining the Validity of a QSAR Model: A Classification Approach

5.1 Introduction

Quantitative structure-activity relationship (QSAR) modeling is based on the construction of predictive models using a set of known molecules and associated activity values. Such models can be generated using a wide variety of methods ranging from linear methods (e.g., linear regression and linear discriminant analysis) to non-linear methods (e.g., random forests and neural networks). As described in Chapter 3, an important step of the QSAR modeling process, irrespective of the nature of the modeling technique used, is validation. In all cases, the predictive ability of the models are tested with a set of molecules (the prediction set), which were not used during the model building process. Once a model has been built and validated it can be used on data for which no activity values are available. However, even though a model may have proved to exhibit good predictive ability based on the statistics for the prediction set, this is not always a guarantee that the model will perform well on a new set of data. The problem boils down to the fact that when a model is built and validated we can compare the predicted values to previously measured activity values. However, when the model is applied to new data, the predicted values of the activity cannot be compared with actual values. This leads to a problem: the training set and prediction set statistics may indicate that the model has good predictive ability. But when we use the model to predict values for molecules with unknown activity, how can we be sure that the predicted activity will be close to the actual activity? If the model were able to provide some measure of confidence for its prediction, this would be helpful. Such confidence measures (also known as scores) can be defined for various models. Examples include confidence bands for linear regression models and frequency based confidence measures for decision trees. However, such measures are specific to the modeling algorithm.

This work was published as Guha, R.; Jurs, P.C., "Determining the Validity of a QSAR Model - A Classification Approach", *J. Chem. Inf. Model.*, **2005**, *45*, 65–73.

This chapter describes a more general approach that should be applicable to any form of quantitative model. One possible approach is based on similarity. This is based on the assumption that a molecule that is structurally very similar (based on some sort of similarity metric such as atom pair[1] similarity or fingerprint similarity) to the training set molecules will be predicted well because the model has captured features that are common to the training set molecules and is able to find them in the new molecule. On the other hand, a new molecule with very little in common with the training set data should not be predicted very well; that is, the confidence in its prediction should be low.

An alternative approach to linking similarity measures and model quality (defined by residuals) is classification. In this method, the regression residuals for the training set are classified as good or bad, and a classification model is trained with the training set residuals. Once a trained model is obtained, we then predict the class of the prediction set residuals. However an important requirement for this process is that we be able to provide some measure of correctness for the predicted class assignments. Clearly this method does not fully solve the problem, as the classification algorithm would rarely be 100% correct. However the attractive feature of the approach discussed here is its generality. That is, it may be applied to any type of quantitative model, whether linear regression or a computational neural network. Furthermore, depending on how one defines a good residual or a bad residual, the classification model may be trained to detect unusual cases.

The fundamental decision that must be made when using the approach described in this chapter is the actual class assignments of the training set residuals. Since the fact that a compound is well predicted or poorly predicted is relatively subjective (except in extreme cases), the initial assignment of classes to the training set residuals is necessarily somewhat arbitrary. Furthermore, the nature of this class assignment defines the sizes of the two classes and hence plays a role in the choice of classification algorithm. These aspects are described in more detail in the following sections.

5.2 Datasets

Since one of the aims of this technique was generality, we attempted to test it on a variety of data. We considered three datasets covering both physical and biological properties. The first dataset was obtained from Goll et al.[2] and consisted of the boiling points of 277 molecules obtained from the Design Institute for Physical Property Data

(DIPPR) Project 801 database. This dataset is relatively diverse but contains several homologous series.

The second dataset consisted of a 179-compound subset of artemisinin analogues described by Avery et al.[3] We have previously developed and reported linear and CNN models based on this dataset.[4] The linear model from our previous study was used for the purposes of this work.

The third dataset consisted of 65 molecules. This dataset contained 56 molecules from a study carried out by Liu et al.[5] The remaining 9 molecules were selected from the literature, such that some were similar in structure to the molecules from Liu and some were distinctly different so as to be well-defined outliers in the final linear model. The molecules taken from Liu were all straight chain or branched hydrocarbons whereas the remaining molecules included polycyclic systems as well as molecules containing heteroatoms. The dependent variable in the original work was a transformation of the boiling point defined as

$$y = \log\left(266.7 - \mathrm{BP}\right) \tag{5.1}$$

where BP was the observed normal boiling points, in degrees Kelvin, of the molecules. Since the linear model we developed for this dataset did not use the all the molecules described by Liu, we did not use the logarithmic transformation and instead used boiling point values directly. The molecules and associated boiling points are shown in Table 5.1.

Each dataset was divided into a training and prediction set. The training set was used to build a linear model, and the prediction set was used to test the models themselves as well as the algorithms developed for this study. In the case of the artemisinin dataset, the same training and prediction sets that were used to develop the reported model were used in this study. Training and prediction sets for the DIPPR dataset were created using the activity binning method. In both cases, the training set contained approximately 80% of the whole dataset and the remainder was placed in the prediction set. The sets for the Liu dataset were created by hand. The training set contained 55 compounds selected from Liu, and the prediction set contained 10 compounds. Of these 10 compounds, one was taken from Liu and the remaining nine were selected from the literature. The reasoning for this specific construction was to allow the prediction set to contain molecules which were very dissimilar to the training set, so that the resultant linear model would exhibit distinct outliers.

5.3 Development of Linear Models

The first step of this study involved the development of a multiple linear regression model for each dataset. In the case of the artemisinin dataset, we used the linear model published by Guha et al.[4] This model contained four descriptors, and the statistics of the model are summarized in Table 5.2. Linear models for the DIPPR and Liu datasets were developed using the ADAPT[6,7] methodology described in Chapter 3. In all cases the final linear models were subjected to a PLS analysis to ensure that they were not overfitted. The statistics of the models for the DIPPR and Liu datasets selected by this procedure are summarized in Tables 5.3 and 5.4. Summary statistics for all three linear models are presented in Table 5.5.

5.4 The Classfication Approach

The aim of this study was to be able to decide whether a compound with unknown activity will be predicted well by a previously developed model. Though we focused only on linear regression models, the idea is general enough to be extended to other types of quantitative models (such as neural networks, support vector machines).

Initial attempts to develop a methodology to answer the above question focussed on evaluating a similarity measure between the new compound and the training set used to develop the existing model and then attempting to correlate the similarity measure with some measure of model quality. As we restricted ourselves to linear models we considered standard error of predictions and residuals. This line of attack did lead to the observation of some general trends. That is, compounds that were more similar to the training set generally exhibited smaller residuals and standard error of predictions. However, the observations were not conclusive, and the plots of the trends appeared to be too noisy to be able to draw any firm conclusions.

We then considered a classification approach. That is, can we classify a compound with no measured activity as *well predicted* or *poorly predicted* given a previously generated model and its associated training set? Our approach was to build a classification model using the original training set and the descriptors used in the original model and use this to predict the class of new compounds. The key word here is class. Before any model can be built we must decide how to classify the training set. We decided to consider regression residuals, as it would allow the technique to be generalized to other types of quantitative algorithms. The training set members were classified as bad or

good depending on whether their residuals were above or below a user specified cut-off value. This cut-off value plays a central role as it determines the size of the two classes. Our current strategy is to use a cut-off value obtained by visual inspection of a residual plot for the training set. The value of the cut-off was selected so that the minor class contained approximately 20% to 30% of the whole dataset. Clearly this leads to a highly imbalanced classification problem, but we felt that it would model a real world application of this technique more closely than allowing the classes to be of similar size. Alternative (non-arbitrary) methods include classifying training set members as good or bad depending on whether some regression diagnostic (Cooks distance, Mahalanobis distance) determined that it was an outlier. Fig. 5.1 shows the plots of residuals along with a line at the cut-off value for the three datasets studied here. Table 5.6 summarizes the cut-off values and associated class sizes for each dataset.

5.4.1 Classification Algorithms

Given the training set class structure, the choice of algorithm is guided by two requirements. First, the goal is to be able to test compounds for which we have no measured activity value. As a result, the classification algorithm must be able to produce some measure of confidence in its class predictions or else a probability of class membership (posterior probability). In the absence of such a quantity the final output of the classification model does not provide any more information than produced by simply processing the new compound through the original predictive model. The second requirement is that the algorithm must be able to handle unbalanced classes. In general, several schemes are available that can be used to modify the standard classification algorithms. These include over-sampling the minority class[8] and under-sampling the majority class.[9] Maloof[10] discusses the application of receiver operator characteristics (ROC) analysis in comparing how various sampling strategies and cost schemes affect classification of skewed datasets. Breiman[11] describes a simple method to increase the size of the dataset without simply repeating observations. The extra samples are termed convex pseudo-data and the generating algorithm requires a single parameter (as opposed to kernel density methods). We investigated the use of this method in attempt to improve classification accuracy.

We considered a wide variety of classification algorithms: logistic regression, partial least squares (PLS), discriminant analysis, neural networks and random forests. Random forests were first described by Breiman[12] and have been used in a variety of

QSAR applications. The original random forest algorithm was not suited for very unbalanced datasets, but current implementations[13] use a weighting scheme which overcomes this problem. We decided not to use this algorithm, due to the fact that it works with large descriptor pools owing to its ability to ignore irrelevant descriptors as well as the fact that the algorithm is resistant to overfitting. We did not want to build the classification models with more (and different) information than was available to the original regression models. Given that the good performance of random forest models is due to their ability to build trees based on good descriptor subsets, restricting the descriptor pool to four or five descriptors would probably result in lower quality random forest models.

In the case of discriminant analysis, we investigated the use of linear and quadratic discriminant methods. In each case, the algorithms employed were able to generate posterior probabilities via a cross-validation scheme. As the results were quite similar we only present the results of the linear discriminant analysis. All the algorithms mentioned above were obtained using the R software package.[14]

As mentioned in the previous section, we used the descriptors from the original regression model to build the classification model. We also investigated the use of a similarity measure as a source of extra information. Intuitively, one would expect that a new molecule that is similar to the molecules in the training set should be well predicted by the regression model. Thus, in addition to classification models built only with descriptors from the regression model we also built models that also contained a similarity value. We chose to use the atom pair similarity described by Carhart et al.[1] Atom pair similarities are calculated between pairs of molecules. To provide a single similarity value for each compound we calculated the average similarity value between each compound and all the compounds in the training set.

5.5 Results

Most of the algorithms exhibited good predictive ability considering the fact that the datasets used were not very large (especially the Liu dataset). As expected, the neural network performed very well, with a 90% correct prediction rate on the training set and 72% to 85% correct on the prediction set. The inclusion of the similarity values as a descriptor did not appear to improve the results significantly.

Table 5.7 shows the confusion matrices for the training and prediction sets generated using linear discriminant analysis for the artemisinin dataset. The implementation

used for this study allowed us to specify the prior probabilities for each class. We assumed that the priors could be approximated by the class proportions. Clearly, the very poor predictions for the minority class indicate the problem due to the imbalanced nature of the class distributions. To try and remove the bias due to the imbalanced nature of the problem, the model was regenerated with an over-sampled minority class. However the results did not improve significantly. To investigate whether extra information might improve the situation we also regenerated the model using the averaged atom pair similarity values as an extra independent variable. We felt that this was justified (as compared to using extra molecular descriptors) since this descriptor essentially compares the molecules amongst themselves. Table 5.7 displays the confusion matrices for the resultant model. The predictions for the good class are now 100% but members of the bad class are mispredicted in all cases. The results for this algorithm when applied to the DIPPR dataset give similar results. The classes assigned in this dataset are also quite unbalanced. The confusion matrices are presented in Table 5.8. The results for the Liu dataset (Table 5.9) are marginally better, more so for the prediction set than the training set. This is probably due to the slightly higher proportion of the minor class in the training set.

The results from the PLS classification scheme were not significantly better than those obtained with LDA and in some cases worse, and as a result we omit their presentation.

Table 5.10, 5.11 and 5.12 present the confusion matrices for the three datasets generated using a neural network. The network used entropy outputs[15] and thus provided the associated probabilities with each class assignment. In all cases, the inverse of the class proportions were used as example weights. Table 5.10 shows that the performance for the artemisinin dataset was not very impressive. However the imbalanced nature of the dataset does not affect the performance as much as in the case of LDA. In contrast, the DIPPR dataset showed very good performance using the neural network methodology as can be seen from Table 5.11. In this case, the bad class was very well predicted in both the training and prediction sets. Finally the Liu dataset also yielded good results (Table 5.12). In all cases, the use of average atom pair similarity as an extra independent variable did not appear to improve results.

Table 5.13 displays the weighted success rates for all the classification methods on all the datasets. This measure of classification success was described by Weston et

al.[16] and is defined as

$$w = \frac{1}{2}\left(\frac{\text{No. true positives}}{\text{Total positives}} + \frac{\text{No. true negatives}}{\text{No. negatives}}\right)$$

The above expression indicates that $0 \leq w \leq 1$. As mentioned by Weston, this measure is suitable for unbalanced classification problems. The values indicate the poor performance of the LDA (in fact, it appears to be not much better than random) and PLS methods and the much better performance of the neural network approach.

We also attempted to improve the classification results by using the convex pseudo-data method described by Breiman[11] to increase the size of the training sets. We considered two approaches. In the first method, we simply extended the whole dataset without regard to class. The new samples were placed in the training set and the extended training sets were used to build models. In the second approach we only extended the portion of the training set that was assigned to the bad class (essentially increasing the size of the bad class). Though the results in some cases (PLS and LDA) did improve to some extent, the increases in classification rates did not appear to be significant and hence we omit them in this study.

One way to consider the performance of the models is shown in Figs. 5.2, 5.3, and 5.4. The probability for membership in the good class is plotted against the residuals from the original linear regression model. The probabilities were obtained from the neural network classification models. Fig. 5.2 is the plot for the prediction set of the DIPPR dataset. Ideally one would expect that such a graph would have a cluster of points in the upper left quadrant and a cluster in the lower right quadrant. However, in practice such a perfect distribution is rare, although the graph does indicate the general trends. In the lower right there is a vertical set of points with probability 1.0 that exhibit low values of the absolute standardized residual. On the left hand side of the graph, we see a similar set of points with probability values equal to 0.0 (indicating that they belong to the bad class). In between these two extremes we see points that have probabilities indicating membership to the good class. However for points with probabilities lying in the range 0.5 to 0.7 such membership is probably not conclusive, and we see that their residuals are also midway between the two extremes. The points at the left and right edges of the graph indicate that when the class predictions of the CNN classifier are accompanied by high or low probabilities, the residuals from the linear regression model can be expected to be low or high, respectively. The two anomalous points marked by red triangles represent the misclassified cases. The one on the right was predicted as belonging to

the good class, whereas its true membership was to the bad class, and vice versa for the point on the right. It is not apparent why these points would be misclassified. But more importantly, it is not clear how one might consider them misclassified without having the actual residuals available, since in a real application we would be dealing with observations whose actual activities are not known.

Fig. 5.3 shows the corresponding plot for the Liu dataset. As before, observations predicted to be in the bad class and the good class (with high certainty) are located in the upper left and lower right quadrants respectively. In this case, there is only a single point whose membership is not absolutely certain.

Finally, Fig. 5.4 shows the plot for the artemisinin dataset. In this case the plot is not as tight as the previous ones, with the probability values of a number of observations indicating that membership in the good class is not very conclusive. The misclassified observations are interesting. The one misclassified point on the left hand edge would certainly be difficult to detect in the absence of residuals. However, the remaining two misclassified points are more or less on the border between the two lower quadrants. In addition they are also quite close to points that have been correctly classified. This is indicative of the fact that membership of observations when their probabilities lie around 0.5 can inconclusive and thus one should be wary of such points.

5.6 Further Work

The methodology described here appears to perform reasonably well on the three datasets we investigated. However, there are several features that require further study. First, the classification approach described here is a two-class problem. We restricted ourselves to the two-class problem for simplicity. Considering the scheme as a three-class problem might enable the user to draw more fine-grained conclusions regarding the validity of the results obtained from a regression model. However, increasing the number of classes will certainly require a large dataset and even if such a dataset is used, the unbalanced nature of the classes will require careful selection of a classification technique. We note that the results presented in this study are dependent on the nature of the datasets employed – specifically the distribution of residuals which is itself dependent on the distribution of the compounds in descriptor space. However, the datasets that we selected for testing include both physical properties for a number of congeneric series as well as biological properties for a set of molecules containing exhibiting varying structures and functionality. Furthermore the datasets we selected allowed us to test our techniques

with different types of linear models. For example, the DIPPR dataset was described by a linear model with very good statistics and very low residual values in general. On the other hand, the artemisinin dataset was characterized by lower values of R^2, high RMSE value and a number of observations with large residuals. As a result the DIPPR dataset presented our methodology with severely unbalanced classes whereas the class distribution was not as skewed in the case of the arteminsin dataset. Furthermore it is often the case that linear models for biological properties do not exhibit high quality statistics and contain a number of outliers. Thus the use of this dataset allowed us to test our technique in a real world scenario. Finally, the Liu dataset that was prepared by hand allowed us to have specific observations with large residuals and thus test the ability of the methodology to specifically detect these types of compounds. As has been shown, our methodology appears to perform well on these varied datasets. The only downside to the selection of our datasets is that the sizes are not as large as we would have liked them to be. Larger datasets would allow us to experiment with more than two classes as well as other classification schemes as discussed below. Clearly, one possible avenue of investigation is the validation of our methodology on different (and larger) datasets.

Modified sampling schemes like those described do not appear to improve the results significantly. The initial assignment of classes to the training set data is a step that could be modified, as the current approach employs an arbitrary assignment scheme. To remove this user defined task, class assignments can be automated by the use of regression diagnostics. However, such a scheme would then restrict the application of this methodology to linear models only. It appears that for full generality some form of cut-off value must be specified by the user. However, one advantage of a user- specified cut-off value is that it allows the user to focus on a range of residual values. Coupled with multiple (more than two) classes, this would allow the user to perform a fine-grained analysis of the residual classes.

Of the classification techniques investigated in this study it appears that neural networks performed the best with overall classification rates ranging from 79% to above 90% for the training set and 73% to 90% for the prediction set. The linear methods did not appear to perform significantly better than random. Furthermore, introduction of a similarity measure as an independent variable did not lead to improved classification results using any of the methods.

An alternative approach that may be considered is a Bayesian classification scheme whereby the training set class assignments are used to build up a prior probability distribution and the probability of new compounds belonging to a given class can be obtained by sampling from the simulated distribution. Associated with each class prediction is a probability for the membership to the predicted class. This requirement restricted our choice of classification technique somewhat but we feel that the lack of such a posterior probability would result in this method not being any more useful than simply recalculating the original regression model with some sort of scoring feature. The plots of posterior probability versus residuals are a good indicator of the performance of this methodology and also allows us to identify misclassifications in general. However, misclassified examples that are associated with posterior probabilities around 0.5 are, in general, not distinguishable from correctly predicted examples with similar posterior probabilities. In such cases one would probably be justified in ignoring compounds whose class predictions are borderline and rather concentrate on those compounds that are classified with high posterior probabilities of belonging to the good or bad class.

5.7 Conclusions

This chapter describes a novel and general scheme to provide a measure of confidence for the predictions from a regression model. The methodology described here attempts to answer the following question: how well will a regression model predict the property value for a compound that was not in the training or prediction set of the model? That is, we have attempted to extend and unify the characterization of generalizability for different types of QSAR models. Multiple approaches were investigated resulting in a classification scheme in which the training set residuals were assigned to one of two classes depending on whether they lay above or below a cut-off value. A classifier was then built with these assignments and used to predict the class of the residual for a new compound. The technique appears to be general enough to be applicable to any given regression model. We investigated several classification techniques and a neural network approach produced the best classification rates. The performance of the algorithm was visualized by considering plots of posterior probabilities versus residuals.

Though the performance of regression models may be judged via other scoring methods, such as confidence bands or frequency based scores, these methods are generally specific to the regression modeling technique employed. The method described here is quite general and thus can be applied to regression models developed using linear

regression, neural networks or random forests. Furthermore, the methodology is not dependent on the original dataset. All that is required is the availability of the original residuals (which is generally available in models developed with common statistical packages). Another attractive feature is that apart from the threshold residual value, the methodology does not require extra information such as similarity measures or new descriptors, since it restricts itself to using the descriptors that were used in the original quantitative model. We believe that such a parsimonious approach minimizes complexity as well as user intervention. The net result of our methodology is a probability of whether a compound (with an unknown property value) will have a high or low residual (relative to a user specified cut-off value) when processed by the regression model. Clearly, this does not replace the use of the original quantitative model. Rather, the methodology allows us to generate confidence measures for new compounds for any type of quantitative regression model in the absence of the original data and in a parsimonious manner. As a result methodology could be used as a component of a high throughput screening process in which different regression techniques are employed in a consensus based strategy.

Table 5.1: Molecules and experimental boiling point values comprising the toy dataset selected by hand from Liu et al.[5] and the literature

Name	BP (K)	Name	BP(K)
methane	-164.00	2,2,3-trimethylpentane	110.00
ethane	-88.60	2,2,4-trimethylpentane	99.20
propane	-42.10	2,3,3-trimethylpentane	114.70
butane	-0.50	2,3,4-trimethylpentane	113.40
2-methylpropane	-11.70	2-methyl-3-ethylpentane	115.60
pentane	36.10	3-methyl-3-ethylpentane	118.20
2-methylbutane	27.80	2,2,3,3-tetramethylbutane	106.50
2,2-dimethylpropane	9.50	nonane	150.77
hexane	69.00	2-methyloctane	142.80
2-methylpentane	60.30	3-methyloctane	143.80
3-methylpentane	63.30	4-methyloctane	142.40
2,2-dimethylbutane	49.70	2,2-dimethylheptane	132.70
2,3-dimethylbutane	58.00	2,3-dimethylheptane	140.50
heptane	98.40	2,4-dimethylheptane	133.50
2-methylhexane	90.00	2,5-dimethylheptane	136.00
3-methylhexane	92.00	2,6-dimethylheptane	135.20
2,2-dimethylpentane	79.20	3,3-dimethylheptane	137.30
2,3-dimethylpentane	89.80	3,4-dimethylheptane	140.10
2,4-dimethylpentane	80.50	3,5-dimethylheptane	136.00
3,3-dimethylpentane	86.10	4,4-dimethylheptane	135.20
3-ethylpentane	93.50	3-ethylheptane	143.00
2,2,3-trimethylbutane	80.90	4-ethylheptane	141.20
octane	125.70	benzene[a]	80.10
2-methylheptane	117.60	benzoic acid[a]	249.00
3-methylheptane	118.00	cyclohexane[a]	80.70
4-methylheptane	117.70	decane[a]	174.10
2,2-dimethylhexane	106.80	bromomethane[a]	3.50
2,3-dimethylhexane	115.60	propylamine[a]	48.00
2,4-dimethylhexane	109.40	2,3,3-trimethylhexane	131.70

Table 5.1: (continued)

Name	BP (K)	Name	BP(K)
2,5-dimethylhexane	109.00	pyrrole[a]	130.00
3,3-dimethylhexane	112.00	anthracene[a]	340.00
acetic acid[a]	117.90		

[a]Boiling point obtained from www.chemfinder.com

Table 5.2. Statistics for the linear regression model using the artemisinin dataset.

Description	β	Std. Error	t	P	VIF
Constant	-60.5625	5.2834	-11.5	2×10^{-16}	
N7CH	-0.2148	0.0134	-16.1	2×10^{-16}	1.6
NSB-12	0.2238	0.0238	9.4	2×10^{-16}	1.3
WTPT-2	27.9391	2.6136	10.7	2×10^{-16}	1.4
MDE-14	0.1118	0.0247	4.5	1.18×10^{-5}	1.5

N7CH - number of 7^{th} order chains;[17–19] NSB-12 - number of single bonds; WTPT-2 - the molecular ID number[20] considering only carbon atoms; MDE-14 - the molecular distance edge vector,[5] considering only primary and quaternary atoms.

Table 5.3. Statistics for the linear regression model using the DIPP dataset.

Description	β	Std. Error	t	P	VIF
Constant	179.15628	2.02828	88.329	$< 2 \times 10^{-16}$	
FPSA-3	-175.87824	2.88552	-60.952	$< 2 \times 10^{-16}$	1.6
FNSA-3	1.36298	0.01395	97.675	$< 2 \times 10^{-16}$	1.8
RNCG-1	-0.65982	0.11676	-5.651	4.70×10^{-8}	1.2
RPCS-1	-0.38502	0.07294	-5.279	3.00×10^{-7}	1.1

FPSA-3 - partial positive surface area divided by the total molecular surface area;[21] FNSA-3 - charge weighted partial surface area divided by the total molecular surface area;[21] RNCG-1 - the difference between the relative negative charge and the most negative charge divided by the total negative charge;[21] RPCS-1 - the positive charge analog of RNCG-1 multiplied by the difference between relative positively charged surface area and the most positively charged surface area.[21]

Table 5.4. Statistics for the linear regression model using the toy dataset.

Description	β	Std. Error	t	P	VIF
Constant	-381.6960	60.3677	-6.323	8.72×10^{-8}	
EMIN-1	-43.2189	9.1003	-4.749	1.95×10^{-5}	1.1
EMAX-1	88.8862	10.4446	8.510	4.46×10^{-11}	1.5
ECCN-1	1.2717	0.1052	12.089	4.99×10^{-16}	1.2
SHDW-6	501.1936	136.7371	3.665	6.27×10^{-4}	1.2

EMIN-1 - minimum atomic estate value;[22] EMAX-2 - maximum atomic estate value;[22] ECCN-1 - eccentric connectivity index;[23] SHDW-6 - the area of the molecule when projected onto the XY plane[24,25]

Table 5.5. Summary statistics for the three linear models used in this study

	Training Set		Prediction Set			
Dataset	R^2	RMSE	R^2	RMSE	F statistic	p value
Artemisinin	0.70	0.87	0.05	0.75	95.28 (4,156)	2.2×10^{-16}
DIPP	0.99	7.22	0.99	7.42	9521 (4,230)	2.2×10^{-16}
Toy	0.90	18.84	0.01	352.30	111.9 (4,47)	2.2×10^{-16}

Table 5.6. Cutoff values used for each dataset and the resultant size of each class

		Class Size	
Dataset	Cutoff	Good	Bad
artemisinin	1.0	133	46
DIPP	1.0	213	64
toy	1.0	44	21

Table 5.7. Confusion matrices for the linear discriminant analysis of the artemisinin dataset with and without atom pair similarity.

	Training Set			Prediction Set		
		Predicted			**Predicted**	
AP similarity excluded	**Actual**	bad	good	**Actual**	bad	good
	bad	2	40	bad	0	4
	good	2	117	good	0	14
		Predicted			**Predicted**	
AP similarity included	**Actual**	bad	good	**Actual**	bad	good
	bad	0	42	bad	0	4
	good	0	119	good	0	14

Table 5.8. Confusion matrices for the linear discriminant analysis of the DIPP dataset with and without average atom pair similarity .

	Training Set			Prediction Set		
		Predicted			**Predicted**	
AP similarity excluded	**Actual**	bad	good	**Actual**	bad	good
	bad	4	50	bad	1	9
	good	4	177	good	1	31
		Predicted			**Predicted**	
AP similarity included	**Actual**	bad	good	**Actual**	bad	good
	bad	4	50	bad	1	9
	good	4	177	good	1	31

Table 5.9. Confusion matrices for the linear discriminant analysis of the toy dataset with and without average atom pair similarity .

AP similarity excluded

Training Set

	Predicted	
Actual	bad	good
bad	7	11
good	4	30

Prediction Set

	Predicted	
Actual	bad	good
bad	2	1
good	3	7

AP similarity included

	Predicted	
Actual	bad	good
bad	4	14
good	4	30

	Predicted	
Actual	bad	good
bad	3	0
good	2	8

Table 5.10. Confusion matrices for the of the artemisinin dataset using a neural network with and without atom pair similarity.*

AP similarity excluded

Training Set

	Predicted	
Actual	bad	good
bad	38	4
good	27	92

Prediction Set

	Predicted	
Actual	bad	good
bad	4	0
good	3	11

AP similarity included

	Predicted	
Actual	bad	good
bad	34	8
good	46	73

	Predicted	
Actual	bad	good
bad	3	1
good	4	10

* The architecture for the CNN with atom pair similarity excluded was 4–9–1 and with the similarity included was 5–5–1

Table 5.11. Confusion matrices for the DIPP dataset using a neural network with and without average atom pair similarity .*

	Training Set			Prediction Set		
		Predicted			**Predicted**	
AP similarity excluded	**Actual**	bad	good	**Actual**	bad	good
	bad	54	0	bad	9	1
	good	5	176	good	1	31
		Predicted			**Predicted**	
AP similarity included	**Actual**	bad	good	**Actual**	bad	good
	bad	54	0	bad	8	2
	good	5	176	good	2	30

* The architecture for the CNN with atom pair similarity excluded was 4–5–1 and with the similarity included was 5–4–1

Table 5.12. Confusion matrices for the toy dataset using a neural network and without average atom pair similarity .*

	Training Set			Prediction Set		
		Predicted			**Predicted**	
AP similarity excluded	**Actual**	bad	good	**Actual**	bad	good
	bad	18	0	bad	3	0
	good	1	33	good	2	8
		Predicted			**Predicted**	
AP similarity included	**Actual**	bad	good	**Actual**	bad	good
	bad	17	1	bad	3	0
	good	2	32	good	2	8

* The architecture for the CNN with atom pair similarity excluded was 4–5–1 and with the similarity included was 5–5–1

Table 5.13. Weighted success rates for the various classification algorithms

Method	Dataset	Without Similarity		With Similarity	
		TSET	PSET	TSET	PSET
LDA	Artemisinin	0.51	0.50	0.50	0.50
	DIPP	0.52	0.53	0.52	0.53
	Toy	0.63	0.68	0.55	0.90
PLS	Artemisnin	0.51	0.46	0.49	0.5
	DIPP	0.36	0.53	0.36	0.53
	Toy	0.59	0.51	0.59	0.73
CNN	Artemisinin	0.79	0.80	0.71	0.73
	DIPP	0.98	0.93	0.98	0.86
	Toy	0.98	0.90	0.94	0.90

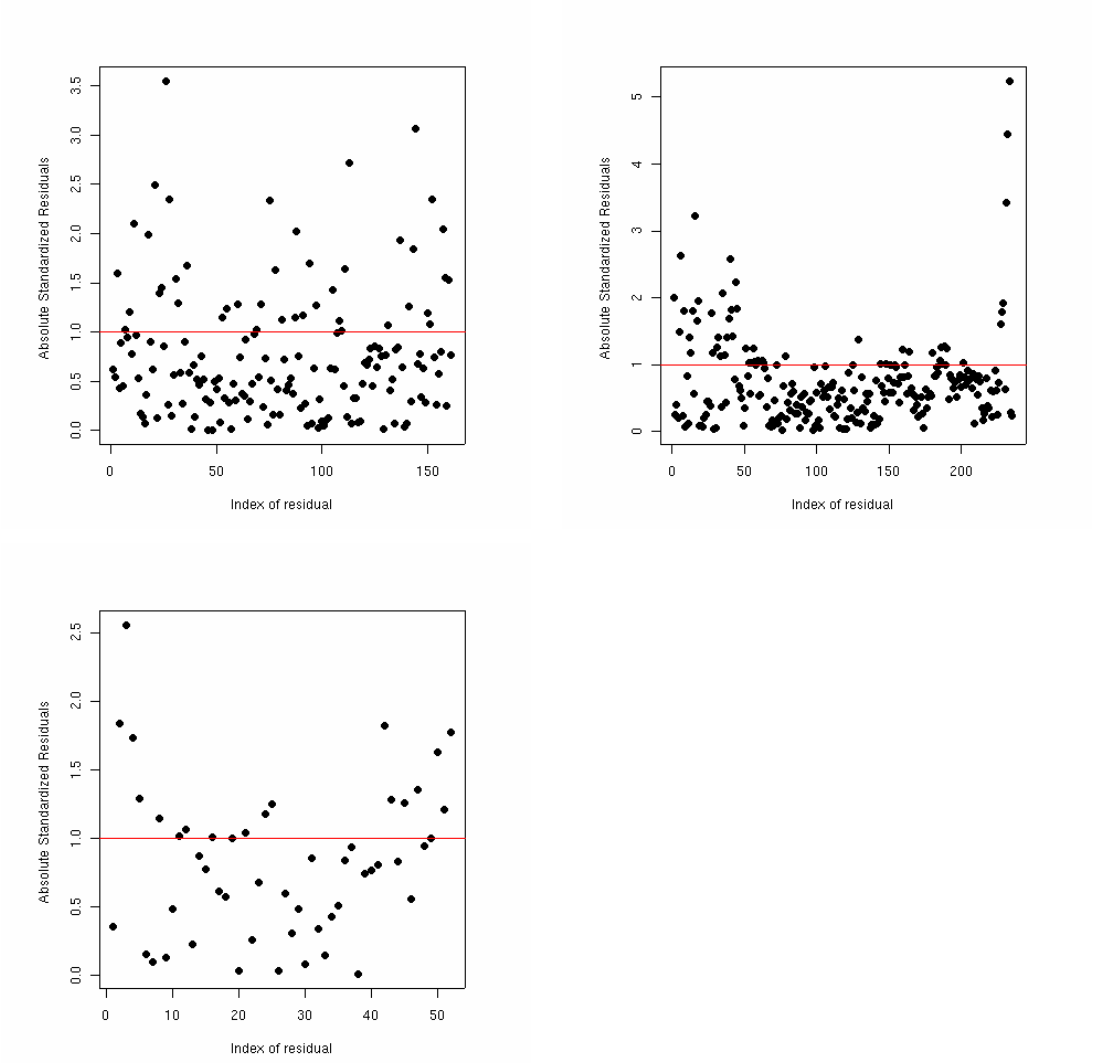


Fig. 5.1. Plots of absolute standardized residuals versus index of residual for the best linear models developed using the training sets for each dataset, with the cutoff value displayed. Residuals lying above the cutoff line are classfied as *bad* and those below as *good*.

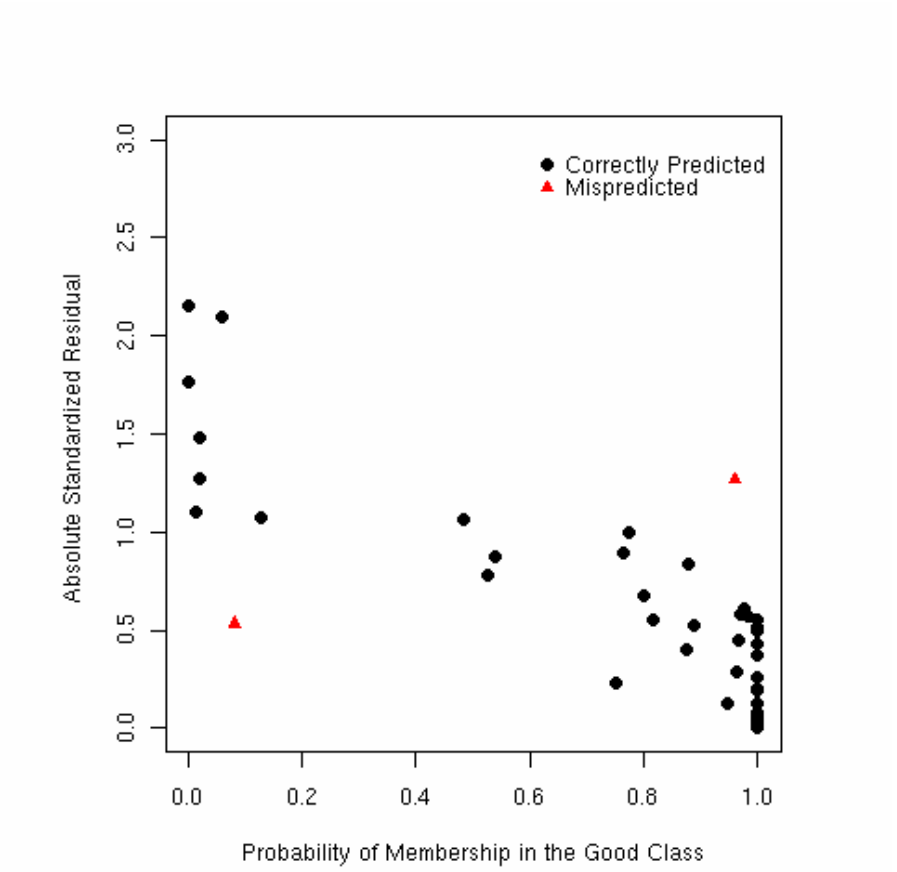


Fig. 5.2. Plot of probability of membership to the good class versus the absolute standardized residual for the DIPP dataset. The probabilities were obtained from the CNN model. The prediction set portion was used to generate the plot.

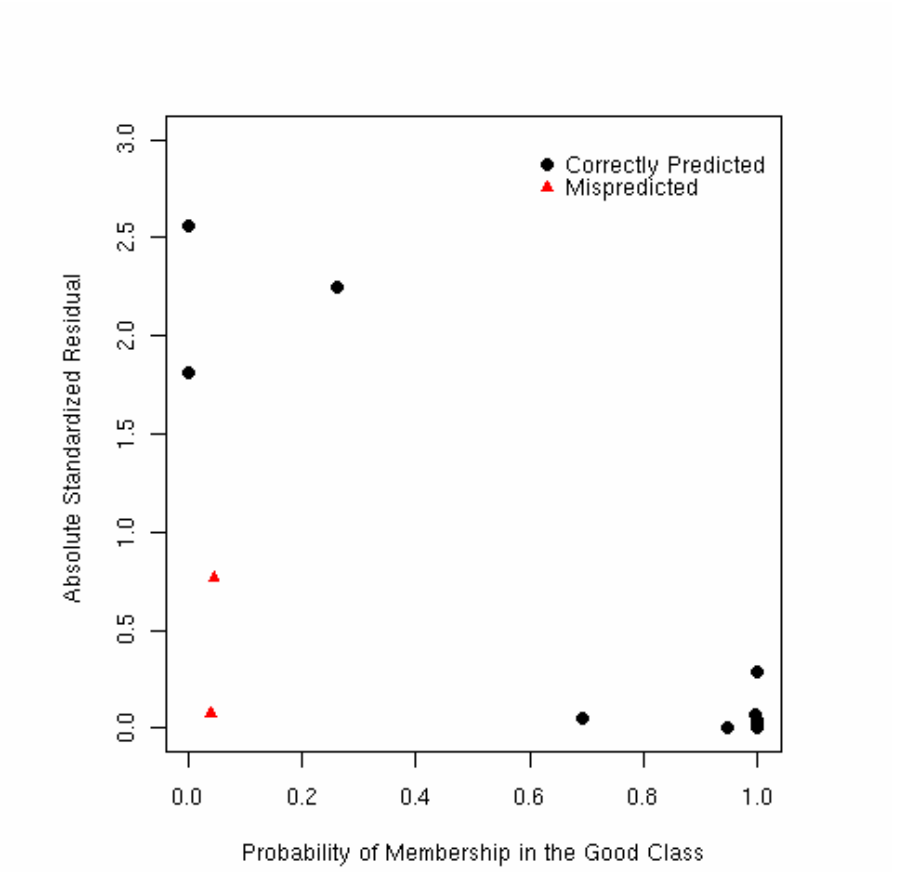


Fig. 5.3. Plot of probability of membership to the good class versus the absolute standardized residual for the toy dataset. The probabilities were obtained from the CNN model. The prediction set portion was used to generate the plot.

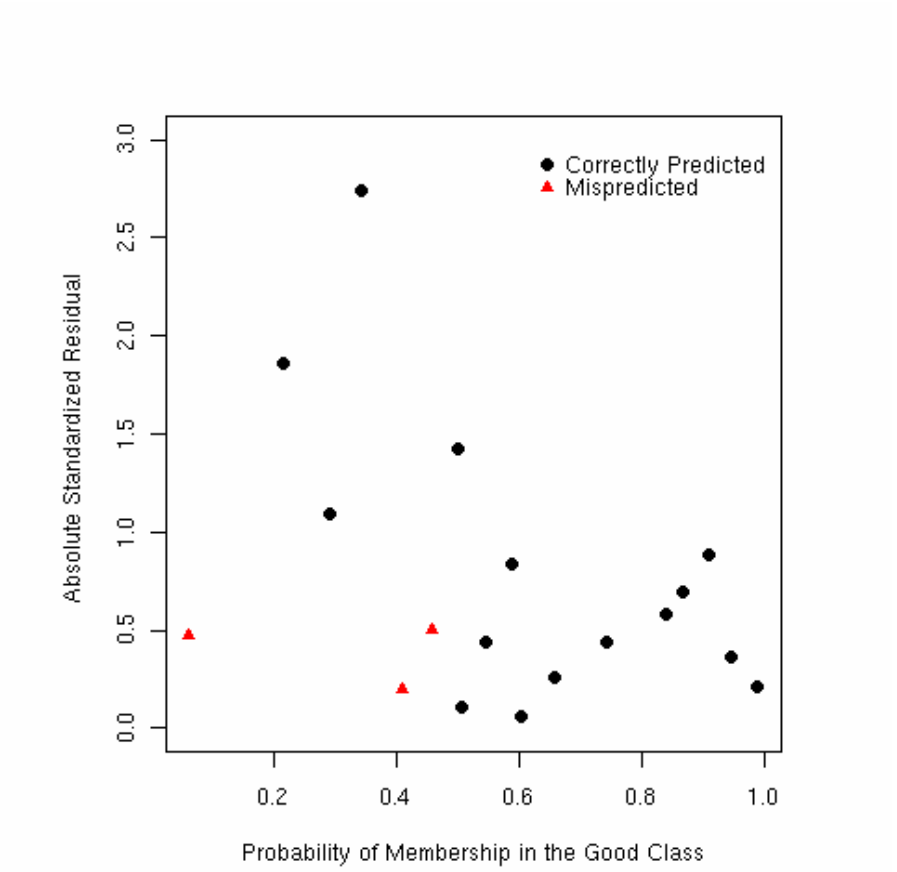


Fig. 5.4. Plot of probability of membership to the good class versus the absolute standardized residual for the artemisinin dataset. The probabilities were obtained from the CNN model. The prediction set portion was used to generate the plot.

References

[1] Carhart, R.; Smith, D.; Venkataraghavan, R. Atom Pairs as Molecular Features in Structure-Activity Studies: Definition and Applications. *J. Chem. Inf. Comput. Sci.* **1985,** *25,* 64–73.

[2] Goll, E.; Jurs, P. Prediction of the Normal Boiling Points of Organic Compounds From Molecular Structures with a Computational Neural Network Model. *J. Chem. Inf. Comput. Sci.* **1999,** *39,* 974–983.

[3] Avery, M. A.; Alvim-Gaston, M.; Rodrigues, C. R.; Barreiro, E. J.; Cohen, F. E.; Sabnis, Y. A.; Woolfrey, J. R. Structure-Activity Relationships of the Antimalarial Agent Artemisinin. The Development of Predictive In Vitro Potency Models Using CoMFA and HQSAR Methodologies. *J. Med. Chem.* **2002,** *45,* 292–303.

[4] Guha, R.; Jurs, P. The Development of QSAR Models to Predict and Interpret the Biological Activity of Artemisinin Analogues. *J. Chem. Inf. Comput. Sci.* **2004,** *44,* 1440–1449.

[5] Liu, S.; Cao, C.; Li, Z. Approach to Estimation and Prediction for Normal Boiling Point (NBP) of Alkanes Based on a Novel Molecular Distance Edge (MDE) Vector, λ. *J. Chem. Inf. Comput. Sci.* **1998,** *38,* 387–394.

[6] Jurs, P.; Chou, J.; Yuan, M. Studies of Chemical Structure and Biological Activity Relations Using Pattern Recognition. In *Computer assisted drug design*; Olsen, E.; Christoffersen, R., Eds.; American Chemical Society: Washington D.C., 1979.

[7] Stuper, A.; Brugger, W.; Jurs, P. *Computer Assisted Studies of Chemical Structure and Biological Function;* Wiley: New York, 1979.

[8] Japkowicz, N. Learning From Imbalanced Datasets: A Comparison of Various Strategies. In *Learning From Imbalanced Datasets: Papers From The AAAI Workshop*; AAAI Press: Menlo Park, CA, 2000.

[9] Kubat, M.; Matwin, S. Addressing The Curse Of Imbalanced Training Sets: One Sided Selection. In *Proceedings Of The 14^{th} International Conference On Machine Learning*; Morgan Kauffmnan: San Francisco, CA, 1997.

[10] Maloof, M. Learning When Data Sets are Imbalanced and When Costs are Unequal and Unknown. In *Workshop on Learning from Imbalanced Data Sets II*; ICML: Washington, D.C., 2003.

[11] Breiman, L. "Using Convex Pseudo-Data to Increase Prediction Accuracy", Technical Report, University of California, Berkeley, 1998.

[12] Breiman, L.; Friedman, J.; Olshen, R.; Stone, C. *Classification and Regresion Trees;* CRC Press: Boca Raton, FL, 1984.

[13] Svetnik, V.; Liaw, A.; Tong, C.; Culberson, C.; Sheridan, R.; Feuston, B. Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling. *J. Chem. Inf. Comput. Sci.* **2003,** *42,* 1947–1958.

[14] R Development Core Team, "R: A Language and Environment for Statistical Computing", R Foundation for Statistical Computing, Vienna, Austria, 2004 ISBN 3-900051-07-0.

[15] Ripley, B. *Pattern Recognition and Neural Networks;* Cambridge University Press: Oxford, 1996.

[16] Weston, J.; Pérez-Cruz, F.; Bousquet, O.; Chapelle, O.; Elisseeff, A.; Schölkopf, Feature Selection and Transduction for Prediction Of Molecular Bioactivity for Drug Design. *Bioinformatics* **2003,** *19,* 764–771.

[17] Kier, L.; Hall, L.; Murray, W. Molecular Connectivity I: Relationship to Local Anesthesia.. *J. Pharm. Sci.* **1975,** *64,*.

[18] Kier, L.; Hall, L. Molecular Connectivity VII: Specific Treatment to Heteroatoms. *J. Pharm. Sci.* **1976,** *65,* 1806–1809.

[19] Kier, L.; Hall, L. *Molecular Connectivity in Structure Activity Analysis.;* John Wiley & Sons: Hertfordshire, England, 1986.

[20] Randic, M. On Molecular Idenitification Numbers. *J. Chem. Inf. Comput. Sci.* **1984,** *24,* 164–175.

[21] Stanton, D.; Jurs, P. Development and Use of Charged Partial Surface Area Structural Descriptors in Computer Assissted Quantitative Structure Property Relationship Studies. *Anal. Chem.* **1990,** *62,* 2323–2329.

[22] Kier, L.; Hall, L. An Electrotopological-State Index for Atoms in Molecules. *Pharm. Res.* **1990,** *7,* 801–807.

[23] Sharma, V.; Goswami, A.; Madan, A. Eccentric Connectivity Index: A Novel Highly Discriminating Topological Descriptor For Structure-Property and Structure-Activity Studies. *J. Chem. Inf. Comput. Sci.* **1998,** *37,* 273–282.

[24] Stouch, T.; Jurs, P. A Simple Method for the Representation, Quantification and Comparison of the Volumes and Shapes of Chemical Compounds. *J. Chem. Inf. Comput. Sci.* **1986,** *26,* 4–12.

[25] Rohrbaugh, R.; Jurs, P. Molecular shape and Prediction of High Performace Liquid Chromatographic Retention Indices of Polycyclic Aromatic Hydrocarbons. *Anal. Chem.* **1987,** *59,* 1048–1054.

Chapter 6

The Development of QSAR Models To Predict and Interpret the Biological Activity of Artemisinin Analogues

6.1 Introduction

Qinghao (*Artemisia annua*) is an herb that has been used for over 2000 years in Chinese medicinal practice to treat fevers.[1] In 1972 the active compound of this herb, artemisinin, was isolated and was demonstrated to have significant antimalarial activity.[1] This finding was significant because artemisinin is structurally very different from the standard family of antimalarial drugs, which are based on quinine and its synthetic analogs. Subsequent research led to derivatives[1,2] of artemisinin such as artemether, arteether and artesunate (Fig. 6.1). The artemisinin family of molecules has been extensively studied to elucidate its mechanism of action as an antimalarial and to develop more potent and selective antimalarial agents.[3–6] An essential feature of artemisinin (and analogous molecules) activity is hypothesized to be the presence of a peroxide bridge, which forms a bond with a high valence non-heme iron molecule, leading to generation of free radicals.[4,5]

A number of QSAR studies have also been reported for prescreening of prospective artemisinin analogs for antimalarial activity.[7–15] A number of these studies[10–12] have used comparative molecular field analysis (CoMFA)[16,17] as a tool to model the activity of artemisinin analogs in terms of active site binding. CoMFA is a 3-D QSAR technique that involves the alignment of a set of molecules in three-dimensional space. Once a suitable alignment is obtained, a steric or electrostatic field is constructed using a probe atom. The resultant field is then correlated with the reported activity values of the molecules. An example of this is the work presented by Avery et al.[10] in which they considered a dataset of 211 artemisinin analogs. They performed PLS analyses of several

This work was published as Guha, R.; Jurs, P.C., "The Development of QSAR Models To Predict and Interpret the Biological Activity of Artemisinin Analogues", *J. Chem. Inf. Comput. Sci.*, **2004**, *44*, 1440–1449.

CoMFA models built using a number of different training sets and a test set of 15 or 20 compounds, depending on the size of the training set. Some of the models considered racemic compounds in the training set whereas other models excluded them. For the former class of models the R^2 values ranged from 0.82 to 0.88 with a Q^2 value of 0.72. For the latter class of models (in which the training set consisted of 157 molecules) they obtained R^2 values ranging from 0.95 to 0.96 for the training set while Q^2 values ranged between 0.68 and 0.73 during cross validation.

The goal of this study was to use the data collected by Avery et al.[10] to develop 2-D QSAR models using the ADAPT[18,19] methodology, which is not dependent on molecular alignments. The study resulted in two models - a linear model that focused on the interpretation of the structure-activity relationship (SAR) present in the dataset and a nonlinear CNN model that focused on predictive ability.

6.2 Dataset

The total dataset consisted of the 211 compounds reported by Avery et al.[10] For each molecule, the logarithm of the relative activity (referred to as log RA), defined as,

$$\log \mathrm{RA} = \log\left(\frac{IC_{50} \text{ of artemisinin}}{IC_{50} \text{ of analog}}\right) \times \log\left(\frac{MW \text{of analog}}{MW \text{of artemisinin}}\right)$$

was used as the dependent variable. However the dataset contained a number of enantiomeric pairs. Since the ADAPT descriptors cannot differentiate between enantiomeric molecules, the member of each pair with the lowest log RA value was removed. This resulted in a dataset of 179 molecules. One of the challenging aspects of this dataset was that it contained several molecules with the same log RA of -4.0. The molecules with log RA values of -4.0 were structurally diverse, thereby making model development a difficult task.

6.3 Methodology

The ADAPT methodology involves several steps. Though this has been described in detail in Chapter 3, we present a short summary describing the specifics of the model building process employed in this study.

The first step is to calculate molecular structure descriptors for the dataset. The suite of descriptors calculated by ADAPT include geometric, topological, electronic and hybrid descriptors. In all, 299 descriptors were calculated for each compound in the

dataset. However, many of these descriptors were highly correlated or contained redundant information. Hence, the next step involved objective feature selection in which highly correlated and redundant descriptors were removed from the pool. Using cut-offs of 0.80 for both the correlation and identical tests resulted in a reduced pool of 65 descriptors which was used for model development.

The molecules were divided into three sets, viz., training, cross-validation and prediction sets using the activity binning methods described in Chapter 3. This method resulted in the training set containing 144 molecules, the cross-validation set 17 molecules and the prediction set 18 molecules.

After objective feature selection, predictive models were generated by using a simulated annealing[20] or genetic algorithm[21] to search the descriptor space for optimal subsets of descriptors. The optimization routines were coupled with either a multiple linear regression routine or a computational neural network to find the best predictive models.

6.4 Results

6.4.1 Linear Models

The best linear model consisted of the four descriptors tabulated in Table 6.1. The first descriptor was the number of 7^{th} order chains.[22–24] A 7^{th} order chain is a series of seven atoms that contain at least one ring. For example, cycloheptane would have one count of a 7^{th} order chain. Furthermore, a molecule such as 1-methyl benzene would also have one count of a single 7^{th} order chain because it consists of seven atoms, six of which form a ring. The second descriptor was the number of single bonds. The third descriptor was a weighted-path descriptor which is based on a modification of the molecular ID number,[25] described by Randic, in which the molecular ID number is divided by the total number of atoms in the molecule. Molecular ID numbers were designed to provide unique identification numbers based on topological path lengths but stressing more on local features. In data presented by Randic,[25] a few general conclusions may be noted regarding the relation between molecular ID numbers and molecular structure. In the case of monocyclic systems, larger rings, larger chains and increased substitution (for a given substituent) lead to higher values of the molecular ID number. Furthermore, for bicyclic structures, equalized branches as well as substitution at carbons of lower valency lead to higher values of the molecular ID number. In the original work,[25] Randic did not discuss the case of polycyclic molecules in depth. However, one may conclude that

for the polycyclic structures used in this study a higher degree of branching (i.e., more substitutions) coupled with equalized branches in the ring system will lead to a higher value of the molecular ID number.

The final descriptor used in the model was a molecular distance edge vector,[26] denoted by λ. The MDE-14 descriptor is defined as the geometric mean of the topological path lengths between primary and quaternary carbons. As a result, one may view this descriptor as characterizing the side chains extending from the main body of a branched molecule. Thus the descriptor may also be correlated to molecular volume.[26] Consequently, molecules with a higher number of rings, longer side chains along with more substitution in the cyclic region will generally have higher values of MDE-14. As described in Liu et al.,[26] the descriptor provides good discrimination between structural isomers in a homologous dataset. Table 6.2 shows the maximum and minimum values for all descriptors in the best linear model.

The statistical details of this model are reported in Table 6.1. All the values of the t-statistic are significant, with low p-values, which confirms the significance of each descriptor. The F-statistic (on 4 and 157 degrees of freedom) for this model is 87.1 (compared to the critical value of 2.42 at the 0.05 level of significance) with a p-value of less than 2×10^{-16}. The lowest partial F-value for the coefficients was 20.5 (compared to a critical value for the F distribution with 1 and 157 degrees of freedom of 3.90 at the 0.05 level of significance). Furthermore, the variance inflation factors are all less than 1.6, which indicates the absence of multicollinearities in the model. Thus the model is statistically valid. The root mean square error (RMSE) for the training set was 0.86 (R^2 = 0.68) and the RMSE for the prediction set was 0.78 (R^2 = 0.77). Fig. 6.2 shows a fit plot of observed versus calculated log RA values. As can be seen from Fig. 6.2 there are several apparent outliers including a group of molecules having an observed log RA of -4.0, which are not well predicted. To detect outliers, and to investigate whether the latter molecules were behaving as outliers or simply as leverage points, a least trimmed squares[27] (LTS) regression algorithm was employed using the R software package.[28] We used LTS rather than the usual least squares regression to detect outliers due to the more robust nature of the LTS algorithm (it is resistant to non-normal error distributions). As a result, it is able to differentiate between leverage points and true outliers to a better extent than ordinary least squares regression. Using the LTS model, a plot of the standardized residuals versus observation was generated. Fig. 6.3 compares the plots of the standardized residuals versus the indices of the training set observations, for the least squares and LTS models. As is evident from the LTS residuals plot, three observations

appear to be distinct outliers with an additional two being borderline. The structures of the five molecules considered to be outliers by LTS regression are shown in Fig. 6.4. It is interesting to note that, in general the group of molecules with log RA values of -4.0 are not considered as outliers by the robust regression algorithm, thus demonstrating that this model was able to characterize these molecules well. However two members of the dataset with log RA values of -4.0 were classified as outliers. Though there are no significant features of these two molecules that sets them apart from other members of the group, it may be noted that in the case of the outliers the peroxide linkage is surrounded by one or two hydroxyl groups. However it is not apparent as to how this would cause the model to classify them as outliers. Once the outliers were detected they were removed from the training set and ordinary least squares regression was carried out again. Fig. 6.5 shows the results of a least squares regression in which the molecules classified as outliers by the LTS model have been removed. The statistics of the resultant model are improved compared to the original least squares model. The RMSE values for the training and prediction sets are both 0.77. The R^2 value for the training has increased to 0.74 along with an F-statistic (on 4 and 152 degrees of freedom) value of 108.3 (compared to a critical value of 2.43 at the 0.05 level of significance). The lowest partial F-value for the coefficients was 26.1 (compared to the critical value of the F distribution on 1 and 152 degrees of freedom of 3.90 at the 0.05 level of significance). The R^2 for the prediction set using the new model was 0.77. At this point it is useful to note that in the original work, outlier removal was not carried out. Outlier detection and regeneration of the linear model in the current work was carried out mainly to increase the quality of the linear model for subsequent interpretation using the PLS. That is, linear models were investigated mainly for the purpose of providing interpretive ability as opposed to predictive ability (which is discussed later, in the context of neural network models).

Finally, to ensure that the linear models were not due to chance correlations, the dependent variable for the training set (with the outliers removed) was scrambled a hundred times and linear models were built with the randomized dependent variables. If a true QSAR relationship exists with the real dependent variable, results for the scrambling runs should be very poor. The average R^2 for the 100 regressions was 0.02 with values ranging from 0.01 to 0.10. For the prediction set the average R^2 was 0.21 with values ranging from 0.0003 to 0.68. Though a value of 0.68 does appear to be unnaturally high it should be noted that this occurred once in one hundred randomized runs and that the next largest R^2 value was 0.30. It may also be noted that the above

results are in close accordance to the theoretically expected value of R^2 for a model built from random variables. Thus, these results indicate that chance correlations played a minimal role (if any) during the model development stages.

The last step in the analysis of the best linear model involved a partial least squares (PLS) analysis to provide an interpretation of the structure-activity relationships captured by the model. The technique described by Stanton[29] enables one to extract information regarding SAR trends captured by a linear model. The PLS interpretation methodology is discussed in detail in Section 3.6. The PLS analysis of the 4-descriptor model was carried out with the help of the Minitab[30] software package (using a leave-one-out cross validation scheme). The analysis indicated that the number of optimal components was four, thus the model was not over-fitted. A summary of the statistics for the 4 components is provided in Table 6.3. Table 6.4 shows the X-weights for the four valid components. Each PLS component is a linear combination of the four descriptors used in the model. Thus, the X-weights represent the contribution (or relative importance) of each descriptor within a given component. However, as can be seen from Table 6.3, component 4 explains less than 0.5% of the total variance (Q^2) explained by the model and so the following discussion only considers the first three components.

From Table 6.4 it is seen that in component 1, the most highly weighted descriptors are NSB and N7CH. The coefficient for NSB is positive indicating that a larger number of single bonds is correlated with a higher activity value. On the other hand the negative coefficient of N7CH indicates that smaller values of this descriptor are correlated with higher values of activity. This trend can be seen in molecules **43**, **45**, **47** and **51** all of which are inactive and have correspondingly high values for the descriptor N7CH. Molecules **11**, **79**, **107** and **159** are relatively active and have correspondingly small values for the N7CH descriptor. The structures of these molecules are compared in Fig. 6.6 and their positions are marked on the score plot for component 1 (Fig. 6.7). The feature common to the less active molecules is the fact that they all have an ether linkage bridging the seven-member ring, whereas the more active molecules contain a peroxide linkage. As a result of the presence of the ether linkage, the number of 7^{th} order chains (i.e., a contiguous series of seven atoms containing a ring structure) increases. From Fig. 6.6 it appears that molecule **51** is similar in size to molecule **107** and thus appears to invalidate the size trend described above. However molecule **51** does not contain an endoperoxide group but does have a number of ether linkages. The absence of the endoperoxide group is responsible for the low activity of this molecule, even though it is similar in size to the active molecules shown in Fig. 6.6. This is further confirmed

by the fact that a number of molecules in the dataset containing a peroxide linkage but lacking the endoperoxide group showed very low activities (with log RA values around -4.0). Since active compounds contain the endoperoxide group, this trend supports the theory that antimalarial activity of artemisinins depends on the presence of this group to form a high valence non-heme iron oxo species[4,5] and is evidence for the fact that the model has been able to capture an important feature of the datasets in the context of anti-malarial activity.

The other highly weighted descriptor in component 1 is NSB, the number of single bonds. This descriptor is very simplistic in nature and essentially characterizes the size of the molecule. Since the weight of the descriptor in the PLS model is positive, higher activity is correlated with a larger number of single bonds; indicating that larger molecules will tend to have higher activity, all other factors being equal. This trend can be seen in the log RA values for compounds **11**, **79**, **107**, and **159** (which are generally larger and have higher log RA values) and compounds **43**, **45**, **47** and **51** (which are generally smaller due to lack of large side chains and have lower log RA values). As can be seen from the score plot for component 1 (Fig. 6.7) the upper left (over-estimated) and lower right (under-estimated) regions of the plot are not significantly populated. Thus it appears that component 1 has been able to capture the majority of information regarding the molecules. This is also confirmed by the fact that the component 1 explains 60% of the variance (out of a total of 69.7%).

A similar analysis is performed with component 2. From the score plot shown in Fig. 6.8, it is seen that it accounts for some molecules that component 1 under-estimated. For example, molecules **116** and **143** are predicted correctly as more active whereas in component 1 they were under-estimated. The most highly weighted descriptors in component 2 are NSB and WTPT-2. In contrast to component 1 NSB is now negatively weighted indicating smaller values correlate with higher activities. This component thus corrects for larger molecules that might not be active. As a result, it moves molecule **175** from its position in the score plot for component 1 to a position closer to the lower left quadrant thus compensating for the over-estimation by component 1. As described previously, the WTPT-2 descriptor essentially characterizes the branched nature of a molecule, with the presence of larger rings, balanced branches (in the case of bicyclic molecules), longer chains and increased substitution (for a given substituent) leading to higher values of the molecular ID number. The molecules in the upper right quadrant of the score plot for component 2 (Fig. 6.8), such as **91** and **92**, support these trends.

They have relatively low values for NSB. They also have higher values of the WTPT-2 descriptor, which can be ascribed to the longer side chains. Both compounds are relatively active. In comparison, compounds **189** and **196** have low values for the WTPT-2 descriptor (which may be due to the absence of side chains) and are predicted as inactive by component 2. The structures of these molecules are compared in Fig. 6.9. In general, component 2 does not predict the inactive molecules very well (since the lower left quadrant is relatively unpopulated) thereby demonstrating the importance of component 2 in predicting the active compounds.

In component 3 the molecules that were not accounted for by components 1 and 2 are now correctly predicted as active (**135**, **185**, **194** and **4**) as can be seen from the score plot of component 3 (Fig. 6.10). In addition, compound **186** is also more accurately predicted thus correcting for the over-estimation by component 2. To some extent, component 3 makes up for the over- or under-estimations made by components 1 and 2. For this component the most significant descriptors are WTPT-2 and MDE-14. As mentioned previously, molecules with a higher number of rings, longer side chains along with more substitution in the cyclic region (especially bridging carbons) will generally have higher values of MDE-14. Taken with the positive sign of the weight for MDE-14, we may conclude that molecules with extended side chains coupled with substitution in the ring system (essentially, larger molecules) would exhibit higher activities, a trend also seen with NSB in component 1. The molecules present in the upper right (**133**, **135** and **64**) satisfy these trends and can be seen to have longer side chains as well as increased substitution on the rings compared to the inactive molecules. The molecules with smaller values of MDE-14 (**186**, **189** and **195**) are predicted as inactive compounds. The structures of these molecules are compared in Fig. 6.11.

It should be noted that a PLS analysis provides a guideline regarding the interpretation of the descriptors in the model and does not provide exact quantitative descriptions of descriptor contributions. Furthermore the analysis is restricted to the descriptors present in the best model. In this case, best implies best statistical quality and not necessarily the presence of meaningful descriptors.

6.4.2 Nonlinear Models

Several of the best nonlinear models selected by the genetic algorithm were analyzed rigorously in order to find the optimal neural network parameters. The four best models were further investigated by systematically varying the network architecture.

The results of the best four models are summarized in Table 6.5. The best model has a 10–5–1 architecture and contains the descriptors: KAPPA-6,[31–33] NDB, MOMI-4,[34] N7CH,[22–24] MOLC-8,[35,36] WTPT-5,[25] MDE-12,[26] MDE-13,[26] ELEC and FPSA-3.[37] The KAPPA-6 descriptor belongs to a class of descriptors termed Kier shape descriptors, denoted by κ^3. These descriptors are defined by the number of vertices and paths of length m, $(1 < m < 3)$ in a hydrogen depleted molecular graph. KAPPA-6 is the atom corrected version of the κ^3 descriptor and thus accounts for heteroatoms in addition to carbons. The values of the κ^3 descriptor are generally larger when molecular branching is absent or when branching occurs at the extremities of a molecular graph and thus this descriptor characterizes centrality of branching in a molecule.[38] The NDB is simply the count of double bonds in the molecule. MOMI-4 is the ratio of the X and Y components of the principal moment of inertia of the molecule. Thus, this descriptor provides information about the shape of a molecule in the XY plane. WTPT-5 is a modification of the molecular ID number[25] which only considers nitrogen atoms. The MOLC-8 descriptor is the 4^{th} order path cluster molecular connectivity index and measures the degree of branching in a structure. The descriptors MDE-12 and MDE-13 are the molecular distance edge vectors between primary & secondary and primary & tertiary carbons respectively. The ELEC descriptor is simply the electronegativity of the molecule. The value of electronegativity is taken as the mean of the HOMO and LUMO energies. The FPSA-3 descriptor belongs to a class of hybrid descriptors termed CPSA[37] descriptors. These combine partial charge and surface area information for a molecule resulting in a holistic description of polar surface area features. FPSA-3 is defined as the atom weighted partial positive surface area divided by the total molecular surface area.

It should be noted that the descriptors selected for the best CNN model are distinct from the descriptors used in the best linear model. The reason underlying this behavior is due to the different selection criteria that are used to include descriptors in the respective models from the reduced pool (which was the same for both linear and non-linear models). At the same time, one must consider the fact that different combinations of descriptors may be equally valid in describing a SAR trend. Furthermore, since a linear model is, by definition, restricted to capturing linear relationships, it cannot be used to investigate subsets of descriptors which when combined non-linearly might better describe a SAR trend. This is evidenced by using the descriptors from the best CNN model in a linear model. The resultant residuals are very high and the predictive power of such a model is very poor and as a result a linear model would not have considered this subset of descriptors.

The plot of observed versus calculated values is shown in Fig. 6.12. The RMSE values for the training, cross-validation and prediction sets were 0.42, 0.47 and 0.76, respectively. It is important to note that the group of compounds with log RA's of -4.0 have been predicted relatively well. The R^2 values for the training, cross-validation and prediction sets were 0.96, 0.94 and 0.88, respectively. To ensure that the behavior of the model was independent of the composition of the training, cross-validation and prediction sets the non-linear model described above was regenerated using a leave n% out procedure. In this procedure the molecules are ranked according to values of their activity and then grouped, the number of groups being determined from the percentage left out. Next, an equal number of empty groups are populated by selecting molecules from each group of the ranked dataset such that the whole range of activity is evenly distributed across the groups. These groups are then used to create the training, cross-validation and prediction sets. Essentially, for n groups, the first group is the prediction set, the second group is the cross-validation set and the remaining $n-2$ groups constitute the training set. These sets were then used to build and validate a CNN with a 10–5–1 architecture. In the next step, the last group was made the prediction set, the first group the cross-validation set and the remaining groups constitute the new training set. The CNN was rebuilt and validated using these sets. This process is repeated such that each group acts as the prediction set once. As a result the entire dataset is predicted once. The whole process is repeated so that each member of the dataset is predicted multiple times and the final reported value for a molecule is the average of all the predictions for that molecule.

In this study a leave 14% out procedure was used and repeated 3 times. Thus, each molecule in the dataset was predicted three times and the final reported value was the average of these predictions. The results of this procedure are summarized in Table 6.6 and a plot of the observed versus predicted values can be seen in Fig. 6.13. As can be seen the RMSE values for the cross-validation and prediction are degraded, compared to the original CNN model. Similar behavior is seen for the R^2 values. However, the standard deviations for these values over all the runs is quite low for the training and cross-validation sets. indicating that the model trains consistently. However when comparing these results to the original CNN model, only the statistics for the prediction set should be compared. The degradation of RMSE and R^2 values of the prediction set is to be expected as the model is trained on different sets of molecules at each stage. The leave 14% out procedure used here gives us a more realistic view of the behavior of this model when biases due to QSAR set composition are removed.

To further investigate the effect of QSAR set composition, a CNN model with a 10–5–1 architecture was generated using random sets (i.e., the training, cross-validation and prediction sets were selected randomly from the dataset). The results of this model are shown in Table 6.7. Though the RMSE and R^2 values for the training and cross-validation sets are similar to the average values for the leave 14% out based model, the prediction set performance is significantly degraded. This observation could simply be explained by considering that a poor combination of QSAR sets was created. However, an alternative explanation is that due to random set generation, the full range of activities are not properly represented in the training, cross-validation and prediction sets.

Finally to test whether the results described above could have been due to chance, Monte Carlo runs were carried out in which the dependent variable is scrambled and models are built using the scrambled dependant variable. The architecture was maintained at 10–5–1 and the descriptors used were those found in the best nonlinear model above. As can be seen from Table 6.8, the RMSE values increase significantly with a corresponding decrease in the R^2 values. This appears to indicate chance correlations did not play a significant role in the results described above.

6.5 Discussion and Conclusions

This chapter presents both linear and nonlinear models to predict anti-malarial activity for a set of 179 artemisinin analogs. The goal of the project was to create QSAR models, which were both interpretable as well as having good predictive ability. The linear regression model was found to be statistically valid and the PLS routine enabled an investigation of the effects of each descriptor in the model. That is, it was possible to isolate the action of the individual descriptors and explain specific SAR trends captured by the descriptors.

The nonlinear models were developed based mainly on their pure predictive ability. The non-linear model presented both superior predictive ability as well as a relatively simple neural network architecture. Interpretation of neural network models is difficult due to the black box nature of the neural network algorithm. Methods exists for a probabilistic interpretation of neural network classification models.[39,40] Techniques also exist to extract rules and decision trees from CNN regression models.[41,42] However these methods do not allow for a clear interpretation of descriptor contributions, as is available from a PLS analysis of a linear model, and are not easily combined with the ADAPT methodology. This problem is studied in more detail, in Chapter 9, where we describe an

approach to the interpretation of neural network models, based on the PLS interpretation technique for linear models. Finally, randomization tests showed that the possibility of chance correlations (if any) in the best models was low.

It may be noted that in both types of models the descriptors themselves are not necessarily amenable to simple physical interpretation. The ADAPT methodology seeks the most information rich subset of descriptors for a given model. In many cases the members of the resultant subset do not have simple physical meaning, but rather contribute information to the statistical model. That is, many descriptors calculated by ADAPT are not designed to necessarily provide a simple physical description of a molecule. Instead, they extract information that, in many cases, may be of a more abstract (such as graph theoretical) nature but provide information about a molecule. An attempt was made to introduce more meaningful descriptors into the models by replacing some of the selected descriptors with other correlated (and physically meaningful) descriptors from the reduced pool. In all cases, the resultant models performed poorly in comparison to the best models reported in this work.

A direct comparison with the original work is not feasible as the model development process in this study was different. However the results of the PLS analysis indicate that in terms of Q^2, the current linear model performs comparably to the original PLS model using 157 docked compounds described by Avery.[10] The PLS analysis was also able to provide an interpretation of the contributions of the individual descriptors in describing the overall activity of the majority of the molecules. One aspect of model interpretability would be a ranking of descriptor contributions. However the PLS technique does not allow a global ranking of individual descriptors since each PLS component is a linear combination of all the descriptors in the model. Thus such a ranking of descriptor contributions is only valid within a given component. One disadvantage of the current methodology was the inability to consider enantiomeric pairs compared to the original work in which the CoMFA methodology was able to handle such pairs. At the same time the current methodology does not involve the problem of alignments inherent in the CoMFA approach and furthermore was able to avoid making any assumptions regarding bioactive conformations.

Finally, though the linear model in this study does not exhibit significant predictive ability when compared to the models described by Avery,[10] it does provide interpretability. Coupled with the good predictive ability of the neural network model developed in this study we believe that these models would perform well as rapid screening tools to uncover new and more potent anti-malarial drugs.

Table 6.1. Statistics for the best linear regression model.

Description	β	Std. Error	t	P	VIF
Constant	-60.56	5.28	-11.50	2×10^{-16}	
N7CH	-0.21	0.01	-16.10	2×10^{-16}	1.60
NSB-12	0.22	0.02	9.40	2×10^{-16}	1.30
WTPT-2	27.94	2.61	10.70	2×10^{-16}	1.40
MDE-14	0.11	0.02	4.50	1.18×10^{-5}	1.50

N7CH - number of 7^{th} order chains;[22–24] NSB-12 - number of single bonds; WTPT-2 - the molecular ID number[25] considering only carbon atoms; MDE-14 - the molecular distance edge vector,[26] considering only primary and quaternary atoms.

Table 6.2. Maximum and minimum values for the descriptors used in the best linear model and the dependant variable.

	Dependent variable	N7CH	NSB	WTPT-2	MDE-14
Maximum	1.47	36	37	2.13	19.99
Minimum	-4.00	0	12	1.87	0.00

Table 6.3. Summary of the PLS analysis for the best 4 descriptor Linear model

Component	X Variance	Error SS	R^2	PRESS	Q^2
1	0.19	174.11	0.60	198.22	0.553
2	0.52	140.65	0.68	145.80	0.670
3	0.83	134.82	0.69	141.12	0.684
4	1.00	132.58	0.69	139.05	0.687

Table 6.4. The X-weights for the four optimal PLS components.

	Component 1	Component 2	Component 3	Component 4
N7CH	-0.68	-0.46	0.34	-0.47
NSB	0.65	-0.56	-0.14	-0.49
WTPT-2	0.27	0.54	0.67	-0.43
MDE-14	0.21	-0.44	0.65	0.59

Table 6.5. A summary of the various nonlinear CNN models generated.

	RMSE			R^2			
Architecture	TSET	CV SET	PSET	TSET	CV SET	PSET	Cost
3–2–1	0.81	0.79	0.81	0.66	0.66	0.70	0.67
7–4–1	0.90	0.89	0.80	0.49	0.48	0.76	0.51
7–5–1	0.91	0.92	0.81	0.47	0.42	0.70	0.50
10–5–1	0.96	0.94	0.88	0.42	0.47	0.76	0.44

Table 6.6. A summary of the statistics generated by a 3 round leave 14% out procedure using the best nonlinear CNN model (10–5–1 architecture).

	RMSE			R^2		
	TSET	CV SET	PSET	TSET	CV SET	PSET
Mean	0.44	0.59	0.89	0.91	0.85	0.69
Std. Deviation	0.05	0.10	0.16	0.01	0.06	0.11

Table 6.7. The results of a nonlinear CNN model (10–5–1 architecture) using randomly generated training, cross validation and prediction sets. The descriptors used were the same as those for the best nonlinear CNN model.

RMSE			R^2		
TSET	CV SET	PSET	TSET	CV SET	PSET
0.41	0.53	0.68	0.93	0.91	0.81

Table 6.8. The results of a nonlinear CNN model (10–5–1 architecture) using a scrambled dependant variable. The descriptors used were the same as those for the best nonlinear CNN model.

RMSE			R^2		
TSET	CV SET	PSET	TSET	CV SET	PSET
1.50	1.40	1.60	0.09	0.08	0.01

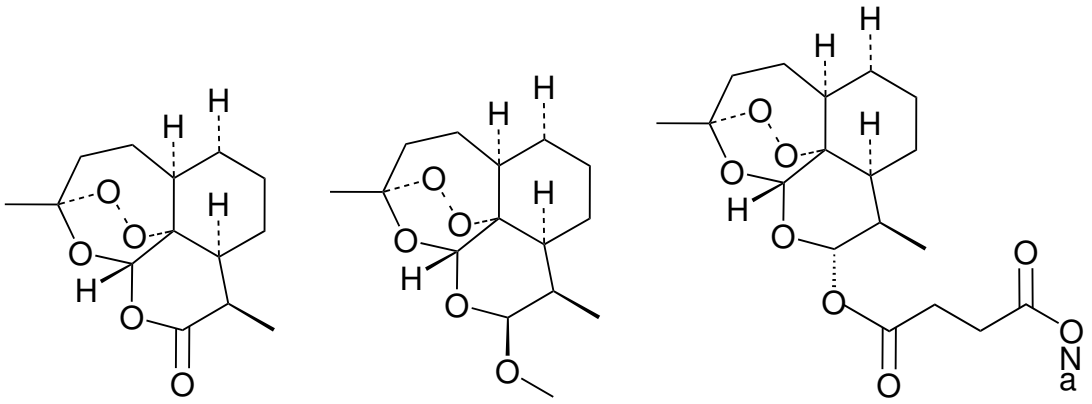


Fig. 6.1. Artemisinin and derivatives

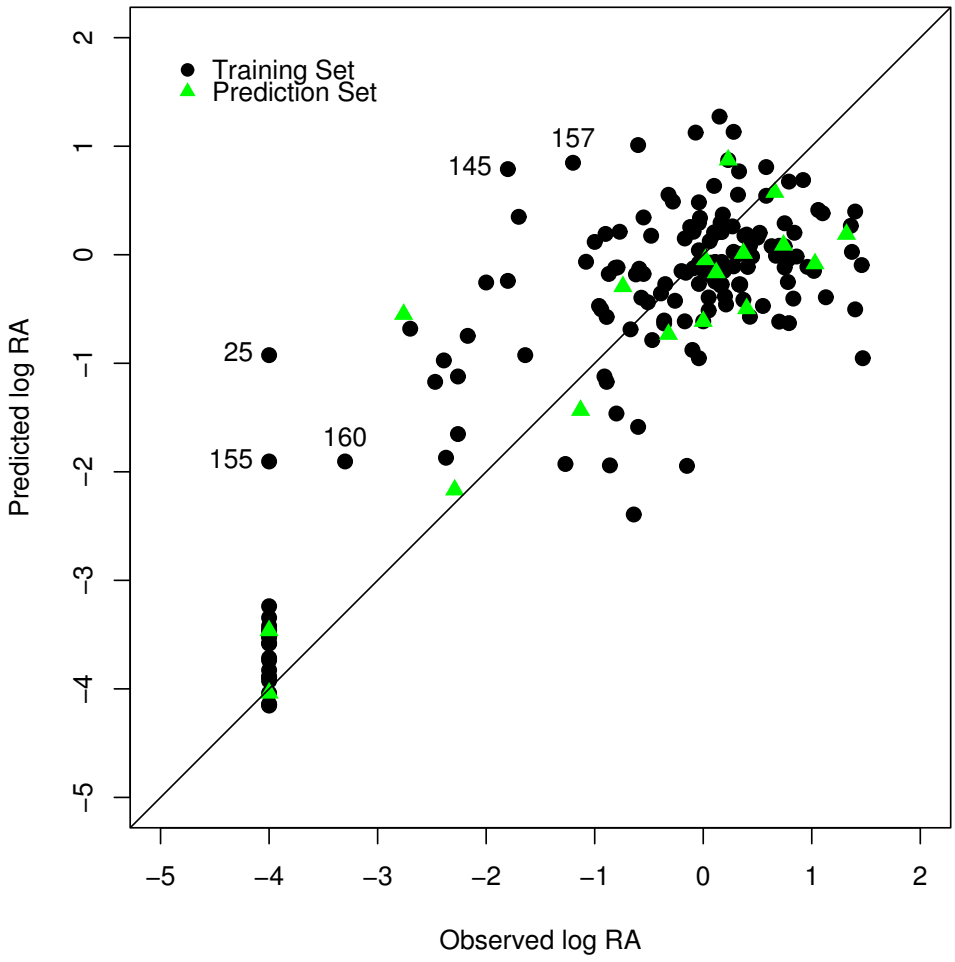


Fig. 6.2. A plot of observed versus predicted log RA from the best linear model. The numbered points are the molecules that were considered to be outliers in the residual plot generated using LTS regression (see Figs. 6.3 and 6.5).

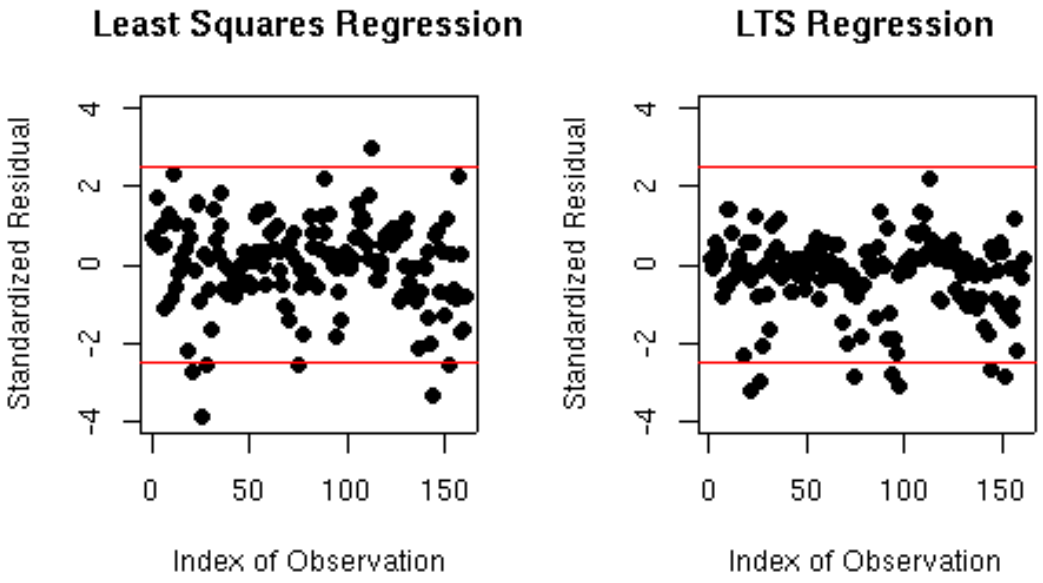
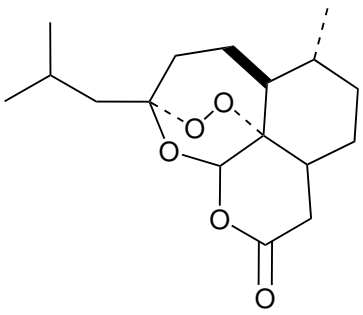
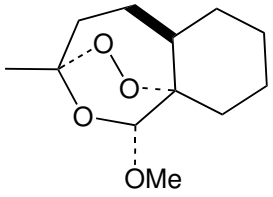


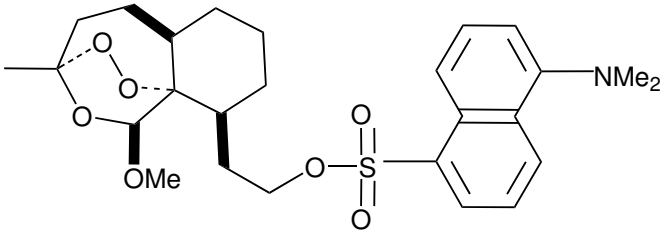
Fig. 6.3. A comparison of standardized residuals versus indices of the training set observations using simple least squares and the more robust LTS algorithm.



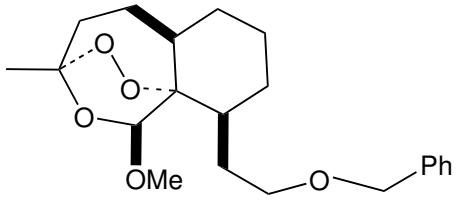
25 (-0.35)



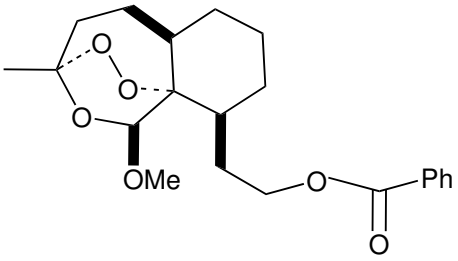
145 (-1.04)



155 (0.33)



157 (0.75)



160 (-0.59)

Fig. 6.4. The structures of the outliers (and corresponding activity values) detected in the best linear model using LTS regression.

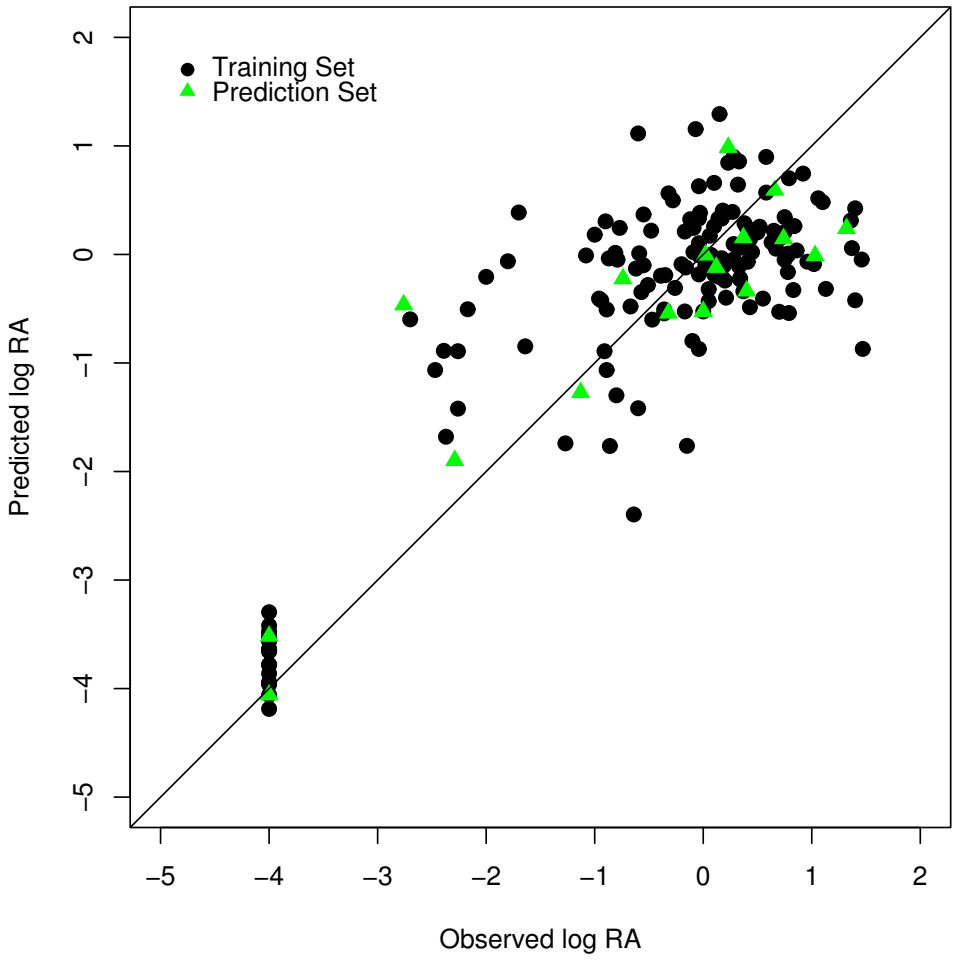


Fig. 6.5. A plot of observed versus predicted log RA after outliers detected via LTS regression have been removed.

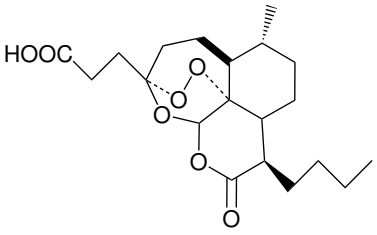
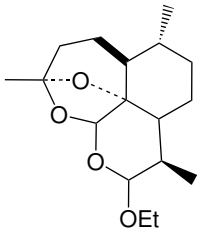
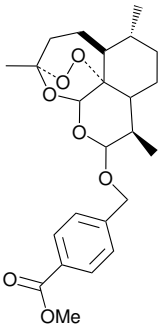
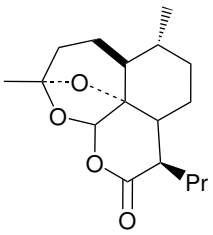
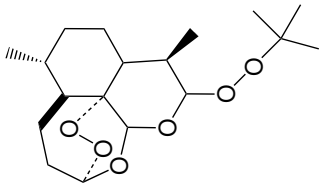
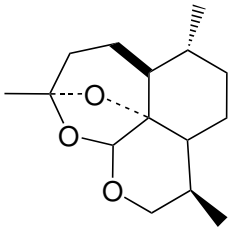
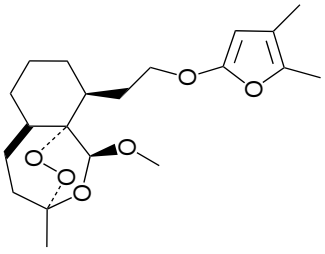
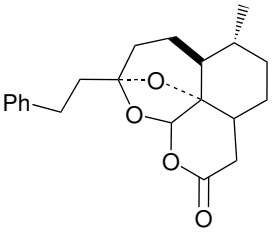
Active	Inactive
	
11 (1.36)	**43** (-4.00)
	
79 (-0.07)	**45** (-4.00)
	
107 (0.92)	**47** (-4.00)
	
159 (0.58)	**51** (-4.00)

Fig. 6.6. A comparison of the more active and less active compounds described using component 1. The value of log RA is provided within parentheses.

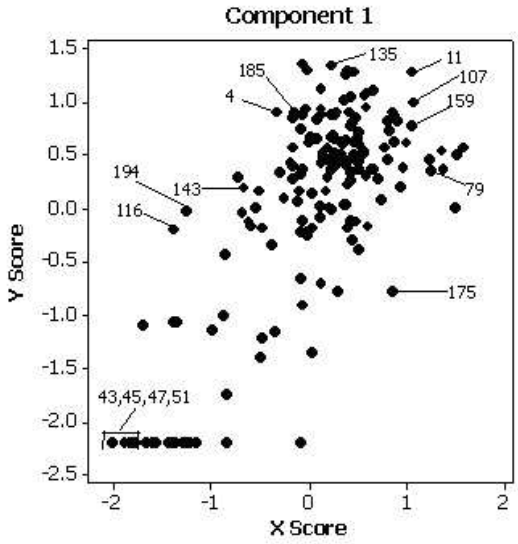


Fig. 6.7. The score plot for component 1.

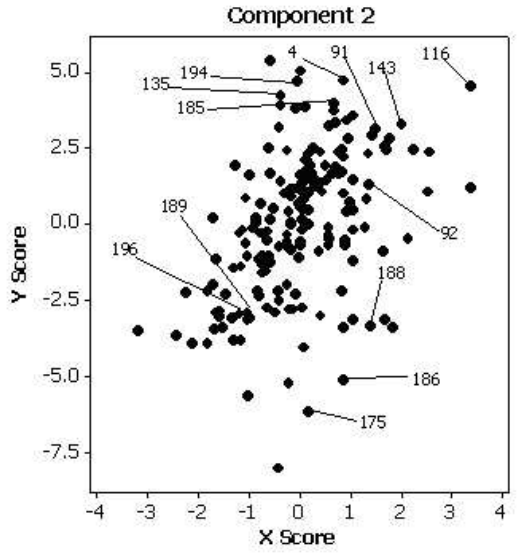


Fig. 6.8. The score plot for component 2.

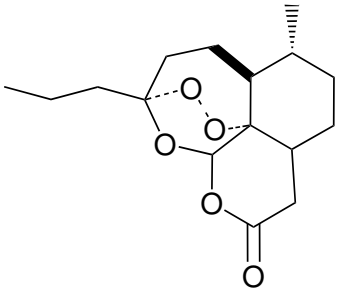
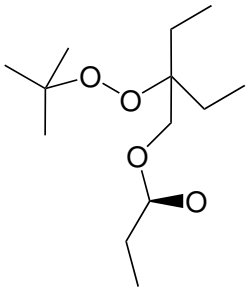
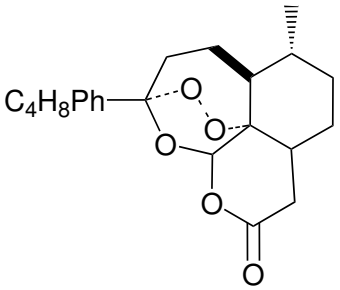
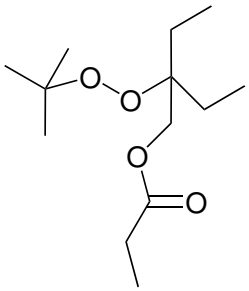
Active	**Inactive**
	
91 (0.84)	**189** (-4.00)
	
92 (0.58)	**196** (-4.00)

Fig. 6.9. A comparison of the more active and less active compounds described using component 2. The value of log RA is provided within parentheses.

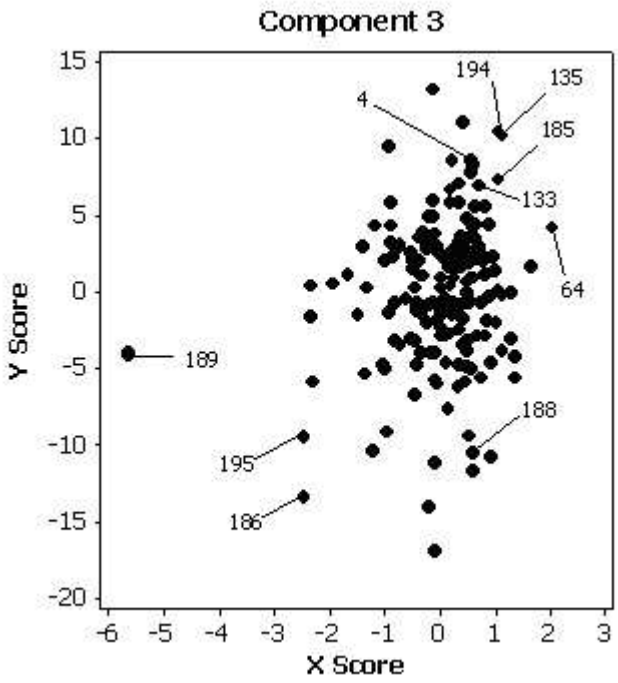


Fig. 6.10. The score plot for component 3.

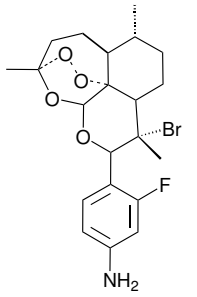
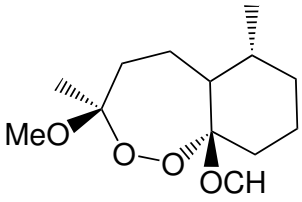
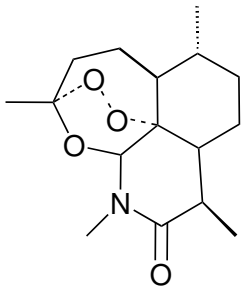
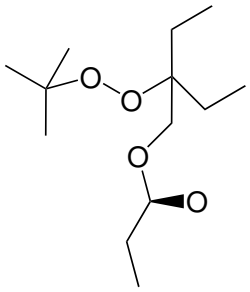
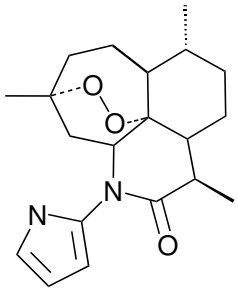
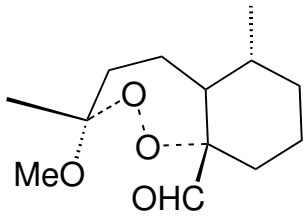
Active	Inactive
	
64 (0.79)	**186** (-4.00)
	
133 (0.43)	**189** (-4.00)
	
135 (1.46)	**195** (-3.30)

Fig. 6.11. A comparison of the more active and less active compounds described using component 3. The value of log RA is provided within parentheses.

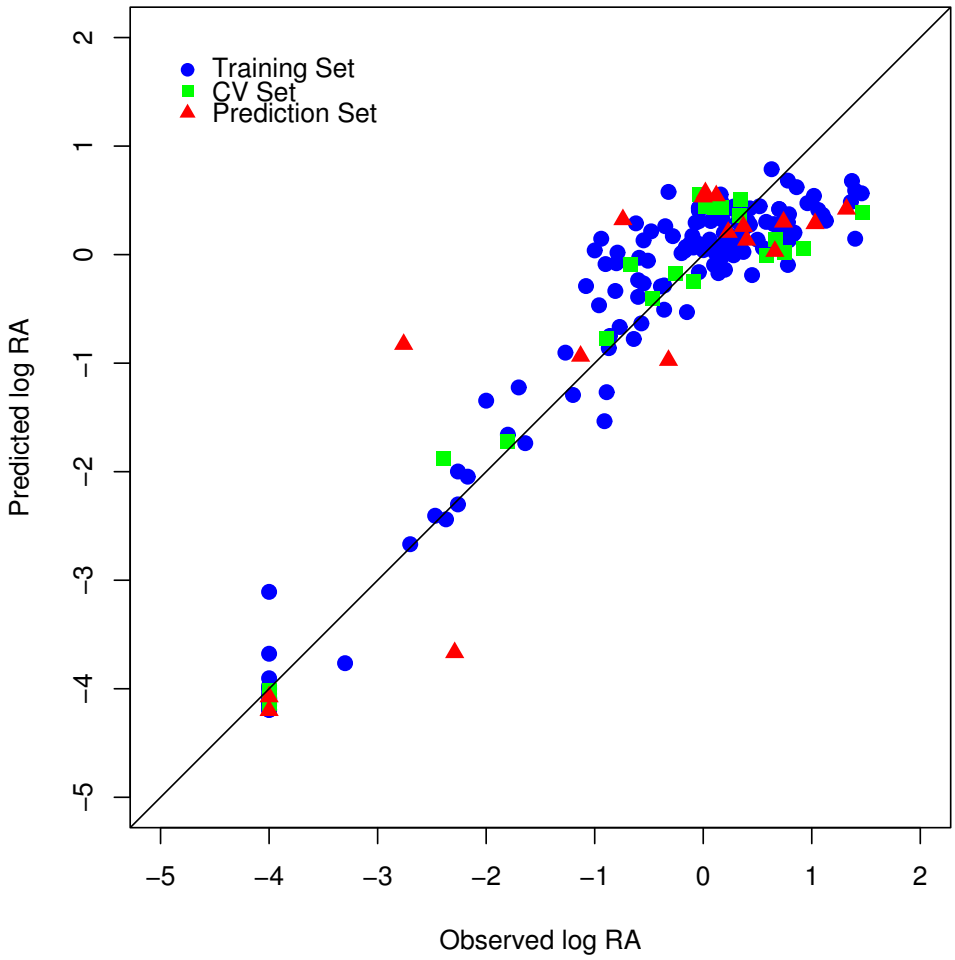


Fig. 6.12. A plot of observed versus predicted log RA produced from the best nonlinear CNN model using a 10–5–1 architecture.

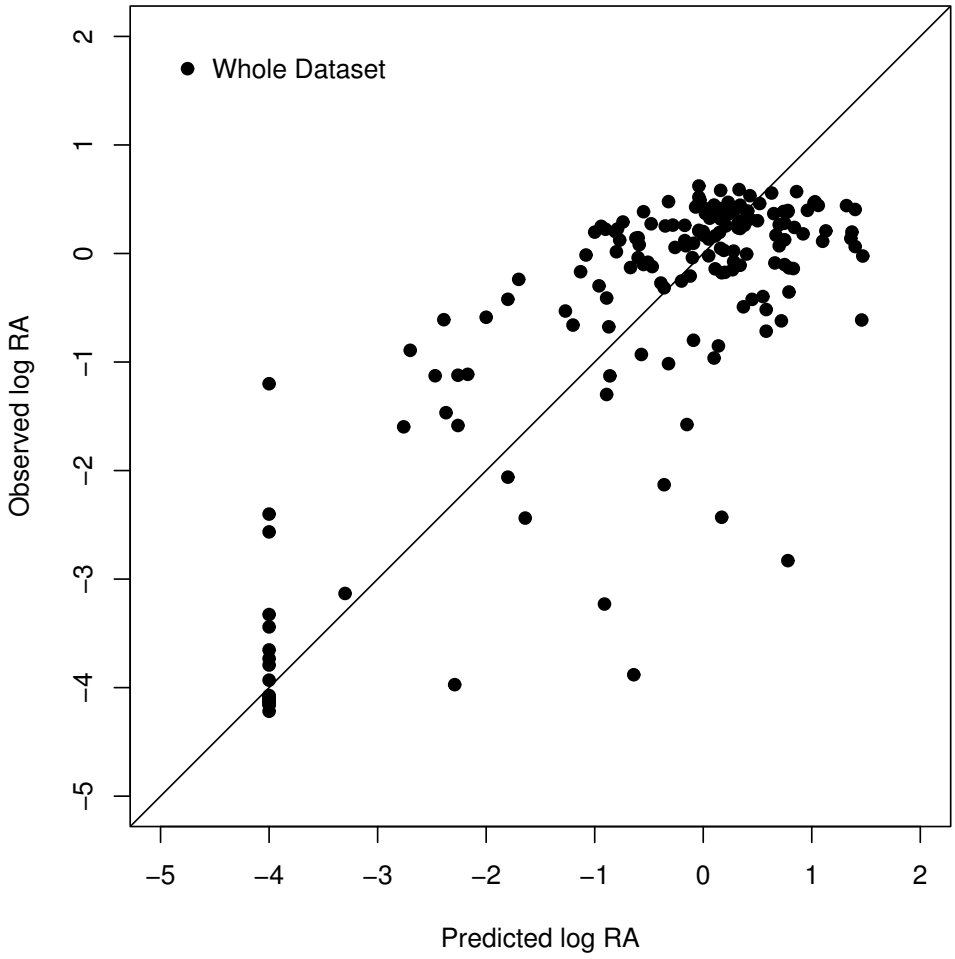


Fig. 6.13. A plot showing the predicted versus observed log RA values for the whole dataset using the best nonlinear CNN model (10–5–1 architecture). This result was obtained by a leave 14% out procedure which was run 3 times giving 3 predictions for each member of the dataset. The average value was taken as the final predicted value.

References

[1] Haynes, R. K.; Vonwiller, S. C. From Qinghao, Marvelous Herb of Antiquity, to the Antimalarial Trioxane Qinghaosu - and Some Remarkable New Chemistry. *Acc. Chem. Res.* **1997,** *30,* 73–79.

[2] Klayman, D. Qinghaosu (Artemisinin): An Antimalarial Drug from China. *Science* **1985,** *228,* 1049.

[3] Kamchonwongpaisan, S.; Meshnick, S. The Mode of Action of the Antimalarial Artemisinin and its Derivatives. *Gen. Pharmac.* **1996,** *27,* 587–592.

[4] Posner, G.; Cumming, J.; Ploypradith, P.; Oh, C. Evidence for Fe(IV)-O in the Molecular Mechanism of Action of the Trioxane Antimalarial Artemisinin. *J. Am. Chem. Soc.* **1995,** *117,* 5885–5886.

[5] Posner, G.; Park, S.; Gonzalez, L.; Wang, D.; Cumming, J.; Klinedinst, D.; Shapiro, T.; Bachi, M. Evidence for the Importance of High Valent Fe-O and of a Diketone in the Molecular Mechanism of Action of Antimalarial Trioxane Analogs of Artemisinin. *J. Am. Chem. Soc.* **1996,** *118,* 3537–3538.

[6] Robert, A.; Meunier, B. Is Alkylation the Main Mechanism of Action of the Antimalarial Drug Artemisinin?. *Chem. Soc. Rev.* **1998,** *27,* 273–274.

[7] Avery, M.; McLean, G.; Edwards, G.; Ager, A. Structure Activity Relationships of Peroxide Based Artemisinin Antimalarials. *Biol. Act. Nat. Prod.* **2000,** 121–132.

[8] Woolfrey, J.; Avery, M.; Doweyko, A. Comparison of 3D Quantitative Structure-Activity Relationship Methods: Analysis of the In Vitro Antimalarial Activity of 154 Artemisinin Analogues by Hypothetical Active Site Lattice and Comparitive Molecular Field Analysis. *J. Comput. Aided Mol. Des.* **1998,** *12,* 165–181.

[9] Avery, M.; Alvim-Gaston, M.; Woolfrey, J. Synthesis and Structure-Activity Relationships of Peroxidic Antimalarials Based On Artemisinin. *Adv. Med. Chem.* **1999,** *4,* 125–217.

[10] Avery, M. A.; Alvim-Gaston, M.; Rodrigues, C. R.; Barreiro, E. J.; Cohen, F. E.; Sabnis, Y. A.; Woolfrey, J. R. Structure Activity Relationships of the Antimalarial Agent Artemisinin. the Development of Predictive in Vitro Potency Models Using CoMFA and HQSAR Methodologies. *J. Med. Chem.* **2002,** *45,* 292–303.

[11] Tommuphean, S.; Kokpol, S.; Parasuk, V.; Wolschann, P.; Winger, R.; Liedl, K.; Rode, B. Comparative Molecular Field Analysis of Artemisinin Derivatives: Ab Intio versus Semi Empirical Optimized Structures. *J. Comput. Aided Mol. Des.* **1998,** *12,* 397–409.

[12] Avery, M.; Gao, F.; Wesley, C.; Mehrotra, S.; Milhous, W. Structure-Activity Relationships of the Antimalarial Agent Artemisinin. 1. Synthesis and Comparative Molecular Field Analysis of C-9 Analogs of Artemisinin and 10-Deoxoartemisinin. *J. Med. Chem.* **1993,** *36,* 4264–4275.

[13] Cheng, F.; Shen, J.; Luo, X.; Zhu, W.; Gu, J.; Ji, R.; Jiang, H.; Chen, K. Molecular Docking and 3D-QSAR Studies On the Possible Antimalarial Mechanism of Artemisinin Analogues. *Bioorg. Med. Chem.* **2002,** *10,* 2883–2891.

[14] Girones, X.; Gallegos, A.; Carbo-Dorca, R. Modeling Antimalarial Activity: Application of Kinetic Energy Density Quantum Similarity Measures as Descriptors in QSAR. *J. Chem. Inf. Comput. Sci.* **2000,** *40,* 1400–1407.

[15] Tonmunphean, S.; Parasuk, V.; Kokpol, S. QSAR Study of Antimalarial Activities and Artemisinin-Heme Binding Properties Obtained from Docking Calculations. *Quant. Struct.-Act. Relat.* **2000,** *19,* 475–483.

[16] Cramer III, R.; Patterson, D.; Bunce, J. Comparative Molecular Field Analysis (CoMFA). I. Effect of Shape on Binding of Steroids to Carrier Protiens. *J. Am. Chem. Soc.* **1988,** *110,* 5959–5967.

[17] Cramer III, R.; Patterson, D.; Bunce, J.; Frank, I. Crossvalidation, Bootstrapping and Partial Least Squares Compared with Multiple Regression in Conventional QSAR Studies. *Quant. Struct-Act. Relat. Pharmacol.,Chem. Biol.* **1988,** *7,* 18–25.

[18] Jurs, P.; Chou, J.; Yuan, M. Studies of Chemical Structure Biological Activity Relations Using Pattern Recognition. In *Computer Assisted Drug Design*; Olsen, E.; Christoffersen, R., Eds.; American Chemical Society: Washington D.C., 1979.

[19] Stuper, A.; Brugger, W.; Jurs, P. *Computer Assisted Studies of Chemical Structure and Biological Function;* Wiley: New York, 1979.

[20] Sutter, J.; Dixon, S.; Jurs, P. Automated Descriptor Selection for Quantitative Structure-Activity Relationships Using Generalized Simulated Annealing. *J. Chem. Inf. Comput. Sci.* **1995,** *35,* 77–84.

[21] Goldberg, D. *Genetic Algorithms in Search Optimization & Machine Learning;* Addison Wesley: Reading, MA, 2000.

[22] Kier, L.; Hall, L.; Murray, W. Molecular Connectivity I: Relationship to Local Anesthesia.. *J. Pharm. Sci.* **1975,** *64,*.

[23] Kier, L.; Hall, L. Molecular Connectivity VII: Specific Treatment to Heteroatoms. *J. Pharm. Sci.* **1976,** *65,* 1806–1809.

[24] Kier, L.; Hall, L. *Molecular Connectivity in Structure Activity Analysis.;* John Wiley & Sons: Hertfordshire, England, 1986.

[25] Randic, M. On Molecular Idenitification Numbers. *J. Chem. Inf. Comput. Sci.* **1984,** *24,* 164–175.

[26] Liu, S.; Cao, C.; Li, Z. Approach to Estimation and Prediction for Normal Boiling Point (NBP) of Alkanes Based On A Novel Molecular Distance Edge (MDE) Vector, λ. *J. Chem. Inf. Comput. Sci.* **1998,** *38,* 387–394.

[27] Rousseeuw, P.; Leroy, A. *Robust Regression and Outlier Detection;* Wiley Series in Probability and Mathematical Statistics John Wiley & Sons: Hoboken, New Jersey, 1987.

[28] R Development Core Team, "R: A Language and Environment for Statistical Computing", R Foundation for Statistical Computing, Vienna, Austria, 2004 ISBN 3-900051-07-0.

[29] Stanton, D. On the Physical Interpretation of QSAR Models. *J. Chem. Inf. Comput. Sci.* **2003,** *43,* 1423–1433.

[30] Minitab, "Minitab", 2003.

[31] Kier, L. A Shape Index from Molecular Graphs. *Quant. Struct-Act. Relat. Pharmacol.,Chem. Biol.* **1985,** *4,* 109–116.

[32] Kier, L. Shape Indices for Orders One and Three from Molecular Graphs. *Quant. Struct-Act. Relat. Pharmacol.,Chem. Biol.* **1986,** *5,* 1–7.

[33] Kier, L. Distinguishing Atom Differences in a Molecular Graph Index. *Quant. Struct-Act.!Relat.* **1986,** *5,* 7–12.

[34] Goldstein, H. *Classical Mechanics;* Addison Wesley: Reading, MA, 1950.

[35] Kier, L.; Hall, L. *Molecular Connectivity in Chemistry and Drug Research;* Academic Press: New York, 1976.

[36] Balaban, A. Highly Discriminating Distance Based Topological Index. *Chem. Phys. Lett.* **1982,** *89,* 399–404.

[37] Stanton, D.; Jurs, P. Development and Use of Charged Partial Surface Area Structural Descriptors in Computer Assissted Quantitative Structure Property Relationship Studies. *Anal. Chem.* **1990,** *62,* 2323–2329.

[38] Todeschini, R.; Consonni, V. *Handbook of Molecular Descriptors;* Methods and Principles in Medicinal Chemistry Wiley-VCH: Weinheim, 2000.

[39] Gupta, A.; Park, S.; Sluwa, M. Generalized Analytic Rule Extraction for Feedforward Neural Networks. *IEEE Trans. Knowl. Data Eng.* **1999,** *11,* 985–991.

[40] Ney, H. On the Probabilistic Interpretation of Neural Network Classifiers and Discriminative Training Criteria. *IEEE Trans. Pat. Anal. Mach. Intell.* **1995,** *17,* 107–119.

[41] Schmitz, G.; Aldrich, C.; Gouws, F. ANN-DT: An Algorithm for Extraction of Decision Trees from Artificial Neural Networks. *IEEE Trans. Neural. Net.* **1999,** *10,* 1392–1401.

[42] Zurada, J. M.; Setiono, R.; Leow, W. K. "Extraction of Rules from Artificial Neural Networks for Nonlinear Regression", 2001.

Chapter 7

The Development of Linear, Ensemble and Nonlinear Models for the Prediction and Interpretation of the Biological Activity of a Set of PDGFR Inhibitors

7.1 Introduction

The investigation of anti-cancer drugs has focussed on a number of targets. One of the initial focus areas was compounds that could interfere in DNA synthesis and function and as a result, stimulate apoptotic pathways. Such a self-destructive approach is limited in terms of efficiacy and selectivity. An alternative approach that has been the target of intense research is the development of compounds that are able to interfere with cellular signal transduction mechanisms. Cell growth is one area in which signal transduction plays a vital role. Essentially, growth factors bind to specific cell surface receptors initiating a cascade of events which lead to activation of genes or other growth mechanisms. An important class of growth receptors are the receptor tyrosine kinases (RTK's). This class of kinase is a member of a family known as protein tyrosine kinases which transmit growth signals via a phosphorylation mechansim.[1] The structures of RTK's consist of three parts - a ligand binding region on the cell membrane, a region spanning the cell membrane and tyrosine kinase domains within the cell.[2–4] Four main RTK's are known, and platelet derived growth factor receptor (PDGFR) is the RTK that is considered in this chapter.

A large number of compounds have been investigated as putative PDGFR inhibitors. Examples include 1–phenylbenzimidazoles,[5] arylquinoxalines,[6] piperazinylquinazolines[3] and various pyrimidine analogs.[7–9] The mode of action of PDGFR inhibitors is competition with ATP binding at the intra-cellular kinase domains. Thus, the biological activity of prospective inhibitors can be investigated with phosphorylation assays. Much experimental work has been carried out on this family of proteins and a number of QSAR

This work was published as Guha, R.; Jurs, P.C., "The Development of Linear, Ensemble and Non-linear Models for the Prediction and Interpretation of the Biological Activity of a Set of PDGFR Inhibitors", *J. Chem. Inf. Comput. Sci.*, **2004**, *44*, 2179–2189.

studies have been carried out as well. Kurup et al.[1] conducted an extensive review of QSAR models for tyrosine kinase inhibitors (including PDGFR). All the models reported were linear in nature and were developed using a limited number of descriptors. Shen et al.[10] developed a series of linear regression models for the set of 1- phenylbenzimidazoles described by Palmer[5] using electronic descriptors and a PLS routine to build the final models.

This study involves the development of a set of linear as well as nonlinear models to predict and interpret the biological activity of a set of piperazinylquinazolines investigated by Pandey et al.[3] The dataset consisted of 79 compounds with the biological activity reported as IC_{50} values. Activity values were obtained from a phosphorylation assay with and without human plasma. The original study investigated the structure-activity trend of these compounds experimentally, but no computational models were developed. We note that Khadikar et al[11] have reported a QSAR study using this dataset. However, their study was restricted to linear regression models using topological descriptors only. Furthermore, they restricted themselves to using IC_{50} values from the assay in the absence of human plasma. The models we present concentrate on the biological activity values obtained from the assay in the presence of human plasma. Furthermore we present results from linear regression models as well as nonlinear computational neural network models. We used a wide variety of descriptors rather than restricting ourselves to any single class. Finally, in addition to prediction, the linear model was analysed using the PLS interpretation method to explain the structure-activity trends embodied in it.

7.2 Dataset

The dataset consisted of 79 compounds that were derivatives of 4–piperazinylquinazolines and were investigated for their ability to inhibit PDGFR phosphorylation.[3] The structures of these compounds have been presented by Pandey et al.[3] The compounds were evaluated for their inhibition of PDGFR phosphorylation in MG63 cells.[3,12] The assays were carried out both in the presence and absence of human plasma resulting in two sets of IC_{50} values. For the purposes of this study, these were converted to $-\log(IC_{50})$ values. However a number of the measurements made in the absence of plasma were reported as < 0.004. Since this indicates that the response was possibly below the limit of detection, these compounds would have to be ignored for the purposes of model building, thus decreasing the size of the dataset. Hence we only considered the

set of measurements made in the presence of human plasma thus allowing the use of all 79 compounds.

7.3 Methodology

This study used the ADAPT[13,14] package to calculate descriptors and develop QSAR models. As described previously, the ADAPT methodology allows for the development of linear models and nonlinear models. In addition a random forest[15] model was built using the R software package.[16] A brief overview of the aspects of the methodology specific to this study are dicussed below and further details are presented in Chapter 3.

The first step was to divide the compounds into the three sets - the training, cross-validation and prediction set (known as QSAR sets) using an activity binning method. This resulted in a training set containing 57 compounds, a cross-validation set containing 9 compounds and a prediction set containing 13 compounds. Next, the 3-D molecular structures were entered using Hyperchem[17] and geometry optimized using MOPAC 7.01 with the PM3 Hamiltonian. After structures were optimized molecular descriptors were calculated resulting in a total of 321 descriptors. This descriptor pool was reduced using objective feature selection. Setting correlation and identical cutoffs of 0.75 and 0.75, respectively, this procedure resulted in a reduced pool of 41 descriptors. This descriptor pool was used to generate the linear and nonlinear models discussed in this chapter.

The next stage was subjective feature selection in which a simulated annealing algorithm[18] or a genetic algorithm[19,20] was employed to search for optimal descriptor subsets to build linear and nonlinear models. For both types of models a number of candidate models were generated. In the case of the linear regression model, the final model selected was the one that had the lowest RMS error. In the case of the CNN models, the set of models for descriptor subsets of a given size were analyzed more rigorously to determine the optimal architecture. The final descriptor subset and architecture selected was the one that had the lowest value of the cost function described in Section 3.4.2.

The random forest technique has been discussed in detail in Section 2.3.1. In this study we did not deviate significantly from the default settings of the random forest implementation in the R software package. We focused on the number of descriptors randomly selected to split nodes on, and the minimum node size (that is, the minimum number of members in a node, below which a node is not split). In general the defaults in the R implementation of the random forest algorithm lead to good models. However, we

performed a grid search to find optimal values of the parameters using the tune function from the e1071 package in R.

7.4 Results

7.4.1 Linear Models

A series of linear models were developed using the genetic algorithm to search for optimal descriptor subsets. A training set of 68 compounds was used initially. The best model obtained was a 9-descriptor model. However, it exhibited poor regression statistics (no t-values were greater than 3.0 and p-values for the coefficients were on the order of 0.01). Furthermore, none of the models except the 9-descriptor model were validated when investigated using a PLS analysis. A 3-descriptor model with similar statistics but a much lower R^2 and RMSE was also investigated. One aspect of these two models, as well as nearly all the models developed using the GA, was that three compounds (**23**, **83** and **90**) were consistently flagged as training set outliers. Outliers were detected by plotting studentized residuals versus the compound index for each of the linear models developed. An example of the residual plot for the 3-descriptor model is shown in Figure 7.1. Apart from the compounds mentioned above, some models usually had one or two other compounds which could be considered as borderline outliers. Since these borderline compounds varied from model to model, we did not consider them further. Since the three outliers mentioned above were found in nearly all the models that were generated, we felt it was justified to remove them from the training pool and to reexamine the models. Thus, the training set was reduced to 65 compounds. One common feature of these compounds that may justify their removal is that the 6 position on the quinazoline ring in these compounds has an ethoxy group (in the case of **83** and **90**) or a hydrogen (in the case of **23**), whereas the majority of compounds have longer (bulkier) functional groups at this position. Furthermore in the case of **83** and **90**, the 7 position does have a ring moiety at the end of the 4-membered chain and thus can be considered relatively bulky. However as will be discussed later, such a feature (bulky groups attached to long chains at the 6 and 7 positions on the quinazoline moiety) is characteristic of compounds with high activity whereas these compounds have quite low values of activity. This observation is supported to some extent by the fact that in Figure 7.1, compounds **83** and **90** are significantly more outlying than **23**. The statistics of all the models improved and most models were validated using the PLS technique (including the 9- and 3-descriptor models mentioned previously). Since the aim of a modeling technique

is parsimony, we chose to present the results, and an interpretation of, the 3-descriptor model.

A plot of the observed versus predicted $-\log(\mathrm{IC}_{50})$ values for the 3-descriptor model (with training set outliers removed) is shown in Figure 7.2. The statistics of the model are summarized in Tables 7.1 and 7.2. The ranges of the descriptors used are shown in Table 7.3. The R^2 for the model was 0.65 and the RMSE was 0.38. The value of the F-statistic was 37.06 (on 3 and 59 degrees of freedom) compared to a critical value of 2.76 (at the 0.05 significance level) with a p-value of 1.4×10^{-13}. Finally the variance inflation factors for all the descriptors was less than 1.6 indicating the absence of collinearities in the model. For the prediction set the R^2 was 0.38 and the RMSE was 0.47. Though the RMSE is not significantly higher than for the training set, the low value of R^2 is influenced by the prediction set outlier noted in Figure 7.2. Removal of this compound (**55**) from the prediction set resulted in a R^2 of 0.84 and RMSE of 0.24. The structure of this outlier is shown in Figure 7.3. A simple comparison of the structure of this outlier with other structures in the dataset does not reveal why it would be predicted poorly. However the PLS analysis of this model, described below, does shed some light on the behavior of this compound in the linear model.

The three descriptors used in the model were MDEN-23, RNHS-3 and SURR-5. The MDEN-23 descriptor is the molecular distance edge vector[21] between secondary and tertiary nitrogens. The descriptor is defined as the geometric mean of the topological path lengths between secondary and tertiary nitrogens. The original implementation of this descriptor only considered carbons and can be interpreted as characterizing the extension of side chains from the main body of a molecule.[22] The characteristic feature of the compounds in this study is that they all contain a piperazine and pyrimidine substructure. The two substructures are connected via the nitrogen on the piperazine group. As a result the MDEN-23 descriptor captures the linkage between the two rings. Furthermore, a number of compounds have side groups containing secondary and (or) tertiary nitrogens (examples include compounds **5**, **32** and **54**). The MDEN-23 descriptor thus characterizes the "nitrogen backbone" of these compounds. For the compounds in this study, tertiary nitrogens were generally members of cycles and all compounds had central pyrimidine and piperazine rings. As a result, larger values of this descriptor indicate the presence of cyclic and non-cyclic side chains containing nitrogen.

The RNHS-3 descriptor is a hydrophobic surface area (HSA) descriptor developed by Stanton et al.[23] It is defined as

$$\frac{\max(SA^{-})\sum H_i}{\log P} \tag{7.1}$$

where $\max(SA^{-})$ is the surface area of the most hydrophilic atom, H_i are the hydrophilic constants (which are values of Wildman and Crippens[24] atomic hydrophobicity constants that are less than 0) and $\log P$ is the logarithm of the octanol-water partition coefficient. Thus, this descriptor is a measure of the relative hydrophilic surface area of a molecule. The presence of this descriptor in the model is not surprising considering that all compounds in the study contain three or more nitrogens along with oxygens in a number of cases.

The SURR-5 descriptor is a modification of the HSA descriptor described by Mattioni.[25] The original HSA descriptors classified atoms as either hydrophilic or hydrophobic using the atomic hydrophobicity constants of Wildman and Crippen.[24] In the modified version hydrophobic atoms are divided into *low hydrophobic* (atoms with hydrophobic constants between 0 and 0.4) and *high hydrophobic* (atoms with hydrophobic constants greater than 0.4). The modification increases the differentiability of the HSA descriptors and has been shown to be effective in structure-activity studies.[25] SURR-5 is defined as the ratio of the atomic constant weighted hydrophobic (low) surface area and the atomic constant weighted hydrophilic surface area. This descriptor thus characterizes the various portions of the molecular surface in terms of hydrophobicity and hydrophilicity. Absolute values greater than one indicate that the molecular surface is mainly hydrophobic, and values less than one indicate that the molecular surface is mainly hydrophilic.

To ensure that the results described above did not arise by chance, randomized runs were carried out. A randomized run consisted of scrambling the dependent variable and building the model using the same descriptors as in the original model. This procedure was repeated 500 times and the average values of the R^2 and RMSE were calculated for both the training and prediction sets. It is expected that if a true structure-activity relationship is captured by the original model, the randomized models should exhibit lower values of R^2 and higher values of RMSE when compared to the original model. The results from our runs indicate this to be the case. The average value of R^2 and RMSE for the training set was 0.05 and 0.72, respectively. For the prediction set they were 0.08 and 1.04, respectively. The statistics of the randomized runs are summarized

in Table 7.4. It should be noted that in all the runs compound **55** was not removed from the prediction set.

The 3-descriptor, linear model was then subjected to a PLS analysis to provide an interpretation of the structure-activity relationship embodied by the model. This technique has been described by Stanton[26] and the details of the interpretation methodology was presented in Section 3.6. A number of examples of this technique have been reported.[22,26] The PLS analysis was carried out with Minitab[27] using a leave-one-out cross-validation scheme. The results of the PLS analysis indicated that all 3 components were validated and thus the model was not overfit. A summary of the statistics for the 3 components are shown in Table 7.5. Table 7.6 shows the X-weights for the 3 PLS components. The X-weights for a given component indicate the contributions of each descriptor to that component. As can be seen, in each component one descriptor has a very high absolute value and thus is the main contributor to that component. We consider each component separately and use the weights and the score plots (Figures 7.4, 7.6 and 7.8) to interpret the structure-activity trend characterized by the model.

The most heavily weighted descriptor in PLS component one is SURR-5. As can be seen, its weight is significantly higher than the other two descriptors and thus plays an important role. Figure 7.4 shows the score plot for the first PLS component. Points in the upper right and lower left are correctly predicted as active and inactive compounds respectively. The structures of some representative active and inactive compounds for this component are compared in Figure 7.5. Compounds **75**, **84**, **86** and **87** are regarded as active and they are characterized by high absolute values of the SURR-5 descriptor. From the description of the SURR-5 descriptor, this indicates that active compounds are characterized by a large hydrophobic surface area. This is consistent with the fact that the cell based assay used by Pandey et al.[3] reports the activity of the compounds against the kinase target modulated by their ability to pass through the cell membrane. Clearly, compounds with a higher proportion of hydrophobic surface area would have a better ability to enter the cell. Component 1 does not under-predict any compounds as shown by the empty upper left corner. However, compounds **11**, **21**, **30** and **55** are over-predicted by this component. An interesting point to note is that compound **55** which was a significant outlier in the linear model (and is also an outlier in the nonlinear CNN model) has a high absolute value of the SURR-5 descriptor but has a low observed activity (-0.39 $-\log(\mathrm{IC}_{50})$ units). As a result this compound does not follow the general structure-activity trend for the SURR-5 descriptor. As will be shown in the results for the random forest, the SURR-5 descriptor is a very significant descriptor. Since **55** does not

follow the trend for this descriptor this explains to some extent its position as an outlier. Compounds **50**, **91** and **93** are predicted correctly as inactive and are characterized by low absolute values of the SURR-5 descriptor. Considering the structures shown in Figure 7.5 it is clear that the piperazinylquinazoline backbone is common to both active and inactive structures. The active structures shown (as well as in nearly all the active compounds for this component) all have a bulky hydrophobic group linked to the 7 position on the quinazoline ring. However, compound **50** has a piperazine ring linked to the 7 position but exhibits a low activity. This can be understood by considering the molecular surfaces. Figures 7.9, 7.10 and 7.11 show molecular surfaces for compounds **75**, **93** and **50** colored by hydrophobicity values, drawn using PyMOL.[28] Blue regions indicate areas of high hydrophilicity and red regions indicate areas of high hydrophobicity. The bulky piperidine group in **75** is largely hydrophobic compared to the trimethyl amine group in **93** which has a distinct hydrophilic center. In light of these observations, the surface of **50** shows that the amide center on the piperazine ring creates a large hydrophilic center and thus is similar in this respect to **93**. One would thus expect that activity would be improved by having bulky groups without hydrophilic centers connected to the 6 or 7 position on the quinazoline ring.

The most heavily weighted descriptor in PLS component 2 is MDEN-23. Figure 7.6 shows the score plot for the second PLS component. Compounds predicted correctly as active (**8**, **18** and **19**) exhibit very high values of this descriptor whereas compounds predicted correctly as inactive (**54**, **66**, **94** and **100**) exhibit smaller values. Large values of this descriptor are characterized by a larger number of longer paths between secondary and tertiary nitrogens. This may be indirectly interpreted as a count of nitrogens. Pandey et al.[3] mention that in several cases removing basic groups (such as secondary amines in this case) greatly reduces potency. Thus, larger numbers of secondary nitrogens would enhance the activity of potential inhibitors. Another aspect of this descriptor that has been described previously is that it may be interpreted, in the case of the current dataset, as an indicator of nitrogen containing rings separated by long paths. This would imply that compounds with large cyclic side chains connected to the backbone via long chains would exhibit higher values of this descriptor. The structures of some of the active and inactive compounds are shown in Figure 7.7. It is evident that the active compounds have bulky nitrogen containing side groups on the phenoxy ring. In the case of the compounds shown here it is an indole group. In the case of the inactive compounds these are absent. This confirms the observations made by Matsuno[29] and Pandey[3] that bulky hydrophobic side groups along with electron donating centers

enhance activity. However, **54** does appear to be anomalous in that it does contain a relatively hydrophobic side group (attached to the quinazoline ring) yet is inactive.

Once again the importance of the SURR-5 descriptor is evident as the second component under-predicts a large number of active compounds which were correctly predicted by component 1. However, component 2 corrects for the over-prediction of some of the compounds from component 1. As can be seen from the score plot in Figure 7.6, compounds **11**, **21**, **30** and **55** are now shifted towards the lower left. Thus, this component compensates for the over-prediction of these compounds by component 1 by taking into account bulky hydrophobic groups attached to the phenyl ring. It should be noted that though **55** is predicted relatively better in this component than the previous one, it is still midway between the two lower quadrants. However, it does follow the trend for the MDEN-23 descriptor (i.e., lower values indicate lower activities) better than for the SURR-5 descriptor

Finally, we consider PLS component 3. Table 7.5 shows that the increase in R^2 gained by adding component 3 to the model is only 0.01. Thus, it is expected that this component will not be able to explain any significant structure-activity trend described by the most heavily weighted descriptor (RNHS-3). As can be seen from the score plot (Figure 7.8), this component does not predict any low activity compounds. Furthermore the under-predicted compounds (**93** and **91**) have already been correctly predicted as inactive by component 1 and the over-predicted compounds in the lower right corner were also correctly predicted as moderately inactive by components 1 and 2. However this component does contribute to the structure-activity relationship to some extent by correctly predicting compounds **47** and **96** as active whereas they were under-predicted by component 2.

Combining the two main trends discussed in this section, we see that there is a competition between a requirement for bulky hydrophobic side groups and higher numbers of nitrogens (which create hydrophilic centers). The fact that component 1 explains the majority of the structure-activity trend implies that the latter requirement plays a stronger role. Thus it may be expected that compounds with a piperazinylquinazoline backbone would exhibit increased activity by having bulky hydrophobic nitrogen containing groups attached to the phenyl moiety as well as at the quinazoline moiety. Furthermore, bulk may be increased at the quinazoline moiety by attaching side groups at both the 6 and 7 positions. This would imply that the groups would have to be bonded by relatively long paths to the 6 and 7 positions to avoid steric hindrance. Assuming that the linker groups contain nitrogen, this would result in larger values of the MDEN-23

descriptor for those compounds. And as has been shown, large values of this descriptor correlate with higher activities.

As noted before, this dataset had been studied by Khadikar[11] who developed a set of linear regression models. However their methodology differed significantly in that they used the compounds with reported activities in the absence of human plasma. As a result this restricted the size of the dataset. Furthermore the linear models were developed after removing 10 compounds from the already reduced dataset. Finally, their models were developed using a stepwise linear regression technique which is not necessarily an efficient way to search for optimal descriptor subsets.[30,31] The best linear model reported in this work exhibits a lower value of R^2 than the corresponding 3-descriptor model reported by Khadikar. However, considering the fact that this statistic is well known to be misleading, and the fact that we used a larger dataset, we believe that the lower value of R^2 for our model does not detract from its main utility as an interpretive model. Furthermore, the descriptors present in our best linear model allow a clear interpretation of the structure-activity trend which confirms observations made by Pandey et al.[3] The topological descriptors present in the model described by Khadikar do not lend themselves to a detailed interpretation.

7.4.2 Nonlinear CNN Models

Nonlinear CNN models were developed by using the CNN routine as the objective function for the genetic algorithm. The full training set of 57 compounds was used. For a given CNN architecture the descriptor space was searched for subsets that lead to CNN models with low values of the cost function described in Section 3.4.2. Once a number of suitable subsets were found, the number of hidden layer neurons were varied to determine the optimal CNN architecture. This procedure resulted in a 7–3–1 CNN model. The statistics of the model are given in Table 7.7. A comparison of the statistics in Tables 7.7 and 7.2 clearly indicate the improved performance of the nonlinear CNN model compared to the linear model. The seven descriptors present in the model are N5CH,[32–34] WTPT-3,[35] WTPT-4,[35] FLEX-4, RNHS- 3,[23] SURR-5[23] and APAVG. It should be noted that two of the descriptors (RNHS-3 and SURR-5) are also present in the best linear model. N5CH is the number of 5^{th} order chains which are defined as a sequence of 5 atoms containing a ring. This definition thus includes 5-membered rings, 4-membered rings with a methyl side chain and a 3-membered ring with an ethyl side chain. The WTPT descriptors are based on Randic's molecular ID and are termed weighted path

descriptors. They combine features of connectivity indices[32–34] and path counts and are independent of molecular geometry. WTPT-3 considers all weighted paths starting from any heteroatom and WTPT-4 considers weighted paths starting only from oxygen atoms. The FLEX-4 descriptor characterizes conformational flexibility. More specifically this descriptor evaluates the fractional mass of the rotatable atoms. RNHS-3 and SURR-5 have been described previously. Finally, the APAVG descriptor is based on atom pairs as defined by Carhart et al.[36] The atom pair method describes molecular features by considering pairs of atoms together with the path between them. As a result, a given molecule will have a set of atom pair strings which contain the start and end atom types and the path length between them. These atom pair strings can be hashed to give a 32 bit number which have been used as a similarity measure. APAVG is defined as the average of the atom pair hash values.

Figure 7.12 shows a plot of the predicted versus observed $-\log(\mathrm{IC}_{50})$ values from the CNN model. It is encouraging to see that the performance of the nonlinear model was very good on the training set as shown the RMSE and R^2 values. The plot is also substantially less scattered than the corresponding plot for the linear model. As noted on the plot, there are two possible prediction set outliers. When compound **55** was removed from the prediction set and the remaining compounds were processed by the model, the R^2 value for the prediction set rose to 0.72 and the RMSE decreased to 0.27.

As in the case of the linear model, the nonlinear CNN model was also tested for random correlations. As before, the dependent variable was scrambled and the CNN model rebuilt. The procedure was repeated 100 times and the averages of the RMSE and R^2 values are reported in Table 7.8. As can be seen the average RMSE is more than triple that of the original runs. The average values of R^2 are also very poor. These results indicate that chance played very little role in the performance of the CNN model.

7.4.3 Random Forest Model

The linear and nonlinear models presented so far have two descriptors in common, RNHS-3 and SURR-5. We also note that using the genetic algorithm resulted in a large number of linear and nonlinear models which contained these descriptors. SURR-5 was present in more than 90% of the models evaluated. Clearly, this descriptor must be information rich. The role played by this descriptor in the linear model has been analyzed using PLS and was described above.

We built a random forest model to investigate whether it would provide any further information regarding the importance of descriptors, specifically SURR-5. As mentioned previously random forest parameters were tuned using a grid search and the final forest was built with 500 trees, a node size of 5, and 13 descriptors were used at each split point. The model was built using all the compounds in the dataset and the entire reduced pool of 41 descriptors. The predictive ability of this model was not significantly better than the linear regression or nonlinear CNN models. However our main focus was on the importance ascribed to specific descriptors by the random forest model. The procedure by which descriptor importances are obtained from a random forest model has been described in Section 2.3.1. Figure 7.13 shows a plot of descriptor importance (only the 10 most important descriptors are shown, ranked in decreasing order of importance).

It is clear that SURR-5 is deemed to be the most important descriptor. Interestingly, RNHS-3 and MDEN-23 are ranked relatively low. Furthermore, the PLS analysis indicated that for the linear regression model, MDEN-23 was able to account for more of the structure-activity trend compared to RNHS-3. From Figure 7.13 it is clear that the increase in MSE is not very large in going from MDEN-23 to RNHS-3. At the same time it should be noted that the algorithms underlying PLS and random forests are substantially different. Most importantly, the random forest is working with the whole reduced pool (41 descriptors) and thus it is able to compare and contrast more descriptors than considered in the PLS analysis. Thus a relationship detected by a PLS analysis will not necessarily show up in a random forest. However it is encouraging that the most important descriptor from the random forest model describes the majority of the structure-activity trend in the PLS analysis. We also note that the CNN model contains the two most important descriptors, as identified by the random forest. Furthermore the remaining descriptors in the CNN model are present in the top 20 descriptors, as measured by the random forest. This is not surprising as the CNN model is built by allowing the GA to search for the best 7-descriptor subset from the whole, 41-descriptor, reduced pool. Once again, a direct correspondence between descriptors is not expected due to the different algorithms underlying the respective models.

The above discussion indicates the relative importance of the SURR-5 descriptor in both linear and nonlinear models. Since SURR-5 describes the hydrophobicity of a surface we investigated its relation to the $\log P$ values of the compounds. The $\log P$ values were calculated using a fragment based approach developed by Mattioni for the HSA descriptors mentioned earlier. A scatter plot of $\log P$ versus SURR-5 for the dataset showed no distinct correlations ($R^2 = 0.17$). We also made scatter plots of $\log P$ versus

the other descriptors, and none of them showed any correlations (R^2 ranging from 0.01 to 0.20) except in the case of RNHS-3. However this is to be expected as the functional form of this descriptor includes the $\log P$ value of the compounds.

We also investigated whether the most important descriptors from the random forest model would lead to good linear or CNN models. We evaluated a regression model and carried out a PLS analysis using the top three descriptors but the RMSE and R^2 were poorer than those reported for the best linear model. Even though the PLS analysis validated all 3 descriptors, the total R^2 explained was less than for the best model. The descriptors were also used in CNN models. Three architectures were investigated, 3–2–1, 3–3–1 and 3–4–1. However none of the models performed significantly better than the reported model.

7.5 Conclusions

The results presented in this chapter indicate that the linear regression and CNN models developed during this study, exhibit interpretability as well as predictive ability. Though the linear model was developed mainly for purposes of structure-activity interpretation, removal of one prediction set outlier improved its predictive ability drastically. The application of a PLS analysis allows for the interpretation of the structure-activity trends embodied in the model. The interpretation clearly indicates the importance of the hydrophobic surface area descriptor, SURR-5. This is also confirmed by the random forest model which provides a measure of descriptor importance. The model ranked SURR-5 as the most important descriptor. However, the other descriptors in the linear are also relatively important with respect to the whole descriptor pool. The main conclusions from the PLS interpretation indicate bulky hydrophobic groups and nitrogen centers increase activity. These observations have been made experimentally, thus supporting our theoretical model. As noted before, these two trends compete against each other. However, the PLS and random forest results also indicate the relatively more important role of hydrophobic groups. The CNN model was developed primarily for predictive ability as such models are generally not amenable to interpretation.[22] It exhibited good statistics for both training and prediction. Furthermore it also contained the top two descriptors, as identified by the random forest, including SURR-5, once again underlying the importance of this descriptor to the structure-activity relationship.

An interesting extension to this work would be to develop a 3-D QSAR model using CoMFA[37,38] which would allow a more detailed view of the specific interactions that

are described by our 2-D models. The predictions described in the preceding sections are based on the correlation of molecular descriptors to experimental activity and thus may be considered relatively abstract. That is, the 2-D methodology we employ cannot provide a direct view of the binding between these compounds and the PDGF receptor, and hence inhibitory activity. This implies that any conclusions made on the basis of our models are oriented towards the activity value rather than activity mechanism (via binding features). A 3-D method such as CoMFA would allow for a more direct understanding of the interactions of the compounds considered here with the PDGF receptor. In addition, a CoMFA model would allow for the prediction of binding energies. Combined with a systematic modification of the side groups at the 6 and 7 positions in-silico, this would allow not only confirmation of the experimental data described here, but could also be used as a stepping stone to the synthesis of more potent inhibitors. The fundamental requirement for such a study would the crystal structure of PDGFR. The crystal structures of tyrosine kinase receptors related to the PDGF receptor have been reported[39,40] though we are not aware of crystal structures of the PDGF receptor specifically. Using 3-D structures based on homology modeling would possibly allow the initial development of a binding model for this receptor and the compounds described here.

In summary this work resulted in the development of 2-D QSAR models which are able to provide a detailed interpretation of the structure-activity relationship for the PDGFR inhibitors studied as well as a predictive model which could conceivably be used as a screening tool for analogous compounds.

Table 7.1. The regression statistics for the best linear regression model.

Descriptor	β	Standard Error	t	P	VIF
Constant	0.50529	0.0499	10.129	1.59×10^{-14}	
MDEN-23	0.13957	0.0516	2.703	8.97×10^{-3}	1.23
RNHS-3	0.23205	0.0501	4.576	2.49×10^{-5}	1.26
SURR-5	-0.43415	0.0529	-8.19	2.56×10^{-11}	1.12

MDEN-23 - molecular distance edge vector between secondary and tertiary nitrogens;[21] RNHS-3 - relative hydrophilic surface area[23] defined as the product of the sum of the hydrophilic constants and surface area of the most hydrophilic atom divided by overall $\log P$; SURR-5 - the ratio of atomic constant weighted hydrophobic (low) surface area to the atomic constant weighted hydrophilic surface area[23,25]

Table 7.2. A summary of overall statistics for the best linear regression model.

	Number of Molecules	RMSE	R^2
Training set	65	0.38	0.65
Prediction set	13	0.47	0.38

Table 7.3. Ranges of the descriptors used in the best linear regression model.

Descriptor	Maximum	Minimum	Mean
MDEN-23	7.466	1.784	2.796
RNHS-3	-1.637	-37.726	-4.752
SURR-5	-1.633	-4.423	-3.180

Table 7.4. The average statistics for the training and prediction set predictions made by 500 randomized models.

	R^2		RMSE	
	Mean	Std. Deviation	Mean	Std. Deviation
Training Set	0.05	0.04	0.72	0.03
Prediction Set	0.08	0.11	1.04	0.12

Table 7.5. A summary of the statistics from the PLS analysis of the best 3-descriptor linear model.

Component	X Variance	Error SS	R^2	PRESS	Q^2
1	0.51	14.80	0.52	16.67	0.45
2	0.78	12.11	0.60	13.43	0.56
3	1.00	12.07	0.61	13.27	0.56

Table 7.6. The weights for the 3 validated components from the PLS analysis of the 3-descriptor linear model.

Descriptor	Component 1	Component 2	Component 3
MDEN-23	-0.16	0.93	0.30
RNHS-3	0.55	-0.17	0.81
SURR-5	-0.82	-0.29	0.48

Table 7.7. The statistics for the best nonlinear CNN model.

	Number of Molecules	RMSE	R^2
Training Set	57	0.22	0.94
Cross Validation Set	9	0.21	0.90
Prediction Set	13	0.32	0.61

Table 7.8. Summary of the statistics for the training, cross-validation and prediction sets from randomized runs using the best CNN model*.

	R^2		RMSE	
	Mean	Std. Deviation	Mean	Std. Deviation
Training Set	0.10	0.19	0.71	0.11
Cross-validation Set	0.10	0.23	0.96	0.14
Prediction Set	0.01	0.10	1.11	0.14

* The architecture used was 7–3–1

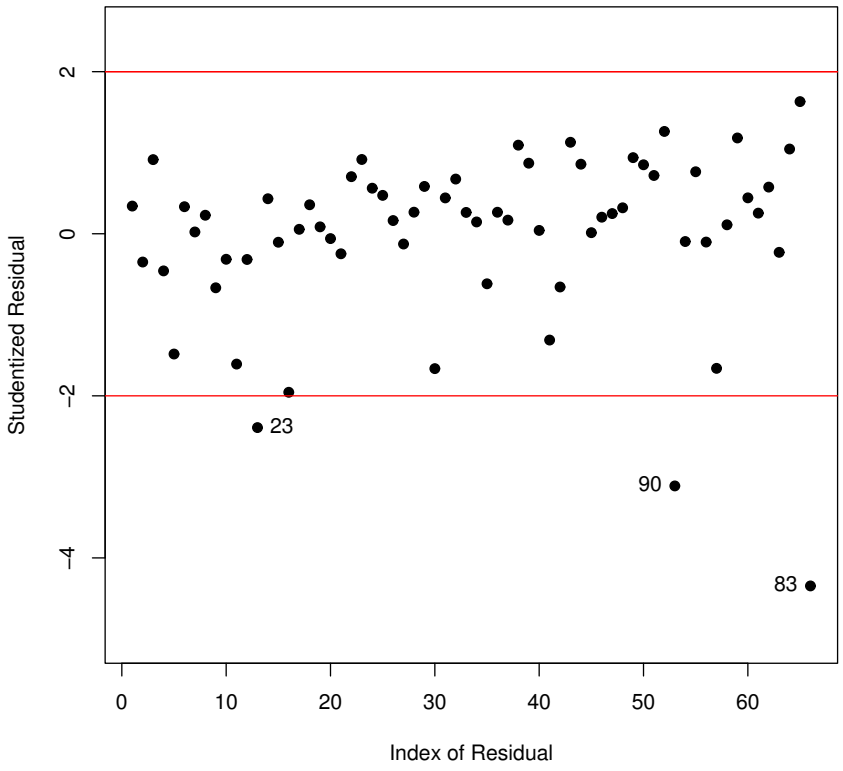


Fig. 7.1. A plot of the studentized residuals from the 3-descriptor linear model with outliers marked.

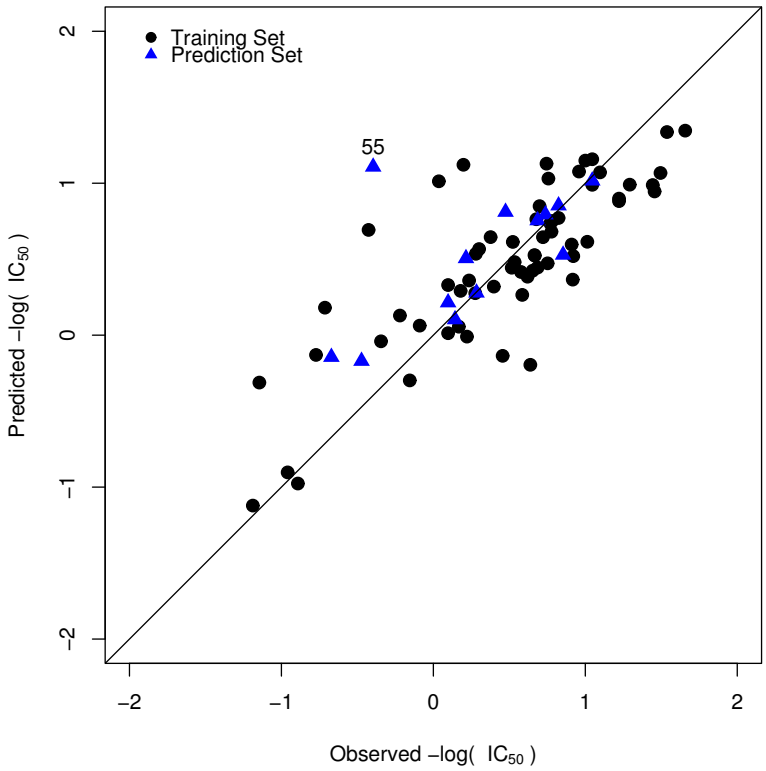


Fig. 7.2. A plot of observed versus predicted $-\log(\mathrm{IC}_{50})$ values from the best linear model after training set outliers were removed. The annotated point represents a prediction set outlier.

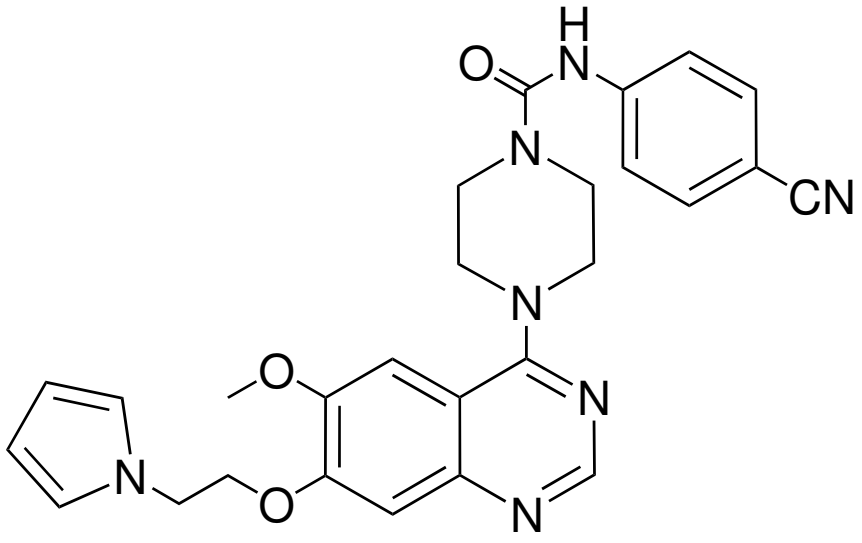


Fig. 7.3. The structure of the prediction set outlier (**55**) from the best linear and nonlinear CNN models.

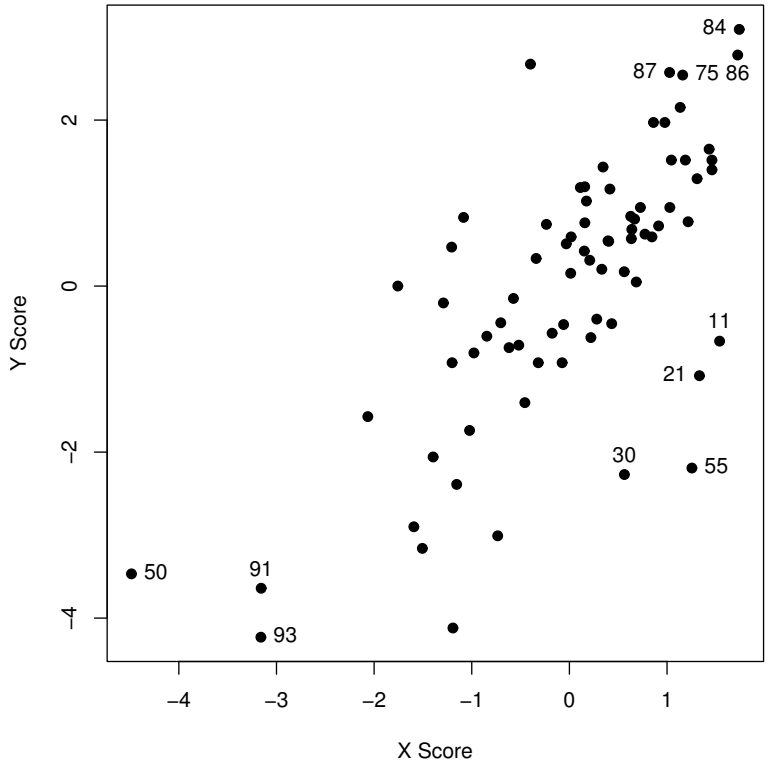


Fig. 7.4. The score plot for PLS component 1.

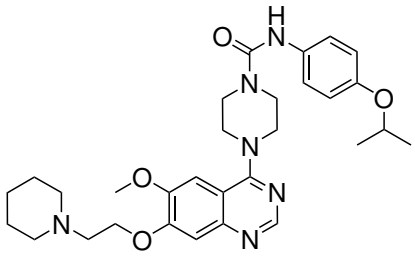
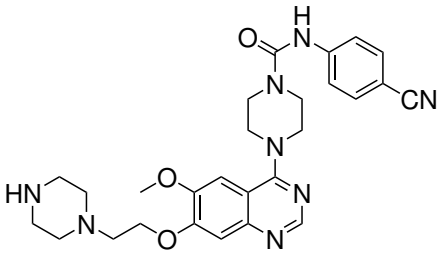
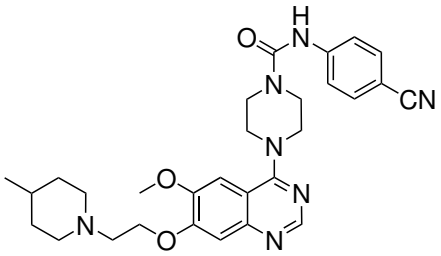
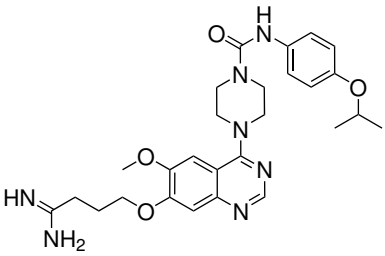
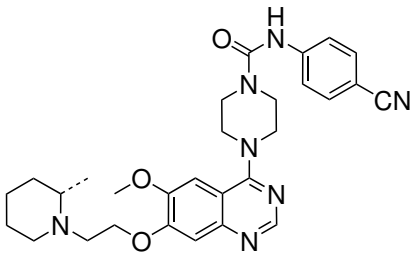
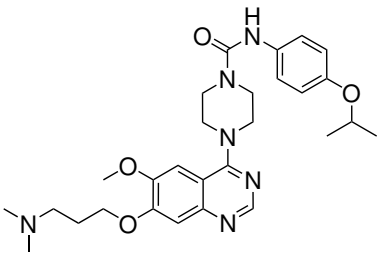
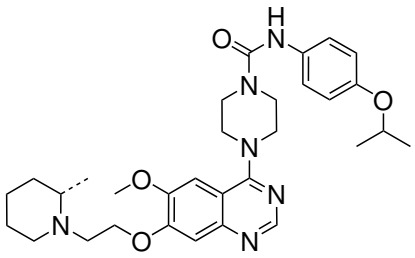
Active	Inactive
	
75 (1.44)	**50** (-0.89)
	
84 (1.65)	**91** (-0.95)
	
86 (1.53)	**93** (-1.18)
	
87 (1.45)	

Fig. 7.5. A comparison of the structures of the active and inactive compounds predicted by component 1 from the 3-component PLS model. Activity values in $-\log(\mathrm{IC}_{50})$ units are provided within brackets.

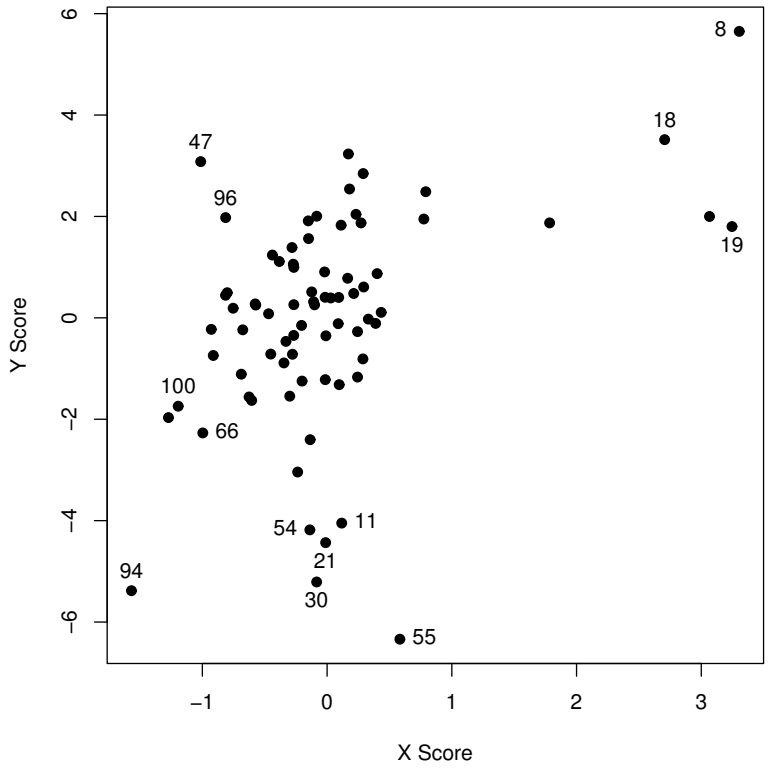


Fig. 7.6. The score plot for PLS component 2.

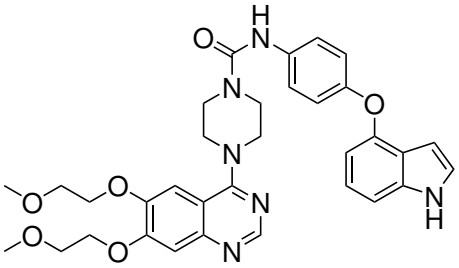
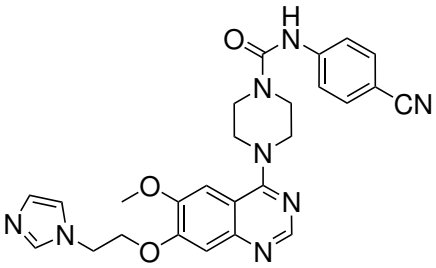
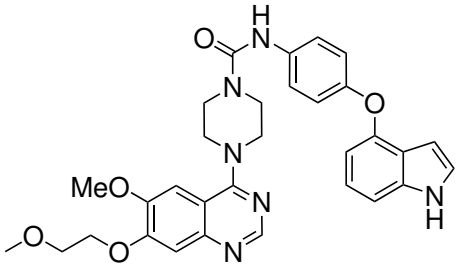
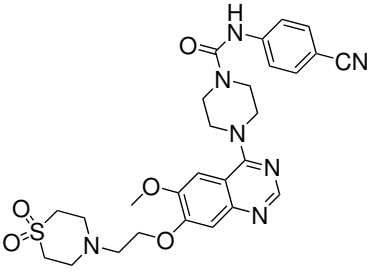
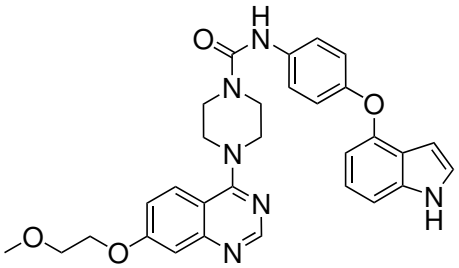
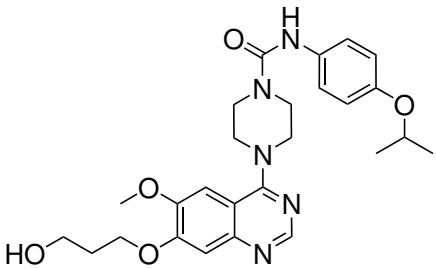
Active	**Inactive**
	
8 (1.49)	**54** (-0.71)
	
18 (0.77)	**66** (-0.47)
	
19 (0.77)	**94** (-1.14)

Fig. 7.7. A comparison of the structures of the active and inactive compounds predicted by component 2 from the 3 component PLS model. Activity values in $-\log(\mathrm{IC}_{50})$ units are provided within brackets.

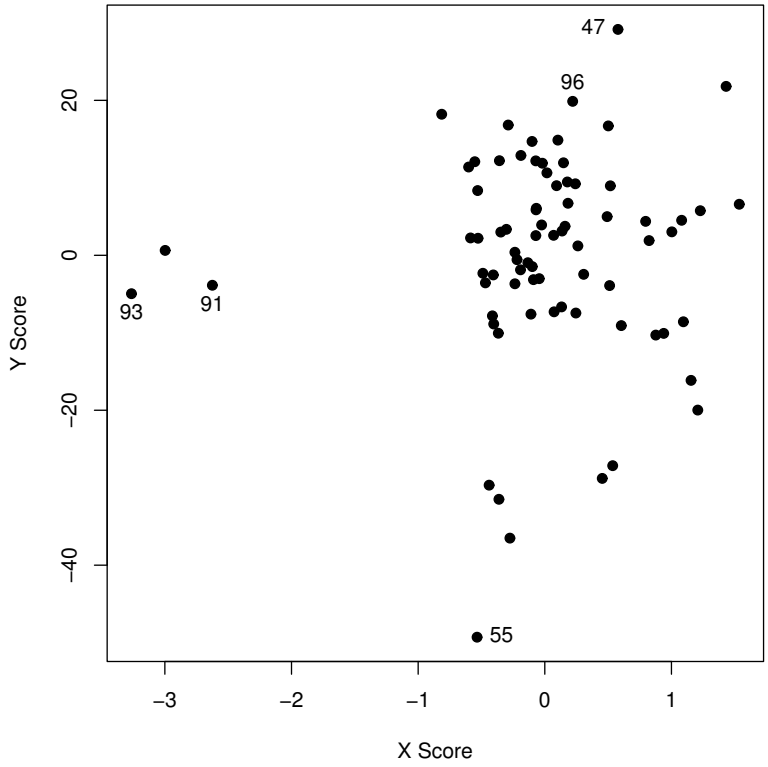


Fig. 7.8. The score plot for PLS component 3.

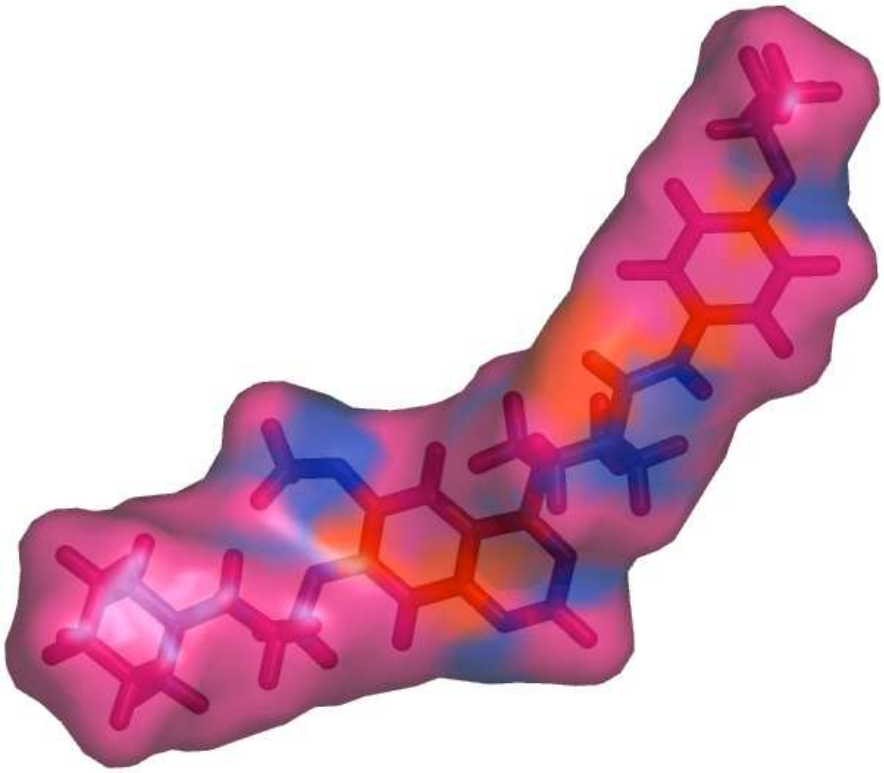


Fig. 7.9. Molecular surface plot of **75**, colored by hydrophobicity values (blue is most hydrophilic and red is most hydrophobic).

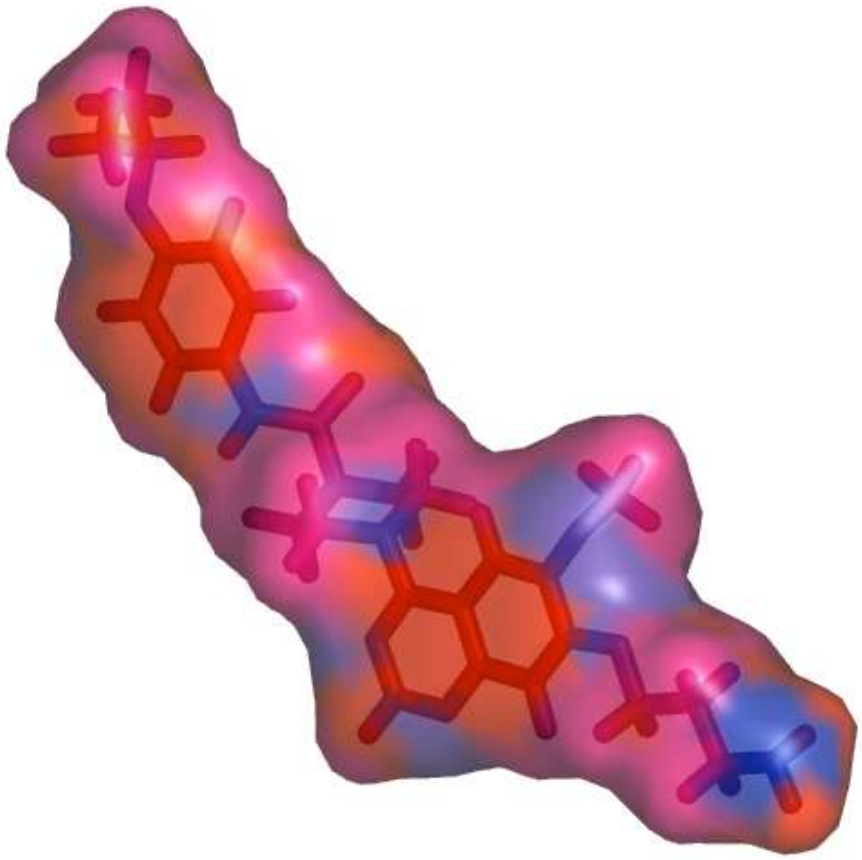


Fig. 7.10. Molecular surface plot of **93**, colored by hydrophobicity values (blue is most hydrophilic and red is most hydrophobic).

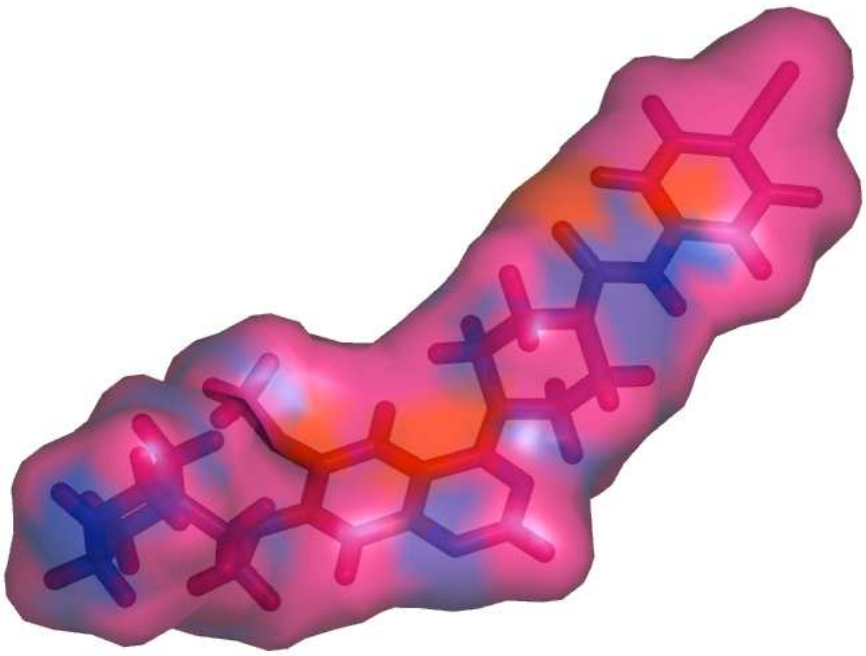


Fig. 7.11. Molecular surface plot of **50**, colored by hydrophobicity values (blue is most hydrophilic and red is most hydrophobic).

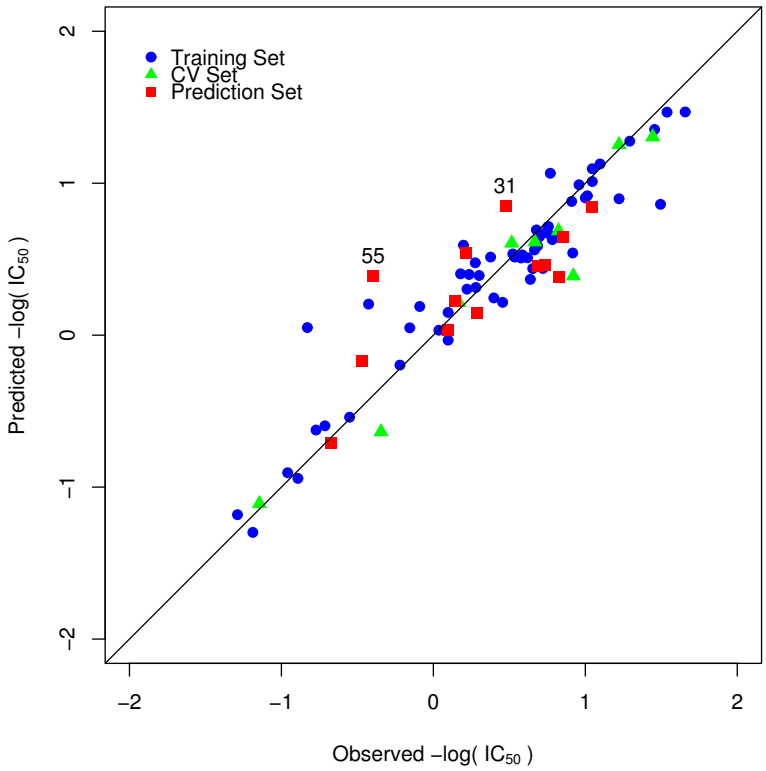
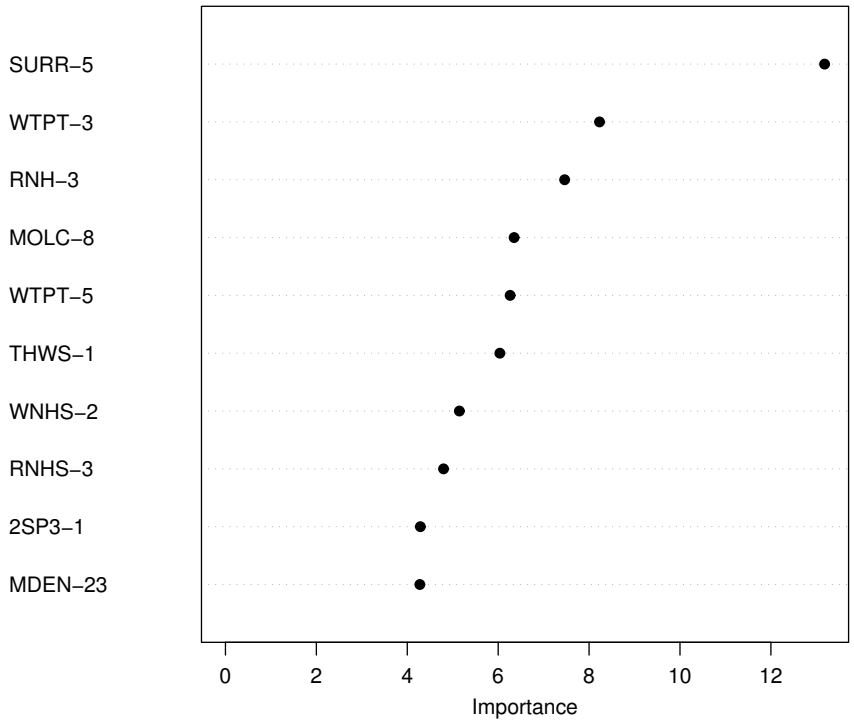


Fig. 7.12. A plot of the observed versus predicted $-\log(\mathrm{IC}_{50})$ values for the best nonlinear CNN model. The annotated points are possible prediction set outliers.

Fig. 7.13. A variable importance plot generated from the random forest model built using the reduced descriptor pool with no compounds excluded from the training or prediction set.*



* SURR-5 - the ratio of atomic constant weighted hydrophobic (low) surface area to the atomic constant weighted hydrophilic surface area;[23,25] WTPT-3 - sum of path lengths starting from heteroatoms;[35] RNH-3 - sum of hydrophilic constants divided by the value of log P;[25] MOLC-8 - path-cluster of length 4 molecular connectivity index;[41] WTPT-5 - sum of path lengths starting from nitrogen;[35] THWS-1 - total hydrophobic weighted surface area[25] defined as the sum of the product of atomic log P values and hydrophobic atom surface areas; WNHS-2 - surface weighted hydrophilic surface area[25] defined as the product of the hydrophilic surface area multiplied by the total molecular surface area divided by 1000; RNHS-3 - relative hydrophilic surface area[23] defined as the product of the sum of the hydrophilic constants and surface area of the most hydrophilic atom divided by overall log P; 2SP3-1 - the number of sp^3 carbons bound to two other carbons; MDEN-23 - molecular distance edge vector between secondary and tertiary nitrogens[21]

References

[1] Kurup, A.; Garg, R.; Hansch, C. Comparative QSAR Study of Tyrosine Kinase Inhibitors. *Chem. Rev.* **2001,** *101,* 2573–2600.

[2] Iida, H.; Seifert, R.; Alpers, C.; Gronwald, R.; Philips, P.; Pritzl, P.; Gordon, K.; Gown, A.; Ross, R.; Bowen-Pupe, D. Platelet Derived Growth Factor (PDGF) and PDGF Receptor (PDGFR) Are Induced in Mesangial Proliferative Nephritis in The Rat. *Proc. Natl. Acad. Sci.* **1995,** *88,* 6560–6564.

[3] Pandey, A.; Volkots, D. L.; Seroogy, J. M.; Rose, J. W.; Yu, J.-C.; Lambing, J. L.; Hutchaleelaha, A.; Hollenbach, S. J.; Abe, K.; Giese, N. A.; Scarborough, R. M. Identification of Orally Active, Potent, and Selective 4-Piperazinylquinazolines as Antagonists of the Platelet-Derived Growth Factor Receptor Tyrosine Kinase Family. *J. Med. Chem.* **2002,** *45,* 3772–3793.

[4] Schlessinger, J.; Ullrich, A. Growth Factor Signaling By Receptor Tyrosine Kinases. *Neuron* **1992,** *9,* 383–391.

[5] Palmer, B.; Kraker, A.; HArtl, B.; Panopoulos, A.; Panek, R.; Batley, B.; Lu, G.; Trumo-Kallmeyer, S.; Showalter, H.; Denny, W. Structure-Activity Relationships for 5-Substituted 1-Phenylbenzimidazoles as Selective Inhibitors of the Platelet-Derived Growth Factor Receptor. *J. Med. Chem.* **1999,** *42,* 2373–2382.

[6] Kubo, K.; Shimizu, T.; Ohyama, S.; Murooka, H.; Nishitoba, T.; Kato, S.; Kobayashi, Y.; Yagi, M.; Isoe, T.; Nakamura, K.; Osawa, T.; Izawa, T. A Novel Series of 4-Phenoxyquinoxazolines: Potent and Highly Selective Inhibitors of PDGF Receptor Autophophorylation.. *Bioorg. Med. Chem. Lett.* **1997,** *7,* 2935–2940.

[7] Boschelli, D. H. *et al.* Synthesis and Tyrosine Kinase Inhibitory Activity of A Series of 2-Amino-8h-Pyrido[2,3-D] Pyrimidines: Identification of Potent, Selective Platelet-Derived Growth Factor Receptor Tyrosine Kinase Inhibitors. *J. Med. Chem.* **1998,** *41,* 4365–4377.

[8] Klutchko, S. R. *et al.* 2-Substituted Aminopyrido[2,3-d]pyrimidin-7(8H)-ones. Structure-Activity Relationships Against Selected Tyrosine Kinases and in Vitro and in Vivo Anticancer Activity. *J. Med. Chem.* **1998,** *41,* 3276–3292.

[9] Kraker, A.; Hartl, B.; Amar, A.; Barvian, M.; Showalter, H.; Moore, C. Biochemical and Cellular Effects of c-Src Kinase-Selective Pyrido [2,3-d] Pyrimidine Tyrosine Kinase Inhibitors. *Biochem. Pharamcol.* **2000,** *60,* 885–898.

[10] Shen, Q.; Lu, Q.-Z.; Jiang, J.-H.; Shen, G.-L.; Yu, R.-Q. Quantitative Structure-Activity Relationships (QSAR): Studies of Inhibitors of Tyrosine Kinase. *Eur. J. Pharm. Sci.* **2003,** *20,* 63–71.

[11] Khadikar, P. V.; Shrivastava, A.; Agrawal, V. K.; Srivastava, S. Topological Designing of 4-Perazinylquinazolines as Anatagonists of PDGFR Tyrosine Kinase Family. *Bioorg. Med. Chem. Lett.* **2003,** *13,* 3009–3014.

[12] Lokker, N.; O'Hare, J.; Barsoumian, A.; Tomlinson, J.; Ramakrishnan, V.; Fretto, L.; Giese, N. Functional Importance of the Platelet Derived Growth Factor Receptor Extra-Cellular Immunoglobin Like Domains: Identification of PDGF Binding Site and Neutralizing Monoclonal Antibodies. *J. Biol. Chem.* **1997,** *272,* 33037–33044.

[13] Jurs, P.; Chou, J.; Yuan, M. Studies of Chemical Structure Biological Activity Relations Using Pattern Recognition. In *Computer Assisted Drug Design*; Olsen, E.; Christoffersen, R., Eds.; American Chemical Society: Washington D.C., 1979.

[14] Stuper, A.; Brugger, W.; Jurs, P. *Computer Assisted Studies of Chemical Structure and Biological Function;* Wiley: New York, 1979.

[15] Breiman, L. Random Forests. *Machine Learning* **2001,** *45,* 5–32.

[16] R Development Core Team, "R: A Language and Environment For Statistical Computing", R Foundation for Statistical Computing, Vienna, Austria, 2004 ISBN 3-900051-07-0.

[17] Hypercube Inc., "Hyperchem", 2001.

[18] Sutter, J.; Dixon, S.; Jurs, P. Automated Descriptor Selection For Quantitative Structure-Activity Relationships Using Generalized Simulated Annealing. *J. Chem. Inf. Comput. Sci.* **1995,** *35,* 77–84.

[19] Goldberg, D. *Genetic Algorithms in Search Optimization & Machine Learning;* Addison Wesley: Reading, MA, 2000.

[20] Wessel, M. *Computer Assisted Development of Quantitative Structure-Property Relationships and Design of Feature Selection Routines,* PhD thesis, Pennsylvania State University, 1997.

[21] Liu, S.; Cao, C.; Li, Z. Approach to Estimation and Prediction For Normal Boiling Point (NBP) of Alkanes Based on a Novel Molecular Distance Edge (MDE) Vector, λ. *J. Chem. Inf. Comput. Sci.* **1998,** *38,* 387–394.

[22] Guha, R.; Jurs, P. The Development of QSAR Models to Predict and Interpret the Biological Activity of Artemisinin Analogues. *J. Chem. Inf. Comput. Sci.* **2004,** *44,* 1440–1449.

[23] Stanton, D.; Mattioni, B. E.; Knittel, J.; Jurs, P. Development and Use of Hydrophobic Surface Area (HSA) Descriptors for Computer Assisted Quantitative Structure-Activity and Structure-Property Relationships. *J. Chem. Inf. Comput. Sci.* **2004,** *44,* 1010–1023.

[24] Wildman, S.; Crippen, G. Prediction of Physicochemical Parameters by Atomic Contributions. *J. Chem. Inf. Comput. Sci.* **1999,** *39,* 868–873.

[25] Mattioni, B. E. *The Development of Quantitative Structure-Activity Relationship Models for Physical Property and Biological Activity Prediction of Organic Compounds,* PhD thesis, Pennsylvania State University, 2003.

[26] Stanton, D. On The Physical Interpretation of QSAR Models. *J. Chem. Inf. Comput. Sci.* **2003,** *43,* 1423–1433.

[27] Minitab, "Minitab", 2003.

[28] DeLano, W. "The PyMOL Molecular Graphics System", 2002.

[29] Matsuno, K.; Ichimura, M.; Nakajima, T.; Tahara, K.; Fujiwara, S.; Kase, H.; Giese, N.; Pandey, A.; Scarborough, R. M.; Yu, J.-C.; Lokker, N.; Irie, J.; Tsukuda, E.; Oda, S.; Nomoto, Y. Potent and Selective Inhibitors of PDGFR Phosphorylation. I. Synthesis and Structure-Activity Relationship of A New Class of Quinazoline Derivatives. *J. Med. Chem.* **2002,** *45,* 3057–3066.

[30] Derksen, S.; Keselman, H. J. Backward, Forward and Stepwise Automated Subset Selection Algorithms: Frequency of Obtaining Authentic and Noise Variables. *British Journal of Mathematical and Statistical Psychology* **1992,** *45,* 265–282.

[31] Mantel, N. Why Stepdown Procedures in Variable Selection. *Technometrics* **1970,** *12,* 621–625.

[32] Kier, L.; Hall, L. Molecular Connectivity VII: Specific Treatment to Heteroatoms. *J. Pharm. Sci.* **1976,** *65,* 1806–1809.

[33] Kier, L.; Hall, L. *Molecular Connectivity in Structure Activity Analysis.;* John Wiley & Sons: Hertfordshire, England, 1986.

[34] Kier, L.; Hall, L.; Murray, W. Molecular Connectivity I: Relationship to Local Anesthesia.. *J. Pharm. Sci.* **1975,** *64,*.

[35] Randic, M. On Molecular Idenitification Numbers. *J. Chem. Inf. Comput. Sci.* **1984,** *24,* 164–175.

[36] Carhart, R.; Smith, D.; Venkataraghavan, R. Atom Pairs as Molecular Features in Structure-Activity Studies: Definition and Applications. *J. Chem. Inf. Comput. Sci.* **1985,** *25,* 64–73.

[37] Cramer III, R.; Patterson, D.; Bunce, J. Comparative Molecular Field Analysis (CoMFA). I. Effect of Shape on Binding of Steroids to Carrier Protiens. *J. Am. Chem. Soc.* **1988,** *110,* 5959–5967.

[38] Cramer III, R.; Patterson, D.; Bunce, J.; Frank, I. Crossvalidation, Bootstrapping and Partial Least Squares Compared with Multiple Regression in Conventional QSAR Studies. *Quant. Struct-Act. Relat. Pharmacol.,Chem. Biol.* **1988,** *7,* 18–25.

[39] McTigue, M.; Wickersham, J.; Pinko, C.; Showalter, R.; Parast, C.; Tempczyk-Russell, A.; Gehring, M.; Mroczkowski, B.; Kan, C.; Villafranca, J.; Appelt, K. Crystal Structure of The Kinase Domain of Human Vascular Endothelial Growth Factor Receptor 2: A Key Enzyme in Angiogenesis. *Structure* **1999,** *7,* 319–330.

[40] Mohammadi, M.; Froum, S.; Hamby, J.; Schroeder, M.; Panek, R.; Lu, G.; Eliseenkova, A.; Green, D.; Schlessinger, J.; Hubbard, S. Crystal Structure of an Angiogenesis Inhibitor Bound to the FGF Receptor Tyrosine Kinase Domain. *EMBO J.* **1998,** *17,* 5896–5904.

[41] Balaban, A. Higly Discriminating Distance Based Topological Index. *Chem. Phys. Lett.* **1982,** *89,* 399–404.

Chapter 8

Interpreting Computational Neural Network QSAR Models: A Measure of Descriptor Importance

8.1 Introduction

Computational neural networks (CNNs) are an important component of the QSAR practitioner's toolbox for a number of reasons. First, neural network models are generally more accurate than other classes of models. The higher predictive ability of CNNs arises from their flexibility and their ability to model nonlinear relationships. Second, a variety of neural networks are available depending on the nature of the problem being studied. One can consider classes of neural networks depending on whether they are trained in a supervised manner (e.g., back-propagation networks) or in an unsupervised manner (e.g., Kohonen map[1]).

Neural network models also have a number of shortcomings. First, neural network models can be over-trained. Thus, it is frequently the case that a neural network model will have memorized the idiosyncrasies of the training set, essentially fitting the noise. As a result, when faced with a test set of new observations, such a model's predictive ability will be very poor. One way to alleviate this problem is the use of cross-validation and use of root mean square errors (RMSE) for characterizing the preformance of the CNN with the training, cross-validation, and prediction set's as described in Chapter 2. A second drawback is the matter of interpretability. Neural networks are generally regarded as black boxes. That is, one provides the input values and obtains an output value, but generally no information is provided regarding how those output values were obtained or how the input values correlate to the output value. In the case of QSAR models, the lack of interpretability forces the use of CNN models as purely predictive tools rather than as an aid in the understanding of structure property trends. As a result, neural network models are very useful as a component of a screening process. But the possibility of using these models in computationally guided structure design is

This work was published as Guha, R.; Jurs, P.C., "Interpreting Computational Neural Network QSAR Models: A Measure of Descriptor Importance", *J. Chem. Inf. Model.*, **2005**, *45*, 800–806.

low. This is in contrast to linear models, which can be interpreted in a simple manner as has been shown in Chapters 6 and 7.

A number of reports in the machine learning literature describe attempts to extract meaning from neural network models.[2–7] Many of these techniques attempt to capture the functional representation being modeled by the neural network. In a number of cases, these methods require the use of specific neural network algorithms[8,9] and so generalization can be difficult.

Interpretation can be considered in two forms, broad and detailed. The aim of a broad interpretation is to characterize how important an input neuron (or descriptor) is to the predictive ability of the model. This type of interpretation allows us to rank input descriptors in order of importance. However, in the case of a QSAR neural network model, this approach does not allow us to understand exactly how a specific input descriptor affects the network output (for a QSAR model, predicted activity or property). The goal of a detailed interpretation is to extract the structure-property trends from a CNN model. A detailed interpretation should be able to indicate how an input descriptor for a given input example correlates to the predicted value for that input. This chapter focuses on a method to provide a broad interpretation of a neural network model. A technique to provide a detailed interpretation is described in the next chapter.

8.2 Methodology

In this study we restrict ourselves to the use of 3-layer, fully-connected, feed-forward computational neural networks. Furthermore, we consider only regression networks (as opposed to classification networks). The input layer represents the input descriptors, and the value of the output layer is the predicted property or activity. The neural network models considered in this work were built using the ADAPT[10,11] methodology which uses a genetic algorithm[12,13] to search a descriptor pool to find the best subset (of a specified size) of descriptors that results in a CNN model having a minimal cost function. This methodology has been discussed in detail in Chapters 2 and 3. The broad interpretation is essentially a sensitivity analysis of the neural network. This form of analysis has been described by So and Karplus.[14] However, the results of a sensitivity analysis have not been viewed as a measure of descriptor importance. It should also be pointed out that the idea behind sensitivity analysis is also the method by which a measure of descriptor importance is generated for random forest models.[15,16]

The algorithm we have developed to measure descriptor importance proceeds in a number of steps. To start, a neural network model is trained and validated. The RMSE for this model is denoted as the base RMSE. Next, the first input descriptor is randomly scrambled, and then the neural network model is used to predict the activity of the observations. Because the values of this descriptor have been scrambled, one would expect the correlation between descriptor values and activity values to be obscured. As a result, the RMSE for these new predictions should be larger than the base RMSE. The difference between this RMSE value and the base RMSE indicates the importance of the descriptor to the model's predictive ability. That is, if a descriptor plays a major role in the model's predictive ability, scrambling that descriptor will lead to a greater loss of predictive ability (as measured by the RMSE value) than for a descriptor that does not play such an important role in the model. This procedure is then repeated for all the descriptors present in the model. Finally, we can rank the descriptors in order of importance.

An alternative procedure was also investigated; it consisted of replacing individual descriptors with random vectors. The elements of the random vectors were chosen from a normal distribution with mean and variance equal to that of the original descriptor. The RMSE values of the model with each descriptor replaced in turn by its random counterpart was recorded as described above. We did not notice any significant differences in the final ordering of the descriptors compared to the random scrambling experiments. In all cases the most important descriptor was the same. In two of the datasets the only difference occurred in the ordering of two or three of the least significant descriptors.

8.3 Datasets and Models

To test this measure of descriptor importance we considered a number of datasets that have been studied in the literature. In two cases, linear regression models had been built and interpreted using the PLS scheme described by Stanton.[17] In all cases neural network models were also reported, but no form of interpretation was provided for these models, as they were developed primarily for their superior predictive ability.

We applied the descriptor importance methodology to these neural network models and compared the resultant rankings of the descriptors to the importance of the descriptors described by the PLS method for the linear models. It is important to note that the linear and CNN models do not necessarily contain the same descriptors (and indeed may have none in common). However, since both types of models should capture

similar structure-property relationships present in the data, it is reasonable to expect that the descriptors used in the models should have similar interpretations. Due to the broad nature of the descriptor importance methodology, we do not expect a one-to-one correlation of interpretations of the linear and nonlinear models. However, the correlation does allow us to see which descriptors in a CNN model are playing the major role and by comparison with the interpretations provided for the linear models allows us to confirm that this method is able to capture descriptor importance correctly.

We considered three datasets. The first dataset consisted of a 147-member subset of compounds whose measured activity was normal boiling point. The whole dataset contained 277 compounds, obtained from the Design Institute for Physical Property Data (DIPPR) Project 801 database, and it was studied by Goll and Jurs.[18] Although the original work reported linear as well as CNN regression models we generated new linear and nonlinear models using the ADAPT methodology[19,20] and provide a brief interpretation for the linear case, using the PLS technique,[17] focusing on which descriptors are deemed important. The second dataset consisted of 179 artemisinin analogs. These were studied using CoMFA[21] by Avery et al.[22] The measured activity was the logarithm of the relative activity of the analogs (compared to the activity of artemisinin). This dataset has been discussed in Chapter 6 where it was used to build linear and nonlinear models using a 2-D QSAR methodology. An interpretation of the linear model using the PLS technique was also presented. The third dataset consisted of a set of 79 PDGFR phosphorylation inhibitors described by Pandey et al.[23] The reported activity was $\log(IC_{50})$ and was obtained from a phosphorylation assay. A set of linear and nonlinear models were developed using this dataset and are described in Chapter 7, where we also provide an interpretation of the linear model.

8.4 Results

8.4.1 DIPPR Dataset

We first consider the linear and CNN models for the DIPPR dataset for modeling boiling points. The statistics of the linear regression model for this dataset are summarized in Table 8.1 and the meanings of the descriptors used in the model are summarized in Table 8.2. The R^2 value was 0.98, and the F-statistic was 1001 (for 7 and 139 degrees of freedom) which is much greater than the critical value of 2.076 ($\alpha = 0.05$). The model is thus statistically valid. The corresponding PLS analysis is summarized in Table 8.3.

The PLS statistics indicate that the increase in Q^2 beyond the fourth component is negligible. Thus, we need only consider the most important descriptors in the first three components. To see which descriptor is contributing the most in a given component, we consider the X weights obtained from the PLS analysis which are displayed in Table 8.4. In component 1 it is clear that MW and V4P-5 are the most heavily weighted descriptors. Higher values of molecular weight correspond to larger molecules and thus elevated boiling points. The V4P-5 descriptor characterizes branching in the molecular structure, and higher values indicate a higher degree of branching. Thus, both of the most important descriptors in the first component correlate molecular size to higher values of boiling point. In the second component we see that the most weighted descriptors are RSHM and PNSA-3. RSHM characterizes the fraction of the solvent accessible surface area associated with hydrogens that can be donated in a hydrogen-bonding intermolecular interaction. PNSA-3 is the charge weighted partial negative surface area. Clearly, both these descriptors characterize the ability of molecules to form hydrogen bonds. In summary, the structure-property relationship captured by the linear model indicates that London forces dominate the relationship. Although individual atomic contributions to the trend are small, larger molecules will have more interactions leading to higher boiling points. In addition, attractive forces, originating from hydrogen bond formation, also play a role in the relationship and these are characterized in the second component of the PLS model. We can use the above discussion and information from the PLS analysis to rank the descriptors considered in the PLS analysis in decreasing order of contributions: MW, V4P-5, RSHM, PNSA-3.

The next step was to develop a computational neural network model for this dataset. The ADAPT methodology was used to search for descriptor subsets ranging in size from 4 to 6. The final CNN model had a 5–3–1 architecture, and the statistics of the model are reported in Table 8.5. The descriptors in this model were FNSA-3, MOLC-6, WPHS-3 and RPHS, which are described in Table 8.2. The increase in RMSE values for the descriptors in each neural network model are reported in Tables 8.6, 8.7 and 8.8. In each table the third column represents the increase in RMSE, due to the scrambling of the corresponding descriptor, over the base RMSE. It is evident that scrambling some descriptors leads to larger increases, whereas others lead to negligible increases in the RMSE. The information contained in these tables is more easily seen in the descriptor importance plots shown in Figs. 8.1, 8.2 and 8.3. These figures plot the increase in RMSE for each descriptor, in decreasing order.

Considering the DIPPR dataset (Table 8.6 and Fig. 8.1) we see that although the linear and nonlinear models have only one descriptor in common (RSHM) the types of descriptors are the same in the both models (topological and charge-related). The CNN model contains charged partial surface area descriptors (RSHM and FNSA-3) as well as hydrophobicity descriptors (WPHS-3 and RPHS). One may expect that the structure-activity trends captured by the CNN model are similar to those captured by the linear model. If we look at Fig. 8.1 we see that the most important descriptor is WPHS-3. This descriptor represents the surface weighted hydrophobic surface area. Since this is correlated to the positively charged surface area, this descriptor should characterize hydrogen-bonding ability. The second most important descriptor is MOLC-6, which represents a topological path containing four carbon atoms. This descriptor essentially characterizes molecular size. The next two most important descriptors are RSHM, which has been previously described, and RPHS, which characterizes the relative hydrophobic surface area. The insignificant separation between these two descriptors along the X-axis indicates that these two descriptors are probably playing similar roles in the predictive ability of the CNN model. When compared to the ranking of descriptor contributions in the PLS analysis we see that the CNN descriptor importance places a hydrophobicity descriptor as the most important descriptor, followed by MOLC-6 which, as mentioned, characterizes molecular size. The difference in ordering may be due to the fact that the CNN is able to find a better correlation between the selected descriptors, such that WPHS-3 provides the maximum amount of information. However, in general, the broad interpretation provided by this method does compare well with that of the linear model using PLS.

8.4.2 PDFGR Dataset

We now consider the PDGFR dataset. Chapter 7 described a 3-descriptor linear regression model. The PLS interpretation of this model indicated that all three descriptors were important. These descriptors were SURR-5, RNHS-3 and MDEN-23. The first two descriptors are hydrophobic surface area descriptors and the last descriptor is a topological descriptor which represents the geometric mean of the topological path length between secondary and tertiary nitrogens. If we take into account the PLS components in which these descriptors occur, they can be ordered as SURR-5, MDEN-23 and RNHS-3 (decreasing importance). The CNN model for this dataset, described in Chapter 7, contained 7 descriptors: N5CH, WTPT-3, WTPT4, FLEX-4, RNHS-3, SURR-5

and APAVG. A summary of these descriptors can be found in Table 8.2 As can be seen, the linear regression and CNN models have two descriptors in common: SURR-5 and RNHS-3.

The descriptor importance plot for the PDFGR dataset (Fig. 8.2) shows the most important descriptor to be SURR-5. The next most important descriptor is RNHS-3. The position of RNHS-3 on the plot indicates that it plays a much less important role than SURR-5 in the model's predictive ability. However, it is notable that the two most important descriptors identified by our method in the CNN model are also the two most important descriptors identified by the PLS analysis of the linear regression model. Given that the CNN model and linear model should capture similar structure property trends in the dataset, the similarity between the important descriptors in the two models serve to confirm the validity of this method. It is also interesting to note that FLEX-4, N5CH-12 and WTPT-3 descriptors are relatively closely spaced along the X-axis. Though these descriptors are ranked, the relatively small increase between each one might be indicative that these descriptors are performing similar roles within the network. The position of WTPT-5 in the plot indicates that it contributes relatively little to the model's predictive ability.

8.4.3 Artemisinin Dataset

Finally, we consider the artemisinin dataset. Chapter 6 presented a 4-descriptor linear regression model. The most important descriptors identified by a PLS analysis of this model were N7CH, WTPT-2 and NSB. N7CH and WTPT-2 are topological descriptors. N7CH is the number of seventh order chains and WTPT-2 is the weighted path descriptor divided by the total number of atoms. NSB is the count of single bonds. Summaries of the descriptors are presented in Table 8.2 and further details of these descriptors can be found in the original work.[24] Taking into account the PLS components in which these appear, we can order them as: N7CH, NSB, WTPT-2 (decreasing order of importance). The CNN model described in Chapter 6 had 10 descriptors: KAPPA-6, NDB, MOMI-4, N7CH, MOLC-8, WTPT-5, MDE-12, MDE-13, ELEC and FPSA-3. Brief summaries of the descriptors can be found in Table 8.2. In this case, the linear model and the CNN model have only one descriptor in common (N7CH), though both models contain weighted path descriptors (WTPT-2 for the linear model and WTPT-5 for the CNN model). The CNN model also contains a number of topological and geometric descriptors (KAPPA-6, MOLC-8, MDE-12, MDE-13 and MOMI-4) as well as

two electronic descriptors (ELEC and FPSA-3). When compared to the linear model, which contained only topological descriptors, it appears that the CNN model has been able to capture a more detailed view of the structure-property relationship by including information from electronic descriptors.

If we now consider the descriptor importance plot in Fig. 8.3 we see that the most important descriptor is N7CH. Its contribution to the model's predictive ability is clearly significant. In contrast, the next most significant descriptor, MDE-13, is much further left than N7CH. MDE13 is a distance edge descriptor (the number of paths between primary and tertiary carbons) and characterizes molecular size. In this sense, it is similar in nature to the count of single bonds (the second most important descriptor in the linear model). Once again we see that a number of descriptors are relatively closely spaced along the X- axis (such as NDB, MOLC-8, FPSA-3 and KAPA-6). It is interesting to note that the electronic descriptors are approximately in the middle of the ranking, whereas the top four descriptors are mainly topological in nature. One can conclude that electronic factors do not play a major role in the structure property relationship captured by this CNN model, but do enhance the predictive ability of the model when compared to the performance of the linear model. It is also interesting to see that the WTPT-5 is ranked quite low in the CNN model and is the least important descriptor in the linear model. As before, if we assume that the linear and nonlinear models capture similar structure-property relationships we see that the CNN descriptor importance method ranks the descriptors such that their order of importance is similar to that in the linear model.

8.5 Conclusions

From the preceding results, we see that the proposed measure of importance is able to characterize the relative importance of descriptors. The resultant ranking of descriptors corresponds well to a ranking of descriptors obtained by PLS analysis of a linear model using the same dataset. The comparison with the linear models is not necessarily one-to- one since the descriptors in the linear and CNN models will generally be different.

The representation of descriptor importance plots allows easy visual analysis of the descriptors in the model. In addition, apart from merely ranking, the plots provide a qualitative view of how important a given descriptor is relative to others. That is, by looking at the separation between two descriptors on the X-axis, one may determine

that a certain descriptor plays a much more significant role than another in the model's predictive power. We conjecture that descriptors with little separation along the X-axis play similar roles within the CNN architecture. However, confirmation of this requires a more detailed interpretation of the CNN model, which we present in the next chapter.

It is clear that the proposed measure of descriptor importance does not allow the user to elucidate exactly how the model represents the information captured by a given descriptor. That is, the methodology does not indicate the sign (or direction) of the effect of each input descriptor. Thus we cannot draw conclusions regarding the nature of the correlation between input descriptors and network output. However, even in the absence of a detailed understanding of the behavior of the input descriptors, the method described here allows the user to determine the most important descriptor (or descriptors) and investigate whether replacement by similar descriptors might lead to an improvement in the model's predictive ability. We investigated this possibility by replacing the SURR-5 descriptor from the 10–5–1 CNN model for the artemisinin dataset by other hydrophobic surface area descriptors. In some cases the RMSE of the resultant model did increase, though not significantly. However in a few cases the RMSE of the model decreased, compared to the original RMSE. This is not surprising, as the importance plots show that theoretically related descriptors (such as WTPT-3 and WTPT-5 in Fig. 8.2) may have significantly different contributions. However, the importance measure does allow us to change model descriptors in a guided manner. This procedure is functionally similar to feature selection but the main difference is that it is applied to a subset of descriptors that have already been deemed "good" by a feature selection algorithm (genetic algorithm[12] or simulated annealing[25]). Hence, the result of the "tweaking" procedure described here, is akin to locally fine-tuning a given descriptor subset, rather than selecting whole new subsets.

Finally, we note that the machine learning literature describes a number of approaches to interpretation of CNN models. In general, these methods are closely tied to specific types of neural network models.[8,9] A useful feature of this interpretation methodology is that it is quite general in nature. That is, the methodology is not dependent on the specific characteristics of a neural network and depends only on the input data. This implies that the methodology can be applied to obtain descriptor importance measures for any type of neural network model such as single layer or multilayer perceptrons[26] or radial basis networks.[27] Furthermore many interpretation methods are focused on extracting rules or analytical forms of the CNN's internal model.[28–30] In many cases, this necessitates a complex analysis of the model utilizing another pattern recognition[31]

or optimization algorithm.[28,32] In contrast, the method described here is simple to carry out and provides easily understandable conclusions. In summary, the interpretation methodology described here provides a broad view of descriptor importance for a neural network model, thus alleviating the black box nature of the neural network methodology to some extent.

Table 8.1. Summary of the linear regression model developed for the DIPPR dataset.

	Estimate	Std. Error	t	P
(Intercept)	−215.092	29.451	−7.30	0.000
PNSA-3	−3.561	0.210	−16.90	0.000
RSHM	608.071	21.302	28.55	0.000
V4P-5	19.576	3.308	5.92	0.000
S4PC-12	12.089	1.572	7.69	0.000
MW	0.579	0.061	9.42	0.000
WTPT-2	236.108	16.574	14.25	0.000
DPHS	0.198	0.028	7.07	0.000

Table 8.2: Glossary of descriptors reported in this paper

Descriptor Code	Description	Reference
APAVG	The mean value of all unique atom pairs present in the molecule	33
DPHS	The difference between the hydrophobic and hydrophilic surface area	34
ELEC	Electronegativity ($0.5 \times (\text{HOMO} + \text{LUMO})$)	
FLEX-4	Molecular mass of rotatable atoms divided by the hydrogen suppressed molecular weight	35
FNSA-3	Charge weighted partial negative relative surface area	36
FPSA-3	Atom weighted partial positive surface area divided by the total molecular surface area	36
KAPA-6	Atom corrected shape index (${}^3\kappa$)	37–39
MDE-12	Molecular distance edge vector between primary and secondary carbons	40
MDE-13	Molecular distance edge vector between primary and tertiary carbons	40

Table 8.2: (continued)

Descriptor Code	Description	Reference
MDEN-23	Molecular distance edge vector between secondary and tertiary nitrogen's	40
MOLC-6	Path of length 4 molecular connectivity index	41, 42
MOLC-8	Path-cluster of length 4 molecular connectivity index	41, 42
MOMI-4	Ratio of the principal component of the moment of inertia along the X-axis to that along the Y-axis	43
MW	Molecular weight	
NDB	Number of double bonds	
NSB	Number of single bonds	
N5CH-12	Number of 5^{th} order chains	44–46
N7CH	Number of 7^{th} order chains	44–46
PNSA-3	Charge weighted partial negative surface area	36
RNHS-3	A hydrophobic surface area descriptor defined as the product of the surface area of the most hydrophilic atom and the sum of the hydrophilic constants divided by the logP value for the molecule	34, 47
RPHS	Product of the surface area of the most hydrophobic atom and its hydrophobic constant divided by the sum of all hydrophobic constants	34, 47
RSHM	Fraction of the solvent accessible surface area associated with hydrogens that can be donated in a hydrogen-bonding intermolecular interaction	36
S4PC-12	4th order simple path cluster	44–46
SURR-5	Ratio of the atomic constant weighted hydrophobic (low) surface area and the atomic constant weighted hydrophilic surface area	34, 47
V4P-5	4^{th} order valence path molecular connectivity index	44–46
WPHS-3	Surface weighted hydrophobic surface area	34, 47
WTPT-3	Sum of all path lengths starting from heteroatoms	48
WTPT-5	Sum of all path lengths starting from nitrogen's	48

Table 8.3. Summary of the PLS analysis based on the linear regression model developed for the DIPPR dataset.

Components	X Variance	Error SS	R^2	PRESS	Q^2
1	0.431	94868.5	0.863	99647.6	0.857
2	0.660	26221.6	0.962	29046.7	0.958
3	0.768	16614.8	0.976	19303.3	0.972
4	0.843	14670.8	0.978	17027.6	0.975
5	0.911	14032.5	0.979	16281.3	0.976
6	0.987	13775.9	0.980	15870.6	0.977
7	1.000	13570.9	0.980	15653.0	0.977

Table 8.4. The X weights for the PLS components from the PLS analysis summarized in Table 8.3.

	Component						
Descriptor	1	2	3	4	5	6	7
PNSA-3	-0.303	-0.423	0.202	-0.250	0.254	-0.737	-0.127
RSHM	0.190	0.779	0.347	-0.032	0.222	-0.377	0.209
V4P-5	0.485	-0.157	-0.071	-0.664	-0.368	-0.096	0.384
S4PC-12	0.289	-0.079	-0.578	0.531	-0.031	-0.469	0.264
MW	0.499	-0.085	0.368	0.242	-0.397	-0.170	-0.602
WTPT-2	0.483	-0.051	-0.265	-0.221	0.708	0.138	-0.350
DPHS	0.263	-0.415	0.540	0.322	0.297	0.187	0.487

Table 8.5. Summary statistics for the best CNN model for the DIPPR dataset. The model architecture was 5–3–1.

	R^2	RMSE
TSET	0.98	9.92
CVSET	0.99	7.89
PSET	0.98	8.61

Table 8.6. Increase in RMSE due to scrambling of individual descriptors. The CNN architecture was 5–3–1 and as built using the DIPPR dataset. The base RMSE was 9.92

	Scrambled Descriptor	RMSE	Difference
1	FNSA-3	30.50	20.58
2	RSHM	35.76	25.84
3	MOLC-6	51.32	41.39
4	WPHS-3	66.27	56.35
5	RPHS	35.75	25.83

Table 8.7. Increase in RMSE due to scrambling of individual descriptors. The CNN architecture was 7–3–1 and was built using the PDGFR dataset. The base RMSE was 0.29

	Scrambled Descriptor	RMSE	Difference
1	N5CH-12	0.50	0.20
2	WTPT-3	0.49	0.19
3	WTPT-5	0.39	0.09
4	FLEX-4	0.51	0.21
5	RNHS-3	0.51	0.21
6	SURR-5	0.72	0.42
7	APAVG	0.48	0.18

Table 8.8. Increase in RMSE due to scrambling of individual descriptors. The CNN architecture was 10–5–1 and was built using the artemisinin dataset. The RMSE for the original model was 0.48

	Scrambled Descriptor	RMSE	Difference
1	KAPA-6	0.88	0.40
2	N7CH-20	1.97	1.49
3	MOLC-8	0.98	0.50
4	NDB	0.99	0.51
5	WTPT-5	0.78	0.30
6	MDE-12	0.85	0.37
7	MDE-13	1.18	0.70
8	MOMI-4	0.77	0.29
9	ELEC	0.93	0.45
10	FPSA-3	0.89	0.41

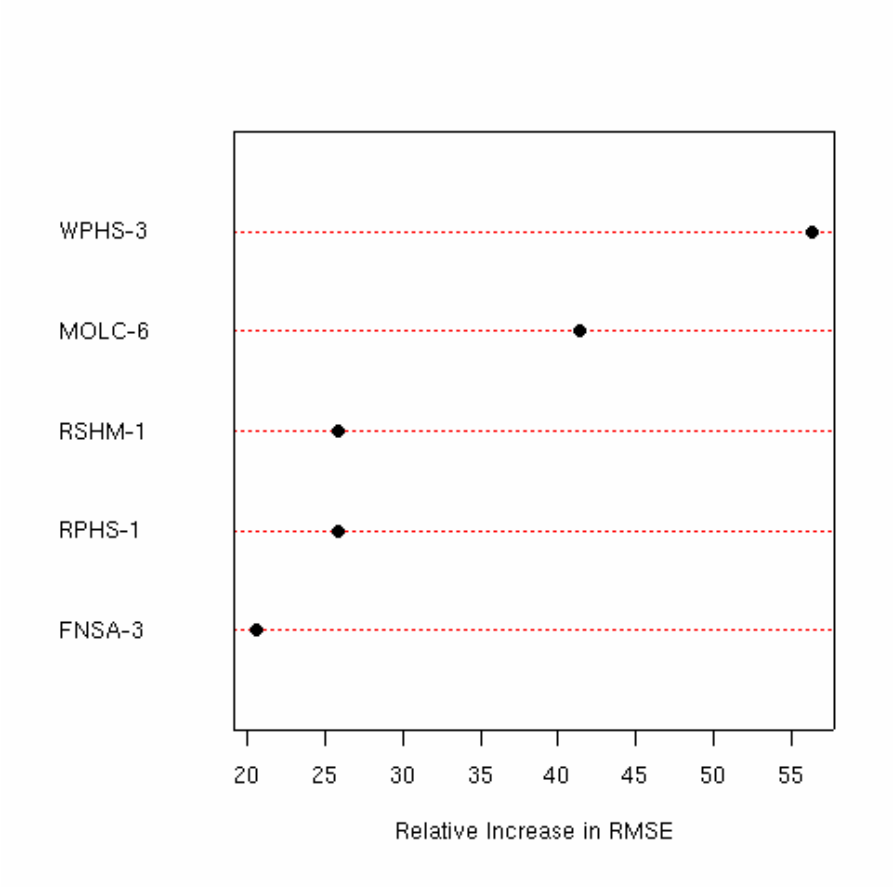


Fig. 8.1. Importance Plot for the 5–3–1 CNN model built using the DIPPR dataset

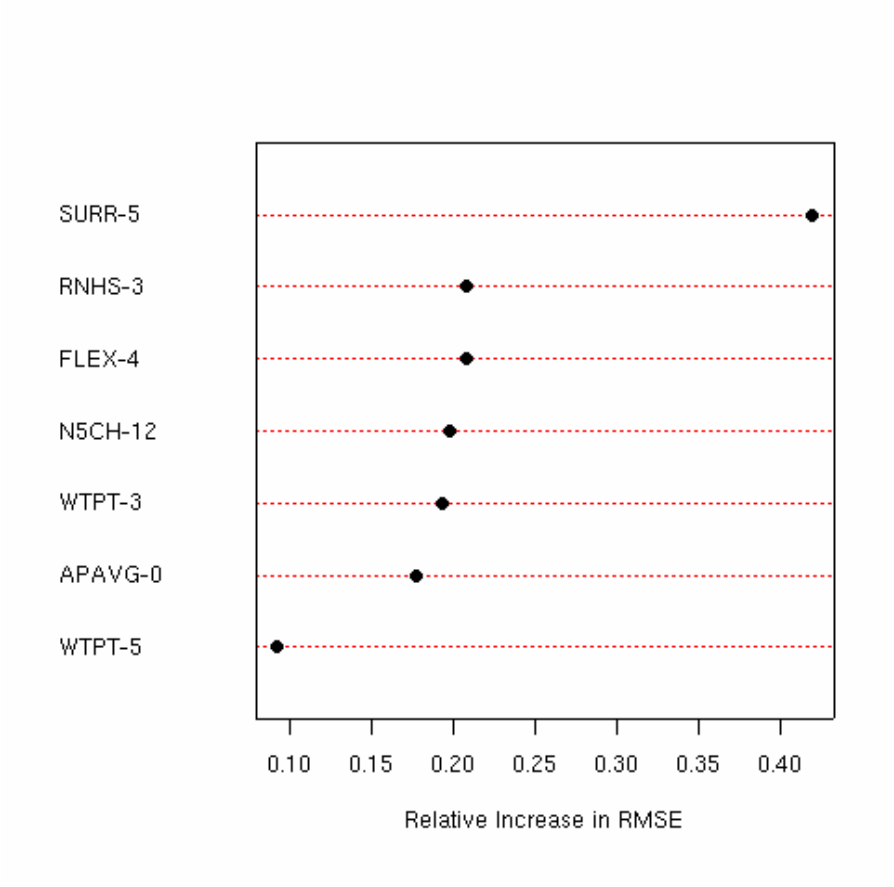


Fig. 8.2. Importance Plot for the 7–3–1 CNN model built using the PDGFR dataset

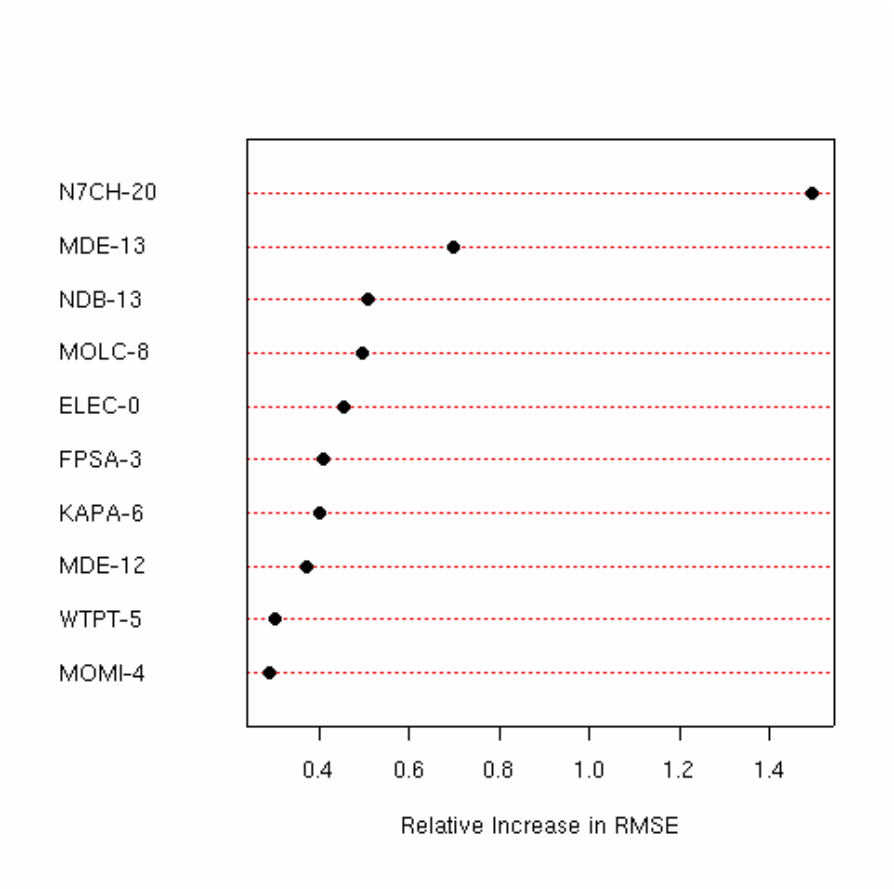


Fig. 8.3. Importance Plot for the 10–5–1 CNN model built using the artemisinin dataset

References

[1] Kohonen, T. *Self Organizing Maps;* Springer: Berlin, 1994.

[2] Castro, J.; Mantas, C.; Benitez, J. Interpretation of Artificial Neural Networks by Means of Fuzzy Rules. *IEEE Trans. Neural Networks* **2002,** *13,* 101–116.

[3] Jones, W.; Vachha, R.; Kulshrestha, A. DENDRITE: A System for Visual Interpretation of Neural Network Data. In *Southeastcon, Proceedings of*, Vol. 2; IEEE: New York, NY, 1992.

[4] Limin, F. Rule Generation from Neural Networks. *IEEE Trans. Systems, Man and Cybernetics* **1994,** *24,* 1114–1124.

[5] Ney, H. On the Probabilistic Interpretation of Neural Network Classifiers and Discriminative Training Criteria. *IEEE Trans. Pat. Anal. Mach. Intell.* **1995,** *17,* 107–119.

[6] Taha, I.; Ghosh, J. Symbolic Interpretation of Artificial Neural Networks. *IEEE Trans. Knowl. Data Eng.* **1999,** *11,* 448–463.

[7] Takahashi, T. An Information Theoretical Interpretation of Neuronal Activities. In *Neural Networks, International Joint Conference on*, Vol. 2; IEEE: New York, NY, 1991.

[8] Bologna, G. Rule Extraction from Linear Combinations of DIMLP Neural Networks. In *Proceedings of the Sixth Brazilian Symposium on Neural Networks*; IEEE: New York, NY, 2000.

[9] Hervas, C.; Silva, M.; Serrano, J. M.; Orejuela, E. Heuristic Extraction of Rules in Pruned Artificial Neural Network Models Used for Quantifying Highly Overlapping Chromatographic Peaks. *J. Chem. Inf. Comput. Sci.* **2004,** *44,* 1576–1584.

[10] Jurs, P.; Chou, J.; Yuan, M. Studies of Chemical Structure Biological Activity Relations Using Pattern Recognition. In *Computer assisted drug design*; Olsen, E.; Christoffersen, R., Eds.; American Chemical Society: Washington D.C., 1979.

[11] Stuper, A.; Brugger, W.; Jurs, P. *Computer Assisted Studies of Chemical Structure and Biological Function;* Wiley: New York, 1979.

[12] Goldberg, D. *Genetic Algorithms in Search Optimization & Machine Learning;* Addison Wesley: Reading, MA, 2000.

[13] Wessel, M. *Computer Assisted Development of Quantitative Structure-Property Relationships And Design of Feature Selection Routines,* PhD thesis, Pennsylvania State University, 1997.

[14] So, S.-S.; Karplus, M. Evolutionary Optimization in Quantitative Structure-Activity Relationship: An Application of Genetic Neural Networks. *J. Med. Chem.* **1996,** *39,* 1521–1530.

[15] Breiman, L. Random forests. *Machine Learning* **2001,** *45,* 5–32.

[16] Breiman, L.; Friedman, J.; Olshen, R.; Stone, C. *Classification and Regression Trees;* CRC Press: Boca Raton, FL, 1984.

[17] Stanton, D. On the Physical Interpretation of QSAR Models. *J. Chem. Inf. Comput. Sci.* **2003,** *43,* 1423–1433.

[18] Goll, E.; Jurs, P. Prediction of the Normal Boiling Points of Organic Compounds from Molecular Structures with a Computational Neural Network Model. *J. Chem. Inf. Comput. Sci.* **1999,** *39,* 974–983.

[19] Lu, X.; Ball, J.; Dixon, S.; Jurs, P. Quantitative Structure-Activity Relationships for Toxicity of Phenols Using Regression Analysis and Computational Neural Networks. *Environ. Toxicol. Chem.* **1994,** *13,* 841–851.

[20] Wessel, M.; Jurs, P. Prediction of Reduced Ion Mobility Constants from Structural Information Using Multiple Linear Regression Analysis and Computational Neural Networks. *Anal. Chem.* **1994,** *66,* 2480–2487.

[21] Cramer, R.; Patterson, D.; Bunce, J. Comparative Molecular Field Analysis (CoMFA). I. Effect of Shape on Binding of Steroids to Carrier Protiens. *J. Am. Chem. Soc.* **1988,** *110,* 5959–5967.

[22] Avery, M. A.; Alvim-Gaston, M.; Rodrigues, C. R.; Barreiro, E. J.; Cohen, F. E.; Sabnis, Y. A.; Woolfrey, J. R. Structure Activity Relationships of the Antimalarial Agent Artemisinin. the Development of Predictive in Vitro Potency Models Using CoMFA and HQSAR Methodologies. *J. Med. Chem.* **2002,** *45,* 292–303.

[23] Pandey, A.; Volkots, D. L.; Seroogy, J. M.; Rose, J. W.; Yu, J.-C.; Lambing, J. L.; Hutchaleelaha, A.; Hollenbach, S. J.; Abe, K.; Giese, N. A.; Scarborough, R. M. Identification of Orally Active, Potent, and Selective 4-Piperazinylquinazolines as Antagonists of the Platelet-Derived Growth Factor Receptor Tyrosine Kinase Family. *J. Med. Chem.* **2002,** *45,* 3772–3793.

[24] Guha, R.; Jurs, P. C. The Development of QSAR Models to Predict and Interpret the Biological Activity of Artemisinin Analogues. *J. Chem. Inf. Comput. Sci.* **2004,** *44,* 1440–1449.

[25] Sutter, J.; Dixon, S.; Jurs, P. Automated Descriptor Selection for Quantitative Structure-Activity Relationships Using Generalized Simulated Annealing. *J. Chem. Inf. Comput. Sci.* **1995,** *35,* 77–84.

[26] Haykin, S. *Neural Networks;* Pearson Education: Singapore, 2001.

[27] Patankar, S.; Jurs, P. Prediction of Glycine/NMDA Receptor Antagonist Inhibition from Molecular Structure. *J. Chem. Inf. Comput. Sci.* **2002,** *42,* 1053–1068.

[28] Fu, X.; Wang, L. Rule Extraction by Genetic Algorithms Based on a Simplified RBF Neural Network. In *Evolutionary Computation, Proceedings of the 2001 Congress on*, Vol. 2; IEEE: New York, NY, 2001.

[29] Gupta, A.; Park, S.; Lam, S. Generalized Analytic Rule Extraction for Feedforward Neural Networks. *IEEE Trans. Knowl. Data Eng.* **1999,** *11,* 985–991.

[30] Ishibuchi, H.; Nii, M.; Tanaka, K. Fuzzy-Arithmetic-Based Approach for Extracting Positive and Negative Linguistic Rules from Trained Neural Networks. In *Fuzzy Systems, Proceedings of the IEEE International Conference on*, Vol. 3; IEEE: New York, NY, 1999.

[31] Chen, P.; Mills, J. Modeling of Neural Networks in Feedback Systems Using Describing Functions. In *Neural Networks, International Conference on*, Vol. 2; IEEE: New York, NY, 1997.

[32] Yao, S.; Wei, C.; He, Z. Evolving Fuzzy Neural Networks for Extracting Rules. In *Fuzzy Systems, Proceedings of the Fifth IEEE International Conference on*, Vol. 1; IEEE: New York, NY, 1996.

[33] Carhart, R.; Smith, D.; Venkataraghavan, R. Atom Pairs as Molecular Features in Structure-Activity Studies: Definition and Application. *J. Chem. Inf. Comput. Sci.* **1985,** *25,* 64–73.

[34] Stanton, D.; Mattioni, B. E.; Knittel, J.; Jurs, P. Development and Use of Hydrophobic Surface Area (HSA) Descriptors for Computer Assisted Quantitative Structure-Activity and Structure-Property Relationships. *J. Chem. Inf. Comput. Sci.* **2004,** *44,* 1010–1023.

[35] Mosier, P. D.; Counterman, A. E.; Jurs, P. C.; Clemmer, D. E. Prediction of Peptide Ion Collision Cross Sections from Topological Molecular Structure and Amino Acid Parameters. *Anal. Chem.* **2002,** *74,* 1360–1370.

[36] Stanton, D.; Jurs, P. Development and Use of Charged Partial Surface Area Structural Descriptors in Computer Assissted Quantitative Structure Property Relationship Studies. *Anal. Chem.* **1990,** *62,* 2323–2329.

[37] Kier, L. A Shape Index from Molecular Graphs. *Quant. Struct.-Act. Relat. Pharmacol.,Chem. Biol.* **1985,** *4,* 109–116.

[38] Kier, L. Shape Indexes for Orders One and Three from Molecular Graphs. *Quant. Struct.-Act. Relat. Pharmacol.,Chem. Biol.* **1986,** *5,* 1–7.

[39] Kier, L. Distinguishing Atom Differences in a Molecular Graph Index. *Quant. Struct.-Act. Relat. Pharmacol.,Chem. Biol.* **1986,** *5,* 7–12.

[40] Liu, S.; Cao, C.; Li, Z. Approach to Estimation and Prediction for Normal Boiling Point (NBP) of Alkanes Based on a Novel Molecular Distance Edge (MDE) Vector, λ. *J. Chem. Inf. Comput. Sci.* **1998,** *38,* 387–394.

[41] Balaban, A. Higly Discriminating Distance Based Topological Index. *Chem. Phys. Lett.* **1982,** *89,* 399–404.

[42] Kier, L.; Hall, L. *Molecular Connectivity in Chemistry & Drug Research;* Academic Press: New York, 1976.

[43] Goldstein, H. *Classical Mechanics;* Addison Wesley: Reading, MA, 1950.

[44] Kier, L.; Hall, L.; Murray, W. Molecular Connectivity I: Relationship to Local Anesthesia. *J. Pharm. Sci.* **1975,** *64,*.

[45] Kier, L.; Hall, L. *Molecular Connectivity in Structure Activity Analysis;* John Wiley & Sons: Hertfordshire, England, 1986.

[46] Kier, L.; Hall, L. Molecular Connectivity VII: Specific Treatment to Heteroatoms. *J. Pharm. Sci.* **1976,** *65,* 1806–1809.

[47] Mattioni, B. E. *The Development of Quantitative Structure-Activity Relationship Models for Physical Property and Biological Activity Prediction of Organic Compounds,* PhD thesis, Pennsylvania State University, 2003.

[48] Randic, M. On Molecular Idenitification Numbers. *J. Chem. Inf. Comput. Sci.* **1984,** *24,* 164–175.

Chapter 9

Interpreting Computational Neural Network QSAR Models: A Detailed Interpretation of the Weights and Biases

9.1 Introduction

As we have seen in the preceding chapters, interpretability plays an important role in the QAR modeling process. The statistical and machine learning literature provide a wide variety of modeling methods to choose from, ranging from linear regression models to more complex techniques such as neural networks and random forests. The modeling techniques differ in a number of ways such as complexity, flexibility, accuracy and speed. A very important aspect of these models is interpretability. In the absence of an interpretation, the model can be used only for predictive purposes. This implies that structure-property information encoded in the model is not further utilized. In many cases, such as high throughput screens, such usage of the model is sufficient. But when models are developed with the aim of providing input to structure based drug design, more detailed information than just predicted values must be extracted from the model. That is, one would like to know what structure-property trends have been captured by the model. In other words, we would like to understand how the model correlates the input descriptors to the predicted activity. Furthermore, some measure of interpretability is needed to provide a sense of confidence regarding the soundness of the model, and it would provide evidence to support the use of a particular model in a given instance.

The degree of interpretability of QSAR models varies, depending on the modeling technique. In some cases, such as linear regression models, interpretation is relatively simple and can be carried out using a PLS technique developed by Stanton[1] and described in Chapter 3. In this case, the interpretation is detailed in the sense that one can extract information about how individual descriptors correlate to the predicted property. A

This work was published as Guha, R.; Jurs, P.C., "Interpreting Computational Neural Network QSAR Models: A Detailed Interpretation of the Weights and Biases", *J. Chem. Inf. Model.*, **2005**, *ASAP*.

number of applications of this technique have been reported.[1–3] In other models, the interpretation is not as detailed. This is the case for random forest[4] models. For these types of models interpretation is restricted to a summary of descriptor importance.[5] This summary ranks descriptors in order of importance to predictive ability. Thus, one does not get a detailed view of how the descriptors contribute to the predicted property.

The high predictive ability and flexibility of CNN models have made them very attractive to QSAR modelers. However the lack of interpretability has led to the general characterization of CNN models as *black boxes*. A number of attempts to extract information regarding the internal working of CNN models have been described. In some cases these methods are in the form of rule extractions.[6–8] These methods can be heuristic[9–11] in nature or analytical.[12] A number of these methods are focussed on specific types of neural networks.[8,9,13] Chastrette et al.[13] describe a method for interpreting a CNN model describing structure-musk odor relationships. Their approach was limited to a measure of contribution of the descriptors to the predicted value. Hervás et al.[14] describe a method interpretation that is focussed on a pruning algorithm. As a result the method is not applicable to CNN models developed using alternative algorithms.

The analysis of descriptor contributions is an approach that has been followed. Some of these approaches, such as that described by Chastrette et al.[13] provide only a broad view of which descriptors are important. Other approaches, however, have been devised that allow for a measure of correlation between input descriptors and the network output. An example is the method described by Mak et al.,[15] in which a form for the relative contribution of input neurons to the output value is developed. The relative values are then divided to obtain a measure of contribution to each hidden layer neuron. The result of this approach is that the contributions of the input neurons can be divided into negative or positive contributions. Chapter 8 described a method to obtain a measure of descriptor importance for neural network models based on a sensitivity analysis[16] of a trained network. Though similar in intent to the methods described by Chastrette et al. and Mak et al., the method provides an easily visualization method to understand which descriptors play the main role in the models predictive ability. However, the method also shares the main shortcoming with other approaches to measure descriptor importance (or contributions) in that it provides a very broad view and is not capable of describing in detail, the nature of the correlation between a given descriptor and the network output.

In this chapter we describe a method to interpret a CNN model in a detailed manner by considering the final, optimized weights and biases. As a result of this approach, the method is generalizable to different types of CNN algorithms that result in a set of weights and biases. Currently the method is restricted to the interpretation of 3-layer, feed-forward networks, though extension to more hidden layers is possible. The methodology is similar in concept to the PLS technique in that it interprets the weight matrix in a manner analogous to the interpretation of the X-weights in the PLS analysis. The method also shares certain characteristics with the method described by Mak et al. The next section describes the methodology in detail.

9.2 Methodology

The detailed CNN interpretation methodology was developed by attempting to mimic the procedure used for the interpretation of linear models using partial least squares. Though we have described the PLS methodology in detail in Chapter 3, we provide a short summary below.

The descriptors for the linear model are used to build a PLS model using a leave-one-out cross-validation method. The PLS model consists of a number of latent variables (components) which are linear combinations of the original descriptors. The number of components is equal to the number of input descriptors (assuming no overfitting has occurred). The results of the PLS analysis are summarized by two tables. The first table tabulates the cumulative variance and Q^2 values for each component. In many cases the first few components explain a large portion of the total variance (70% - 90%). As a result, the remaining components can be ignored. The second table lists the X-weights for each component. These are the weights used to linearly combine each input descriptor in a given component. Analysis of these weights allows one to understand how significantly, and in which direction, a given descriptor is correlated to the value predicted by that component. Finally, using plots of X-scores (projections of the observations along the rotated axes) versus Y-scores (that portion of the observed Y that is explained by that component) one can focus on the correlations between structural features and property for specific molecules.

9.2.1 Preliminaries

The CNN interpretation method is based on the assumption that the hidden layer neurons are analogous to the latent variables in a PLS model. Clearly, this is not a one-to-one correspondence due to the sigmoidal transfer function employed for each neuron in the CNN. By considering the weights connecting the input descriptors to a specific hidden layer neuron, we can then interpret how each descriptor correlates to the output of that hidden layer neuron. Finally, by defining the contribution of each hidden layer neuron to the output value of the network, we can determine which hidden layer neurons are the most significant and which ones can be ignored. The problem of interpreting a CNN model involves understanding how the output value of the network varies with the input values. This in turn is dependent on how the weights and biases modify the input values as they pass through the layers on the network. First, we present a brief analysis of how the input values will, in general, relate to the output value. We restrict ourselves to a 3-layer, fully-connected, feed-forward network.

The output value of a CNN for a given set of input values is obtained via a sigmoidal transfer function. Thus we can write the output value, O, as

$$O = \frac{1}{1 + \exp(-X)} \tag{9.1}$$

where X is the sum of weighted outputs from the hidden layer neurons. Denoting the output of each hidden layer neuron by $x_j^H,\ 1 \leq j \leq n_H$, where n_H is the number of hidden layer neurons, and the weight between each hidden layer neuron and the output neuron as $w_j^H,\ 1 \leq j \leq n_H$, we can write X as,

$$X = \sum_{j=1}^{n_H} w_j^H x_j^H$$

The above equation does not include a bias term and we provide a justification for ignoring the bias term below. Eq. 9.1 can be rewritten as

$$\begin{aligned} O &= \frac{1}{1 + \exp(-\sum_{j=1}^{n_H} w_j^H x_j^H)} \\ \frac{1}{O} &\sim \exp(-\sum_{j=1}^{n_H} w_j^H x_j^H) \\ O &\sim \exp\left(w_1^H x_1^H + \cdots + w_{n_H}^H x_{n_H}^H\right) \end{aligned} \tag{9.2}$$

where we drop the constant term since it does not affect the general trend between the output value and exponential term. From Eq. 9.2 we can see that O is a monotonic increasing function of the individual components, $w_j^h x_j^h$, of the argument. Keeping in mind that the output from each hidden neuron will be a positive number, Eq. 9.2 indicates that, if a certain hidden neuron has a large weight between itself and the output neuron, then the output from that hidden neuron will dominate the sum. This allows us to *order* the hidden neurons based on their contribution to the output value. Furthermore the sign of the weights indicate how the hidden neuron will affect the output value. Negative weights will correlate to smaller values of the output value and vice versa for positive weights.

9.2.2 Combining Weights

The above discussion applies to connections between the hidden layer and output layer. However it is clear that the same reasoning can be applied to the connections between the input and hidden layers. Thus, one way to consider the effect of the weights is to realise that the weights are cumulative. We denote the weights between the input layer neuron j and the hidden layer neuron i as

$$w_{ij},\ 1 \leq i \leq n_I \text{ and } 1 \leq j \leq n_H$$

where n_I is the number of input layer neurons (i.e., descriptors). Now, let us consider the value of the first input descriptor (for a specific observation). As this value goes from the first input neuron to the first hidden layer neuron, it will be affected by the weight, w_{11}. The value from the first hidden neuron is passed to the output neuron and is affected by the weight w_1^H. Thus we can qualitatively say, that, as the input value passes from the input layer to the output layer, it is affected by a weight denoted by $w_{11}w_1^H$. This is because a large positive value of w_{11} would cause the output of the first hidden neuron to be positively correlated with the first input descriptor. If w_1^H is also a large positive weight then the final output value would be positively correlated with the output value of the first hidden neuron and thus would also be positively correlated with the value of the first input neuron. Thus, we can consider the network as consisting of a connection between the first input neuron and the output neuron weighted by an effective weight equal to $w_{11}w_1^H$.

Similarly, for the same input value passing through the second hidden neuron and then to the output neuron, we can write the corresponding effective weight as $w_{12}w_2^H$.

In general the effective weight between the i^{th} input neuron and the output neuron, via the j^{th} hidden layer neuron will be $w_{ij}w_j^H$. Clearly the effective weights are gross simplifications, since we neglect the intermediate summations (over all neurons in a layer) and transfer functions. However as shown in the previous section we can see that the output value of the transfer function is a monotonic increasing function of the product of the weights and neuron outputs. More importantly, our main interest is in the *sign* of the effective weight, rather than its absolute value. The absolute value of the weights between the hidden layer neurons and the output neuron might be one indication of which hidden neuron is more important that another in terms of contribution to the final output value. However as pointed out above, the sign of the weights indicates the trend of the output value. Thus for example if the weights w_{11} and w_1^H are both positive we can expect that input values flowing down that path will show a positive correlation with the output value. If w_{11} and w_1^H are positive and negative respectively, one would expect that the net effect would be a negative correlation between the input values and output values.

9.2.3 Interpreting Effective weights

We can now consider two possible ways to use the effective weights to interpret the behavior of the CNN. From the preceding discussion we can write the effective weights in tabular form as shown in Table 9.1, where H1, H2 and H3 represent the first, second and third hidden neurons and I1, I2, I3 and I4 represent the input neurons. The first step in interpreting the effective weight matrix is to decide the order of the hidden layer neurons, in terms of their contributions to the output value of the net. We discuss hidden layer neuron contributions in detail below and for now we assume that the order of importance of the hidden layer neurons is given by H1 > H2 > H3. Thus, the first hidden neuron is the main contributor to the output value. Next we consider the first column. If the value in a given row is higher than the others it implies that the corresponding input neuron will contribute more to the hidden layer neurons. Since we have already ordered the hidden neurons in terms of their contribution to the output, this means that we can say (indirectly) which input neuron is contributing more to the output. Furthermore, the sign of the element will indicate whether high values of that input neuron correspond to high or low values of the output value. The approach is similar to the PLS interpretation scheme, especially if we consider the hidden layer neurons to be a transformed (via the transfer function) set of latent variables.

9.2.4 The Bias Term

In the preceding discussion we have ignored the role of the bias term when considering effective weights. We now provide a justification for this approach. The input to a given hidden neuron consists of the weighted outputs of the input neurons plus the bias term. As a result, the effective weights for the input neurons, as they pass through a given hidden neuron, should consider the bias term. However, if we consider the effective weights for individual input neurons, we must partition the bias term between these input neurons. The simplest way of partitioning the bias term would be to simply divide the bias term evenly between the input neurons. As a result, for n_I input neurons the effective weights between them and the j^{th} hidden neuron would include b_j/n_I where b_j is the bias term for that hidden neuron. The net result is that for the j^{th} hidden neuron, the effect of the bias term would be the same for all input neurons connected to it. As a result, it is equivalent to ignoring the bias term when considering effective weights. Clearly this is based on the assumption that the bias for a given hidden neuron can be equipartitioned between the input neurons. A priori, there is no reason for choosing an alternative partitioning scheme.

A more rigorous approach is to consider that fact that a bias term is effectively an intercept term. If the hidden neurons contained linear transfer functions, the bias term is precisely an intercept term. The inputs to a hidden neurons form a p-dimensional space and the result of the activation function for a hidden neuron is to draw a hyperplane through the input space. One side of this hyperplane represents the *off* output and the other side represents the *on* output. This description also holds for sigmoidal activation functions in which case the two sides of the hyperplane would correspond to the extreme ends of the functions domain. The region close to the hyperplane would correspond to the intermediate values of the activation function. In the absence of the bias term this hyperplane passes through the origin of the input space. However, when bias terms are included, they merely translate the hyperplanes from the origin. That is, they do not change the form of the hyperplane. In effect, the bias term is a constant. Now, one property of neural networks (more specifically, multi layer perceptrons) is universal function approximation.[17] For this to be true, the bias term must be included. However, it has been shown by Hornik[18] that a sufficient condition for this property to be true in the absence of bias terms is that the derivatives of the activation function must be non-zero at the origin. For a sigmoidal activation function, this implies that the bias term can simply be a constant value as opposed to a trainable weight. Clearly, if the

bias term can be considered as a constant (i.e., training is not required), this implies that it would not affect the interpretation of the optimized weights. Thus, viewing the bias terms in the context of partitioning or in the context of the universal approximation property indicates that development of the effective weights without including the bias terms is a valid approach.

9.2.5 Ranking Hidden Neurons

An important component of the interpretation method is the ranking of the hidden neurons, which is necessary as all hidden neurons will not contribute to the output value equally. The contributions of the input neurons to the output value, via those hidden neurons with lower contributions, will be diminished. A number of methods to determine the relative contribution of input neurons have been described in the literature. These methods can be applied to the case of the hidden layer. For example, the method described by Garson[19] calculates a measure of the relative contribution of the i^{th} input neuron to the k^{th} output neuron and places more stress on the connections between the hidden and output layers. Yoon et al[20] extended this approach but still focussed on contributions of input descriptors to the output via the hidden layer neurons. A common feature of both approaches is their empirical nature. That is, the final contribution values are obtained by using the original training set.

Our first attempt at designing a ranking method followed the approach of Garson. In this case we defined the squared relative contribution of the j^{th} hidden neuron for the k^{th} example in the training set to be

$$SRC_{kj} = \frac{\left({}^{k}x_j^H w_j^H\right)^2}{\sum_{j=1}^{n_H}\left({}^{k}x_j^H w_j^H\right)^2 + b^2} \tag{9.3}$$

where x_j^H and w_j^H are the output and weight to the output neuron for of the j^{th} hidden neuron respectively, b is the bias term for the output neuron and n_H is the number of hidden layer neurons. The superscript k indicates that the output of the j^{th} neuron is for the k^{th} example. The final value of the squared relative contribution for the j^{th} hidden neuron was given by

$$SRC_j = \frac{1}{n}\sum_{k=1}^{n} SRC_{kj} \tag{9.4}$$

where n is the number of examples in the training set. However, interpretations developed from the above ranking were not consistent with the interpretations obtained from the linear models.

An alternative approach that we considered was not empirical in nature. That is, it did not directly utilize the dataset used to build the model. In this approach we considered the fact that the contributions of a given hidden layer neuron depends not only on the nature of its contribution to the output neuron, but also on the nature of the contributions to the hidden neuron from the preceding input layer. This is implicitly considered in the empirical approach described. In this approach we considered the overall contribution of a hidden neuron to the output by taking into account all the effective weights associated with this hidden neuron. That is, the contribution of the j^{th} hidden neuron was initially defined as

$$CV_j = \frac{1}{n_I} \sum_{i=1}^{n_I} w_{ij} w_j^H \tag{9.5}$$

where n_I is the number of input neurons, w_{ij} is the weight between the i^{th} input neuron and j^{th} hidden neuron and w_j^H is the weight between the j^{th} hidden neuron and the output neuron. The above equation simply represents the column means of the effective weight matrix. The resultant values are signed and the absolute value of these contribution values can be used to rank the hidden neurons. However, to make the relative contributions of the hidden neurons clearer we considered the values obtained from Eq. 9.5 as squared contribution values defined as

$$SCV_j = \frac{CV_j^2}{\sum_{j=1}^{n_H} CV_j^2} \tag{9.6}$$

The result of this transformation is that the SCV_j values sum to 1.0. Consequently, the SCV_j values provide a clearer view of the contributions of the hidden neurons and allow us to possibly ignore hidden neurons that have very small values of SCV_j.

One aspect of this approach is that we do not take into account the bias terms. Clearly, this approach is not utilizing all the information present within the neural network. There are two reasons why the bias term should be taken into account when ranking hidden neurons. First, most reported measures of contribution are empirical in nature and thus implicitly take into account the bias terms. Second, since we are focussing on the contribution made by a given hidden neuron, we need to consider all

the effects acting through the hidden neuron. Since the bias terms corresponding to the hidden layer can be considered as weights coming form an extra (constant) input neuron, the effective weights being summed in Eq. 9.5, should include an extra term corresponding to the bias term for that hidden neuron. Thus if we denote the bias term for the j^{th} hidden neuron as b_j then Eq. 9.5 can be rewritten as

$$CV_j = \frac{1}{n_I + 1}\left(\sum_{i=1}^{n_I} w_{ij} w_j^H + b_j w_j^H\right) \tag{9.7}$$

Using this equation, values for SCV_j can be calculated using Eq. 9.6

9.2.6 Validation

To ensure that the methodology provides valid interpretations we compared the results of the method to interpretations developed for linear models. For a given QSAR problem, the descriptor subsets that lead to the best linear model are generally not the same as those that lead to the best CNN model. However, comparing interpretations of CNN and linear models with different descriptors would lead to a less accurate comparison. Furthermore, one would expect that given the same descriptors, both CNN and linear models should capture the structure-property trends present in the dataset in a similar fashion. If the interpretation of these trends in the CNN model does not match those developed using the linear model, the discrepancy would indicate that the CNN interpretation methodology is flawed. As a result, we developed the CNN models using the same subset of descriptors that were present in the corresponding linear models. The CNN models built using these descriptors are not necessarily the best (in terms of training set and prediction set performance). However, this work focusses on the extraction of structure-property trends in a human understandable format rather than investigating the predictive power of the CNN models. In this respect we feel that the comparison of the CNN interpretations to those made for linear models using the same set of descriptors is a valid procedure.

The linear models were developed using the ADAPT methodology, described in Chapter 3. This involved the use of a simulated annealing[21,22] algorithm to search for good descriptor subsets. The models were then interpreted using the PLS analysis technique described above. As mentioned, the CNN models used the same descriptors that were present in their corresponding linear models. For each dataset a number of CNN models with different architectures (i.e., different numbers of hidden neurons) were built.

The architectures were limited by considering a rule of thumb that indicates that the total number of parameters (weights and biases) should not be more than half the size of the training set. The final architecture for each CNN model was chosen by considering the cost function defined by Eq. 2.6. The architecture that gave the lowest value of the cost function was chosen as the best CNN model for that dataset.

9.3 Datasets

We considered three datasets. The first dataset consisted of a 147 member subset of the DIPPR boiling point dataset studied by Goll et al.[23] The dependent variable for this dataset ranged from 145.1 K to 653.1 K. The original work reported linear and nonlinear models. However no interpretations of these models were provided. We consider the linear model for this dataset that was described in Chapter 8. Though that chapter also reported a CNN model, we develop a new CNN model so that we would be able to obtain a more direct comparison between the final interpretations as described above.

The second dataset consisted of 97 compounds studied by Stanton et al.[24] These compounds were studied for their ability to cross the blood-brain barrier and the modeled property was the logarithm of the blood-brain partition coefficient ($\log(BB)$). The dependent variable in this dataset ranged from -2.00 to 1.44 log units. The previously published work reported a linear model and an associated PLS based interpretation.

The final dataset consisted of 136 compounds studied by Patel et al.[25] The work considered the skin permeability of the 136 compounds. The dependent variable for this dataset was the logarithm of the permeability coefficient, $\log(K_p)$ and ranged from -5.03 to -0.85 log units. Though the paper reported a set of linear models, we developed a new linear model using a variety of descriptors including hydrophobic surface area descriptors.[24,26] A PLS analysis of this model is also presented for comparison to the interpretation of the corresponding CNN model.

9.4 Results

For each dataset we present a summary of the linear model and the associated PLS interpretation. We then describe the neural network model built using the dataset and descriptors from the linear models, and present the corresponding interpretation. For all models the descriptors are summarized in Table 9.2.

9.4.1 DIPPR Dataset

The statistics of the linear model are summarized in Table 9.3. The descriptors are shown in Table 9.2. The R^2 and RMSE values were 0.98 and 9.98, respectively. The F-statistics (on 7 and 139 degrees of freedom) was 1001 which is greater than the critical value of 2.78 ($\alpha = 0.05$ level). The model is thus statistically valid. Tables 9.4 and 9.5 summarize the results of the PLS analysis for the linear model. The Q^2 column in Table 9.4 indicates that the first two components explain approximately 95% of the structure-property relationship (SPR) encoded by the model. As a result, the bulk of the linear interpretation is provided by these components. If we now look at the column for the first component in Table 9.5 we see that the most important descriptors are MW (molecular weight) and V4P-5 (4^{th} order valence path connectivity index). Both these descriptors characterize molecular size and it is evident that larger values of these descriptors correlate to higher values of the boiling point. Considering the second component we see that the most important descriptors are now RSHM and PNSA-3. The former characterizes hydrogen bonding ability and the latter is a measure of the charge weighted partial negative surface area. The negative sign for PNSA-3 indicates that molecules with smaller values of the descriptor (i.e., having smaller charge weighted partial negative surface area) should have lower boiling points. On the other hand, the positive sign for RSHM indicates that molecules with better hydrogen bonding ability should have higher boiling points, which is in accord with experimental observations. In summary, the linear model encodes two main SPR's. The first trend is dispersion forces, in which atomic contributions to these forces are individually weak but for larger molecules the greater number of interactions leads to a larger attractive force. The other main trend is the attractive forces mainly due to hydrogen bonding. Clearly, this description of the SPR is not new or novel. However now that we know what type of descriptors contribute to the SPR and the nature of the correlations we can compare these observations with those obtained from the CNN model.

The CNN model that was developed for this dataset had a 7–4–1 architecure. As described above, the descriptors for the CNN model were the same as those used in the linear model. The statistics for the training, cross-validation and prediction sets are shown in Table 9.6. The effective weight matrix for this model is shown in Table 9.7. The columns correspond to the hidden neurons and are ordered by the SCV values described previously, and they are shown in the last row of the table. The SCV values indicate that the first and third hidden neurons are the most important, whereas the second

and fourth hidden neurons play a lesser role. The use of the SCV values in choosing which hidden neurons to concentrate on is analogous to the use of the Q^2 value to focus on components in the PLS approach. Considering the first column in Table 9.7 we see that the most weighted descriptors are V4P-4, RSHM and MW. All three descriptors have positive weights indicating a that these descriptors are positively correlated with boiling point. When we consider the second column (the third hidden neuron) we see that the two of the three most important descriptors are the same as in the first hidden neuron. Since these have the same signs as before, we may ignore them and consider the most important descriptor not already considered. This descriptor is PNSA-3, and it is negatively correlated to the boiling point. It is clear that the types of descriptors, as well as their correlations, that play the main roles in the SPR encoded by the CNN model are the same as described by the linear model. The main difference is that the relative importance of the descriptors, over the hidden neurons being considered, are different from that described in the linear model. For example, the linear model indicates that PNSA-3 plays a very important role in the SPR, whereas the CNN accords it a less significant role. On the other hand, the hydrogen bonding ability described by RSHM plays a very important role in the CNN, making up for the absence of the charged surface area descriptor. Similarly, the relative importance of the V4P-5 and MW descriptors are swapped in the two interpretations, but since both characterize size, the main SPR trends for the dataset are explained in a similar fashion by both models. These differences are not unexpected, since the CNN correlates the descriptors in a nonlinear fashion. Thus it is expected that the relative roles played by each descriptor in the nonlinear relationship will be different when compared to the linear model. The point to note is, that, though MW is relegated to a less important role, the main SPR trends extracted from the CNN model by this interpretation technique are identical to those present in the linear model.

9.4.2 BBB Dataset

The linear model and associated PLS interpretation are described in the original work.[24] However, we summarize the statistical results of the original model in Table 9.8. The R^2 for the model was 0.78 and the F-statistic was 80.7 (on 4 and 92 degrees of freedom) which was greater than the critical value of 2.47 ($\alpha = 0.05$). The results of the PLS analysis are presented in Tables 9.9 and 9.10. From Table 9.9 we see that the first two components explain 76% of the total variance in the observed property thus allowing us to ignore the remaining components. From Table 9.10 we see that in

the first component the most weighted descriptors are PNHS-3 (a hydrophobic surface area descriptor), NDB (count of double bonds) and WNSA-3 (a charged partial surface area descriptor, characterizing the partial negative surface area). Clearly, larger values of PNHS-3 and WNSA-3 will correlate to larger values of the property whereas lower values of NDB will correlate to larger values of the property. In component 2 we see that WNSA-3 is opposite in sign. This indicates that the second component makes up for over-predictions made by the first component. Similar reasoning can be applied to the weight for V4P-5 in the second component. Some large hydrophobic molecules are under-estimated by component 1. In component 2 however, the positive weight for V4P-5 (which is a measure of branching and thus size) indicates that larger molecules will have higher penetration ability. Therefore the second component makes up for under-estimation of large hydrophobic molecules by the first one. In brief, the SPR trends captured by the linear model indicate that smaller hydrophobic molecules will better penetrate the BBB compared to larger hydrophilic molecules. These trends are discussed in more detail in the original work.[24]

The CNN model developed for this dataset had a 4–4–1 architecture. The statistics for this model are presented in Table 9.6 and the effective weight matrix is shown in Table 9.11. The SCV values for the hidden neurons are shown in the last row of Table 9.11. They indicate that the first and second hidden neurons contribute to the bulk of the SPR encoded by the model. If we consider the weights for the first hidden neuron (first column) we see, in general, the same correlations as described in the linear model. Both PNHS-3 and WNSA-3 are positively correlated with the predicted property and NDB is negatively correlated. However the difference we see here is that V4P-5 is one of the most important descriptors and is positively correlated with the predicted property. On the other hand PNHS-3 plays a much smaller role in this model than in the linear model. If we consider the second hidden neuron (second column), we see that the weight for V4P-5 is now lower and that for NDB has increased. One can consider this as the CNN attempting to downplay the increased size effects described by V4P-5. When we consider the fourth hidden neuron we see that the V4P-5 now has a negative weight and thus serves to balance the over-estimation of the property for larger molecules made by the first two hidden neurons. Overall, we see that the main trends described by the CNN model indicate that fewer double bonds and more hydrophobicity lead to higher ability to penetrate the BBB, though it does appear that the model focuses on a positive correlation between size and $\log(BB)$ values via the V4P-5 descriptor. This is quite similar

to the conclusions obtained from the PLS interpretation of the linear model. Some differences are present - mainly in the context of size (described by V4P-5) and the relative importance of the descriptors for a given hidden neuron. As described above, this is not surprising given that the nonlinear relationship between the descriptors generated by the CNN is significantly different from the linear relationship described by the original regression model. However, the fact that the description of the main SPR trends encoded within the CNN model compare well with those of the linear model, serve to confirm our assumption that both models should encode similar trends as well as the validity of this interpretation technique to extract these trends. Furthermore, the structure-property trends extracted from both types of models by the repspective interpretation techniques are consistent with physical descriptions of the factors that are believed to affect the transport of drugs across the blood brain barrier[27, 28]

9.4.3 Skin Permeability Dataset

This dataset was originally studied by Patel et al.[25] where they developed a linear regression model using 158 compounds. However, owing to the presence of outliers, the final models were built using 143 compounds. We considered the original 158 compounds and chose a 136 member subset to work with. The linear model we developed for this dataset is summarized in Table 9.12. The R^2 value for this model was 0.84 and the F-statistic was 97.5 on 7 and 128 degrees of freedom which was greater than the critical value of 2.08 ($\alpha = 0.05$) indicating that the model was statistically valid. We then developed a PLS interpretation of this linear model and the results of the PLS model are summarized in Tables 9.13 and 9.14. The Q^2 values in Table 9.13 indicate that the first three components describe the bulk of the SPR. If we now consider Table 9.14 we see that the most important descriptors in the first component are MOLC-9, FPSA-2 and RNHS. MOLC-9 represents Balabans J topological index which is derived from the distance connectivity matrix and characterizes molecular branching. Smaller values of this descriptor indicate smaller or more linear compounds. The FPSA-2 descriptor characterizes the relative partial positive surface area. The negative weight for this descriptor indicates that molecules with smaller partial positive surface areas will be more active. This descriptor characterizes molecules like 2,4,6-trichlorophenol whose molecular surface has a large number of partial negative charged regions. Finally the RNHS descriptor characterizes the hydrophilic surface area. This descriptor serves to balance the effects of FPSA-2 and this can be seen when comparing the activities for

2,4,6-trichlorophenol and 3,4-dimethylphenol (Table 9.1). It is clear that both are of similar size and both have similar activities. However, if FPSA-2 were acting alone, the high value of this descriptor for 3,4-dimethylphenol (due to the partial positive charges on the methyl groups) would lead to a significantly lower activity. However, RNHS indicates that not all small hydrophobic molecules will have negatively charged atoms. Thus the first components indicates that smaller and more hydrophobic molecules will exhibit higher activities.

If we now consider the second component we see that the most weighted descriptors are PPHS, SA and WPHS-3. PPHS measures the total hydrophobic surface area and WPHS-3 is defined as the surface weighted hydrophobic surface area. The positive weights on these descriptors indicate that a larger hydrophobic surface area is correlated positively with activity. SA, which measures molecular surface area, is positively weighted indicating larger molecules are more active. The role of this component is to account for some larger molecules observed to be moderately active (Table 9.2). Essentially, the larger size of these molecules leads to a larger hydrophobic surface area which enhances permeability. The component thus corrects for the under-estimation of the larger molecules by component 1 (such as fentanyl and sulfentanil) by taking into account their higher hydrophobicity and also corrects for the over-estimation of smaller molecules (such as methanol and urea) by component 1 by taking into account their lower hydrophobicity.

The third component mainly corrects for the over-estimation of hexachlorobutadiene and hexachloroethane by component 2 due to emphasis on the hydrophobic surface area. This is corrected for by component 3 by the negative weights for FPSA-2 and WPHS-3.

Thus the main conclusion that can be drawn from the linear model is that molecular size and hydrophobicity are two key characteristics that appear to explain the observed skin permeability of these compounds. This is consistent with the conclusions of Patel et al.[25] and also with the general understanding regarding the mechanism of skin permeation.

The CNN model developed for this dataset had a 7–5–1 architecture and the statistics are reported in Table 9.6. The effective weight matrix is shown in Table 9.15. In the case of this model, we see that the SCV value for the 5^{th} hidden neuron is nearly six times larger than that for the next most important neuron. If we consider the most important hidden neuron (5) we see that the most weighted descriptors are FPSA-2, NN (the number of nitrogens) and PPHS. The signs of these effective weights are the

same as described by the PLS analysis of the linear model. That is, these descriptors have the same effect on the output of the model in both the linear and nonlinear cases. Thus, the most important hidden neuron indicates that molecules with smaller polar surface area and larger hydrophobic surface area will exhibit higher activity. Moving onto the next most important hidden neuron, we see that the most weighted descriptors are SA, PPHS, MOLC-9. It is clear that this hidden neuron focuses on size effects. However, the negative weight for surface area indicates that larger molecules will be more active. This is a valid conclusion since the dataset does indeed have some larger molecules which are moderately active. This conclusion is further justified by the fact that a larger molecule would have a correspondingly larger hydrophobic surface area, which as the positive weight for PPHS indicates, will lead to higher activity. At the same time, all large molecules do not exhibit high activities. Thus the effect of the SA descriptor is balanced by the positive weight for the MOLC-9 descriptor. Since larger values of MOLC-9 correlate to smaller molecules, the effect of the MOLC-9 descriptor balances the SA descriptor, ensuring that this hidden neuron does not predict all large molecules as active.

In the next most important hidden neuron (4) we see that the most weighted descriptors are RNHS, FPSA-2, PPHS and MOLC-9. In this hidden neuron, MOLC-9 describes the effect of size and indicates that smaller molecules will exhibit higher activity. However, the positive weight FPSA-2 indicates that molecules with larger partially positive charged surface area will exhibit higher activity. When we also consider the negative weight for PPHS (indicating more active molecules should have lower hydrophobic surface area) we see that this neuron focuses mainly on smaller, more polar molecules. This trend is reinforced to some extent by the negative weight for RNHS. RNHS describes both hydrophilic and partial negatively charged regions regions of a molecule. Due to the design of the descriptor, the negative sign on this descriptor indicates that molecules with smaller partial negatively charged surface area and more hydrophilic atoms will exhibit relatively higher activities. At the same time if we consider the SCV value for this hidden neuron we see that it is just 2% of the SCV for the most important hidden neuron. One would thus expect that this neuron would not provide very detailed information regarding the SPR encoded in the model. Similar reasoning can be applied to the last two columns of Table 9.15.

The interpretation of the CNN model described here matches quite closely with that of the linear model. The main difference is in the ordering of the important trends. As described before, this is not surprising due to the nonlinear encoding of the structure

property relationships by the CNN model. However, though the above description is quite detailed and by allows us to look at descriptor values for individual observations and understand why they are predicted to display greater or less skin permeation, a visual approach to understanding the effects of each hidden neuron, analogous to score plots[1] in the PLS interpretation scheme, would be useful. One approach to this problem is to generate plots using the effective weights.

9.4.4 Score Plots

As described above, the use of the effective weights *linearizes* the neural network. In effect, the network is transformed into a set of connections between input descriptors and the output neuron, ignoring nonlinear transfer functions. The *pseudo-network* can be used to generate a set of score values for each hidden neuron. For the k^{th} member of the dataset, the score for that member using the j^{th} hidden neuron can be defined as

$$score_{kj} = \sum_{i=1}^{n_I} w_{ij} w_j^H x_{ki} \tag{9.8}$$

where $w_{ij}w_j^H$ is simply the effective weight for the i^{th} input neuron (see Table 9.1), n_I is the number of input descriptors and x_{ki} is the value of the i^{th} descriptor for the k^{th} member of the dataset. The result of Eq. 9.8 is that for each hidden neuron, a set of scores are obtained for the dataset. Clearly, these are not meant to quantitatively model the observed properties well. However, our interest in lies in the qualitative behavior of the scores. That is, we expect that if a compound has a high observed activity, its score value should be high and vice versa for compounds with low observed activity. Thus a plot of the scores for a given hidden neuron versus the observed property should lie along the 1:1. Points lying in the lower right quadrant would represent compounds over-estimated by the hidden neuron and points lying the in the upper left quadrant would represent compounds under-estimated by the hidden neuron.

We tested this approach by creating score plots for the three most important hidden neurons for the CNN model developed for the skin permeability dataset. These are shown in Figs. 9.5, 9.6 and 9.7. Considering the plot for the 5^{th} hidden neuron we see that the plot does exhibit the behavior we expect. Compounds **21**, **43** and **75** are predicted as active and **42**, **46** and **135** are predicted as inactive. The structures for these compounds are shown in Fig. 9.3. As described previously, active compounds will be characterized by smaller size and increased hydrophobicity. As Fig. 9.3 shows,

active compounds do indeed have a hydrophobic benzene ring. In contrast the inactive compounds are generally larger and, more significantly, have a number of polar regions. An interesting case is urea (compound **135**). This is a very small compound, but it is dominated by polar groups, responsible for hydrogen bonding. The fact that the neuron predicts this compound correctly as inactive is due to the fact that this neuron mainly stresses the FPSA-2 and NN descriptors. As seen from Table 9.15, the negative weights indicate that a larger number of polar groups would inhibit activity. As a result, though compound **135** is small, the polar effects outweigh the size effect. Fig. 9.5 also indicates that compounds **81**, **114**, **69** and **77** are all mispredicted. The first two are over-estimated and the last two are under-estimated.

If we now consider the score plot for the 2nd hidden neuron in Fig. 9.6, we see the four mispredicted compounds, mentioned above, are now more correctly predicted. However **81** does appear to be over-estimated. Apart from these cases, the majority of the compounds do not appear to be well predicted. We believe that this can be explained by the very low contribution value of this hidden neuron compare to the 5th hidden neuron. Due to the very low SCV value for this neuron, we believe that it does not have significant explanatory power. The structures of the active and inactive compounds are compared in Fig. 9.4. As described above, the main focus of the 2nd hidden neuron is to account for compounds which are relatively larger but also moderately active. As can be seen from the structures, though compounds are **78**, **81** and **114** are significantly larger than the active compounds in the preceding component, they are indeed moderately active. Correspondingly, this hidden neuron is also able to account for the low activity of a number of small compounds (**72** and **77**). Though this hidden neuron mispredicts a number of compounds, the majority have already correctly predicted in the preceding hidden neuron. A number of them, such as **87**, are corrected by the next most important hidden neuron.

Considering the score plot for the 4th hidden neuron we see that, though it does correctly predict a number of compounds as active, it performs poorly on inactive compounds. Once again, we believe that the low contribution value (1% of the SCV of the most important hidden neuron) indicates that it will not have significant explanatory power. However, it does correct for the misprediction of **87** by the 2nd hidden neuron. In addition, compound **81** is now shifted closer to the 1:1 line, correcting for the slight overestimation by the preceding hidden neuron. Score plots for the remaining hidden neurons can be similarly analysed, though we observed that they did not explain any

significant trends and rather corrected for a few mispredictions by the preceding 3 hidden neurons.

The above discussion shows that the score plots derived from the effective weights help provide a more visual approach to the interpretation of a CNN model. Coupled with an analysis of the effective weight table, a CNN model can be interpreted in a very focused, compound-wise manner.

9.5 Discussion & Conclusions

The CNN interpretation methodology that we have presented provides a means for using CNN models both for predictive purposes as well as for understanding structure-property trends present in the dataset. The methodology is similar in concept to the PLS interpretation method for linear regression models. The analogy to the PLS method is strengthened when we consider that the hidden neurons are analogous to latent variables (and in the case of linear transfer functions, are identical). Though a number of approaches to understanding a CNN model exist in the literature, our approach provides a detailed view of the effect of the input descriptors as they act via each hidden neuron. Furthermore, previous approaches are empirical in the sense that they require the direct use of the training set to determine the importance of input or hidden neurons. The method described here avoids this by making use of the effective weights only. A justification for this approach is that the weights and biases in the final CNN model are derived from the structure-property trends present in the data. As a result the optimized weights and biases already contain the information regarding the SPR's and thus subsequent use of the training set to develop the interpretation is unnecessary. However, the training set is used to generate the hidden neuron score plots which can be used to focus on the contributions of individual hidden neurons to the overall predictive behavior of the model and understand the behavior of the hidden neurons by considering specific compounds.

The method was validated on three datasets covering physical and biological properties. Interpretations from the CNN model were compared to linear models built for these datasets (using the same descriptors that were present in the CNN model) and it can be seen that the structure property trends described by both models are in very close agreement. The main differences between the interpretations is in the importance ascribed to specific descriptors. That is, the most important descriptor in the most important latent variable in the PLS interpretation might not occupy the same position in

the CNN interpretation. This is not surprising due to the fact that the neural network combines the input descriptors nonlinearly and thus the role of the individual descriptors in the nonlinear relationship may be different from that played in a linear relationship.

Another important aspect of this study was that we considered CNN models which contained the same descriptors as the linear models. The linear models were developed using a genetic algorithm for feature selection and were thus optimal linear models. This is not the case for the corresponding neural network models. This is due to the fact that, in general, when a CNN routine is linked to the genetic algorithm, the optimal descriptor subsets differ from the case where the objective function for the genetic algorithm is a linear regression function. As a result, in the examples we considered, the CNN models were not necessarily optimal and hence the interpretations may differ to some extent when optimal models are built for the datasets. However structure property trends are a feature of the data rather than the model describing the data. Thus even if optimal descriptor subsets are considered, it is expected that these descriptors will capture the structure property trends present in the dataset, albeit with greater accuracy. Hence, it is expected that interpretations from the optimal CNN models will not differ significantly from those described here.

However, there is one aspect that should be considered when interpreting CNN models using this method. The definition of effective weights ignores the effect of the nonlinear transfer function for each neuron. In effect, the effective weights linearize the model. As a result the interpretation does not provide a full description of the nonlinear relationships between structural features and the property. That is, some information regarding the encoded SPR is lost. We feel that the tradeoff between interpretability and information loss is justified due to the simple nature of method. To fully describe the nonlinear encoding of an SPR would essentially require that the CNN model be analyzed to generate a functional form corresponding to the encoded SPR. The neural network literature describes a number of approaches to rule extraction in the form of if-then rules.[8–11,29] as well as some instances of analytical rule extraction.[12,30,31] As mentioned previously, most of the previous approaches to the interpretation of neural networks or extraction of rules from neural networks are focused on specific types of neural network algorithms. In addition, a number of the rule extraction methods described in the literature are carried out by analysing the neural network with the help of a genetic algorithm[29,32] or by decision trees,[11] adding an extra layer of complexity to the methodology. The method described here is quite general as it requires only the optimized weights and biases from the network. The only current restriction on the method

is that the neural network must have a single hidden layer. However, the methodology described in this chapter can be extended to the case of multiple hidden layers though the complexity of the treatment will correspondingly increase.

The interpretation method described in this work expands the role of CNN models in the QSAR modeling field. The black box reputation of CNN models has led to their main usage as predictive tools with no explanation of the structure-property trends that are encoded within the model. We believe that this interpretation method will allow for a detailed understanding of the structure-property trends encoded in CNN models allowing them to be used for both predictive and design purposes.

Table 9.1. Tabular representation of effective weights for a hypothetical 4–3–1 CNN model. I1, I2, I3 and I4 represent the four input neurons (descriptors). w_{ij} represents the weight for the connection between the i^{th} input neuron and the j^{th} hidden neuron. w_j^H represents the weight between the j^{th} hidden neuron and the output neuron. For this example i ranges from 1 to 4 and j ranges from 1 to 3

	Hidden Neuron		
	1	2	3
I1	$w_{11}w_1^H$	$w_{12}w_2^H$	$w_{13}w_3^H$
I2	$w_{21}w_1^H$	$w_{22}w_2^H$	$w_{23}w_3^H$
I3	$w_{31}w_1^H$	$w_{32}w_2^H$	$w_{33}w_3^H$
I4	$w_{41}w_1^H$	$w_{42}w_2^H$	$w_{43}w_3^H$

Table 9.2: A glossary of the descriptors used in this study

Descriptor code	Meaning	Reference
DPHS	The difference between the hydrophobic and hydrophilic surface area	24, 26
FPSA-2	Charge weighted partial positive surface area divided by the total surface area	33
MOLC-9	Balaban J topological index	34, 35
MW	Molecular weight	
NDB	Number of double bonds	
NN	Number of nitrogens	
PNHS-3	Atomic constant weighted hydrophilic surface area	24, 26
PPHS	Total molecular hydrophobic surface area	24, 26
SA	Surface are of the molecule	
S4PC-12	4^{th} order simple path cluster molecular connectivity index	36–38
V4P-5	4^{th} order valence path molecular connectivity index	36–38
WNSA-3	Difference between the partial negative surface area and the sum of the surface area on negative parts of molecule multiplied by the total molecular surface area	33
WPHS-3	Surface weighted hydrophobic surface area	24, 26
WTPT-2	Molecular ID divided by the total number of atoms	39
RNHS	Product of the surface area for the most negative atom and the most hydrophilic atom constant divided by the sum of the hydrophilic constants	24, 26
RSHM	Fraction of the total molecular surface area associated with hydrogen bond acceptor groups	33

Table 9.3. Summary of the linear regression model developed for the DIPPR dataset

	Estimate	Std. Error	t
(Intercept)	−215.09	29.45	−7.30
PNSA-3	−3.56	0.21	−16.90
RSHM	608.07	21.30	28.55
V4P-5	19.57	3.30	5.92
S4PC-12	12.08	1.57	7.69
MW	0.57	0.061	9.42
WTPT-2	236.10	16.57	14.25
DPHS	0.19	0.02	7.07

Table 9.4. Summary of the PLS analysis based on the linear regression model developed for the DIPPR dataset

Components	Error SS	R^2	PRESS	Q^2
1	94868.50	0.86	99647.60	0.85
2	26221.60	0.96	29046.70	0.95
3	16614.80	0.97	19303.30	0.97
4	14670.80	0.97	17027.60	0.97
5	14032.50	0.97	16281.30	0.97
6	13775.90	0.98	15870.60	0.97
7	13570.90	0.98	15653.00	0.97

Table 9.5. The X-weights for the PLS components from the PLS analysis summarized in Table 9.4

	Component						
Descriptor	1	2	3	4	5	6	7
PNSA-3	-0.30	-0.42	0.20	-0.25	0.25	-0.73	-0.12
RSHM	0.19	0.77	0.34	-0.03	0.22	-0.37	0.20
V4P-5	0.48	-0.15	-0.07	-0.66	-0.36	-0.09	0.38
S4PC-12	0.28	-0.07	-0.57	0.53	-0.03	-0.46	0.26
MW	0.49	-0.085	0.36	0.24	-0.39	-0.17	-0.60
WTPT-2	0.48	-0.05	-0.26	-0.22	0.70	0.13	-0.35
DPHS	0.26	-0.41	0.54	0.32	0.29	0.18	0.48

Table 9.6. Summary of the architectures and statistics for the CNN models developed for the datasets considered in this study. In all cases, the input descriptors were the same as those used in the corresponding linear models

		RMSE			R^2		
Dataset	Architecture	TSET	CVSET	PSET	TSET	CVSET	PSET
DIPPR	7–4–1	15.21	38.51	15.07	0.91	0.45	0.94
BBB	4–4–1	0.25	0.38	0.47	0.88	0.88	0.74
Skin	7–5–1	0.23	0.27	0.31	0.94	0.93	0.91

Table 9.7. The effective weight matrix for the 7–4–1 CNN model developed for the DIPPR dataset. The columns (hidden neurons) are ordered by the the squared contribution values (SCV) shown in the last row. Note that the SRC value for the bias term is not considered during the ranking

	Hidden Neuron			
	1	3	2	4
PNSA-3	−1.80	−6.57	0.39	−1.43
RSHM	4.03	6.15	1.50	1.01
V4P-5	9.45	2.15	3.24	0.60
S4PC-12	3.36	2.73	1.99	0.56
MW-16	3.94	8.42	1.94	0.76
WTPT-2	1.71	2.61	1.17	−0.13
DPHS	0.66	0.44	0.33	1.65
SCV	0.52	0.33	0.13	0.01

Table 9.8. Summary of the linear regression model developed for the BBB dataset

	Estimate	Std. Error	t
(Intercept)	0.53	0.07	7.28
WNSA-3	0.04	0.01	6.24
V4P-5	0.24	0.03	7.13
NDB	-0.13	0.03	-5.05
PNHS-3	0.03	0.00	6.93

Table 9.9. Summary of the PLS analysis based on the linear regression model developed for the BBB dataset

Components	Error SS	R^2	PRESS	Q^2
1	22.40	0.62	23.80	0.59
2	13.90	0.76	15.40	0.74
3	13.00	0.78	14.80	0.75
4	13.00	0.78	14.70	0.75

Table 9.10. The X-weights for the PLS components from the PLS analysis summarized in Table 9.9

	Component			
Descriptor	1	2	3	4
WNSA-3	0.54	-0.13	0.79	0.28
V4P-5	-0.09	0.97	0.17	0.12
NDB	-0.57	-0.08	0.58	-0.58
PNHS-3	0.62	0.17	-0.12	-0.76

Table 9.11. The effective weight matrix for the 4–4–1 CNN model developed for the BBB dataset. The columns are ordered by the squared contribution values for the hidden neurons, shown in the last row

	Hidden Neuron			
	1	2	4	3
WNSA-3	52.41	29.30	−19.64	2.26
V4P-5	37.65	22.14	−3.51	−13.99
NDB	−10.50	−16.85	−5.02	22.16
PNHS-3	11.46	6.59	−2.72	8.36
SCV	0.74	0.16	0.08	0.03

Table 9.12. Summary of the linear regression model developed for the skin permeability dataset

	Estimate	Std. Error	t
(Intercept)	-5.47	0.24	-22.94
SA	0.00	0.00	6.92
FPSA-2	-2.38	0.17	-14.12
NN	-0.28	0.05	-6.05
MOLC-9	0.50	0.07	7.19
PPHS	0.009	0.0007	13.47
WPHS-3	-0.02	0.00	-5.41
RNHS	0.05	0.00	7.48

Table 9.13. Summary of the PLS analysis based on the linear regression model developed for the skin permeability dataset

Components	Error SS	R^2	PRESS	Q^2
1	68.16	0.44	73.40	0.40
2	41.24	0.66	44.79	0.64
3	24.22	0.80	28.64	0.77
4	19.79	0.84	23.21	0.81
5	19.40	0.84	22.21	0.82
6	19.39	0.84	22.23	0.82
7	19.39	0.84	22.20	0.82

Table 9.14. The X-weights for the PLS components from the PLS analysis summarized in Table 9.13

Descriptor	1	2	3	4	5	6	7
SA	−0.08	0.52	0.20	−0.31	−0.29	−0.71	−0.07
FPSA-2	−0.52	0.14	−0.48	−0.38	−0.16	0.20	0.52
NN	−0.36	−0.03	0.07	0.45	−0.74	0.18	−0.27
MOLC-9	0.61	0.11	−0.32	0.36	−0.33	−0.16	0.50
PPHS	0.03	0.69	0.45	0.17	0.13	0.48	0.23
WPHS-3	0.09	0.48	−0.65	0.10	0.16	0.10	−0.55
RNHS	0.46	−0.04	0.07	−0.63	−0.42	0.41	−0.21

Table 9.15. The effective weight matrix for the 7–5–1 CNN model developed for the skin permeability dataset. The columns are ordered by the squared contribution values for the hidden neurons, shown in the last row

	Hidden Neuron				
	5	2	4	3	1
SA	−44.17	67.34	8.33	8.18	5.96
FPSA-2	−156.82	−10.72	20.85	−13.07	−92.47
NN	−97.81	2.22	−6.65	1.71	−12.70
MOLC-9	−28.85	17.79	15.40	−11.36	−1.20
PPHS	106.55	31.30	−16.76	−13.99	34.55
WPHS-3	−11.36	−14.31	−2.31	−10.01	54.16
RNHS	20.16	−5.89	−49.57	23.88	27.09
SCV	0.85	0.13	0.02	0.01	0.00

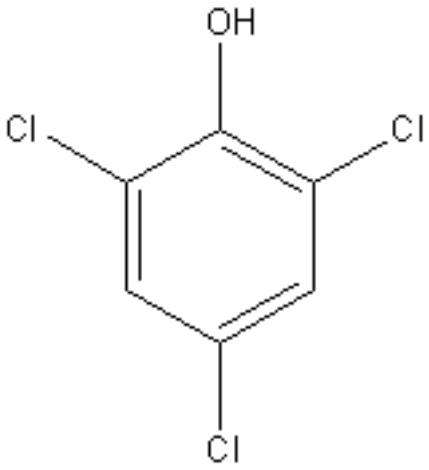
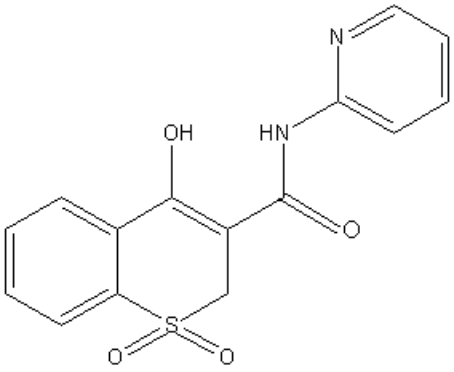
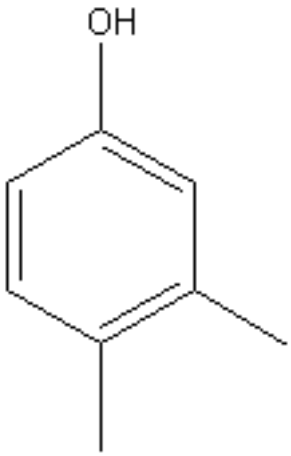
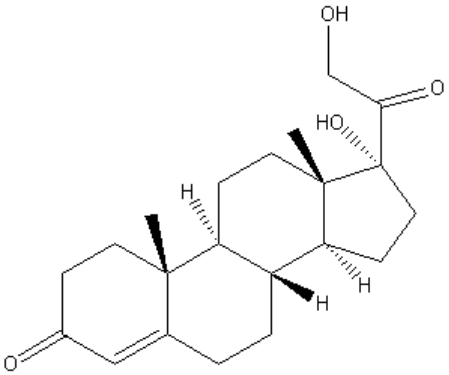
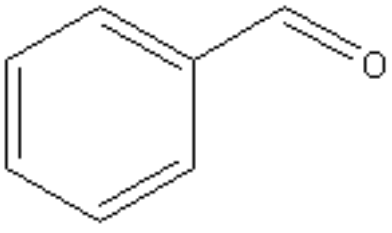
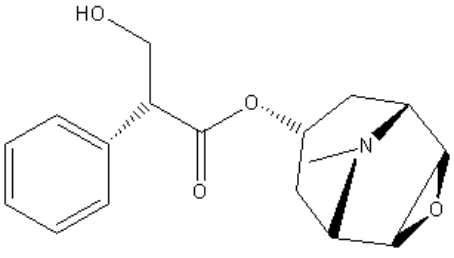
High Permeability	Low Permeability
	
3 (-1.23)	**121** (-1.38)
	
18 (-1.44)	**54** (-4.13)
	
43 (-0.85)	**129** (-4.30)

Fig. 9.1. A comparison of compounds exhibiting high and low skin permeability to illustrate the SPR encoded by component 1. The bold number is the serial number and the measured permeability coefficient is displayed in parentheses.

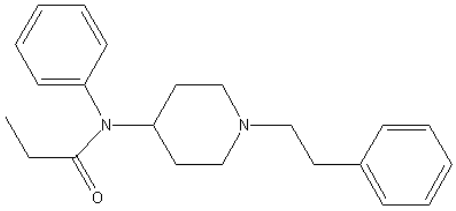
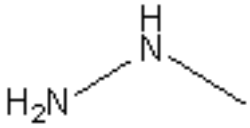
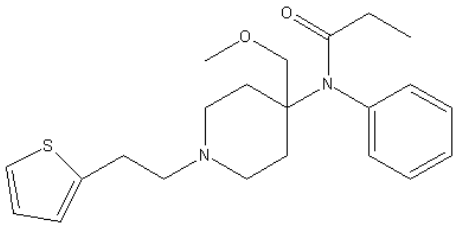
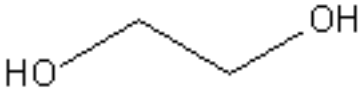
High Permeability	Low Permeability
	
82 (-2.25)	**101** (-3.75)
	
130 (-2.26)	**77** (-4.01)

Fig. 9.2. A comparison of compounds with high and low skin permeability, predicted by the second PLS component. The bold number is the serial number and the measured permeability coefficient is displayed in parentheses.

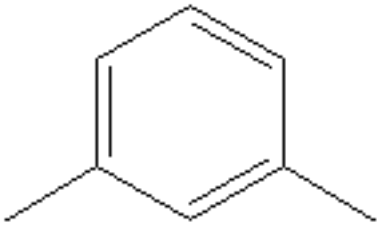
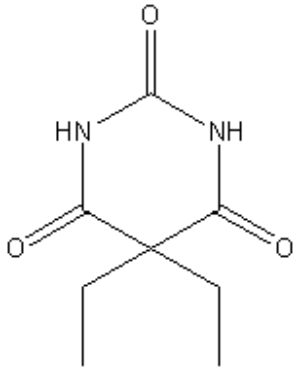
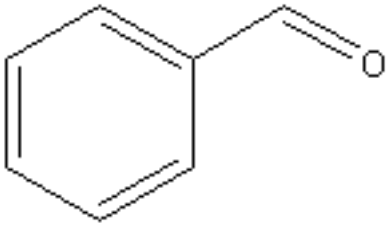
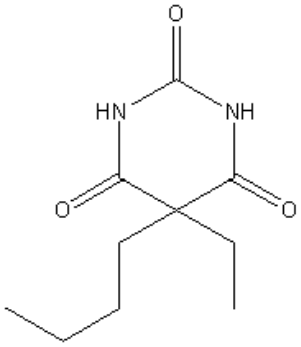
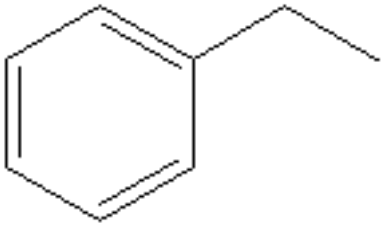
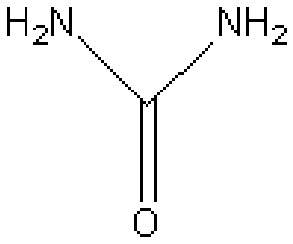
Moderate Permeability	Low Permeability
	
21 (-1.10)	**42** (-3.95)
	
43 (-0.85)	**46** (-3.72)
	
75 (-1.15)	**135** (-3.83)

Fig. 9.3. A comparison of structures illustrating compounds with high and low skin permeability, predicted by the 5^{th} hidden neuron. The bold number is the serial number and the number in brackets is the measured permeability coefficient

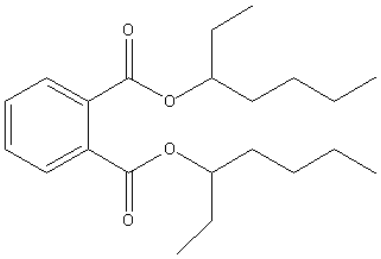
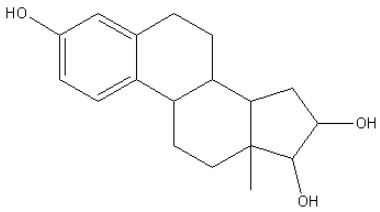
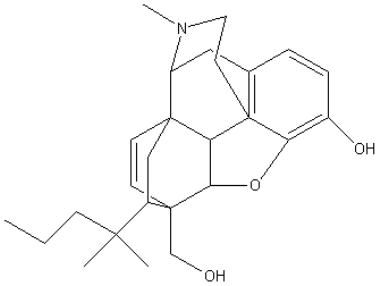
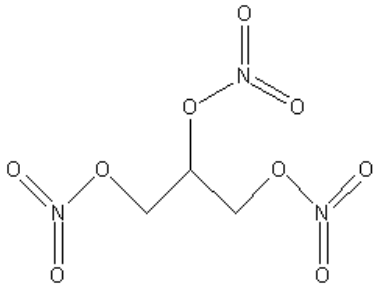
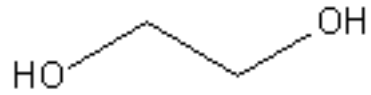
Moderate Permeability	Low Permeability
	
78 (-1.52)	**69** (-4.40)
	
81 (-2.44)	**72** (-4.02)
	
114 (-1.96)	**77** (-4.07)

Fig. 9.4. A comparison of structures illustrating compounds with moderate and low skin permeability predicted by the 2nd hidden neuron. The bold number is the serial number and the number in brackets is the measured permeability coefficient

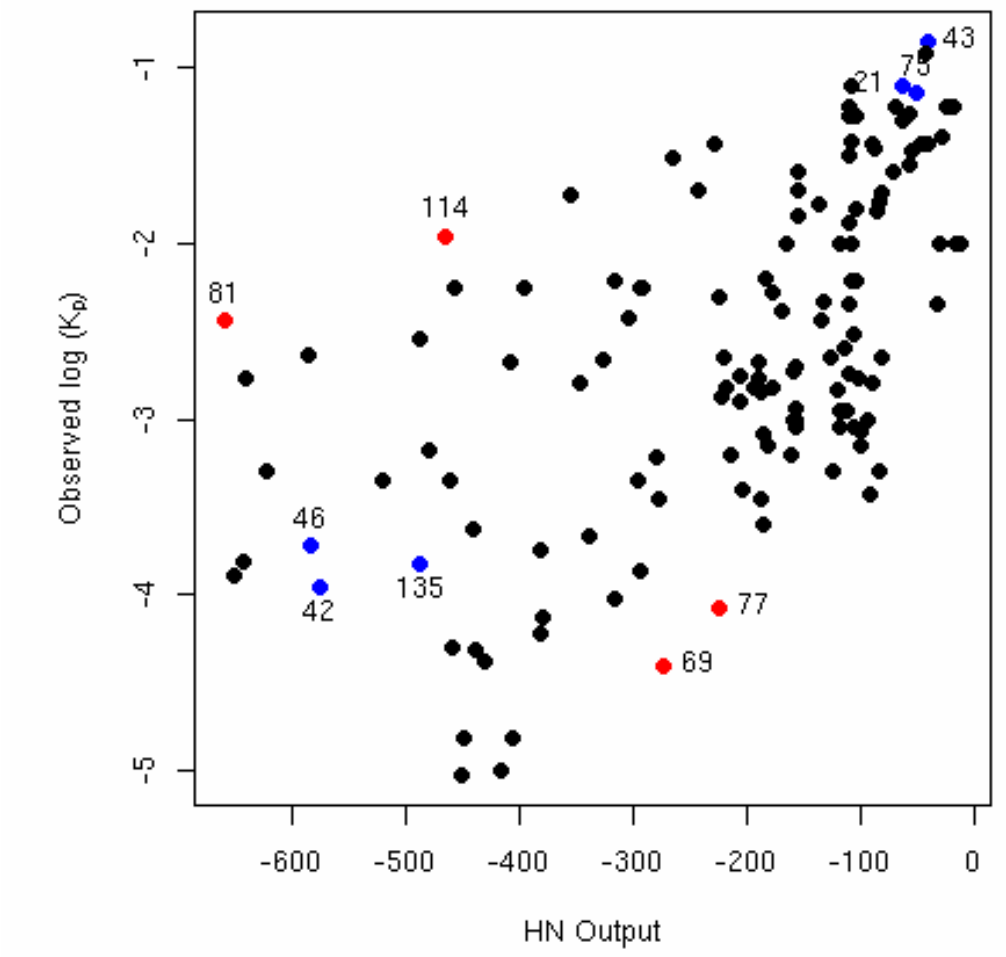


Fig. 9.5. The score plot for the 5^{th} hidden neuron. Points marked in red are examples of mispredicted molecules. Points colored blue are examples of well predicted molecules.

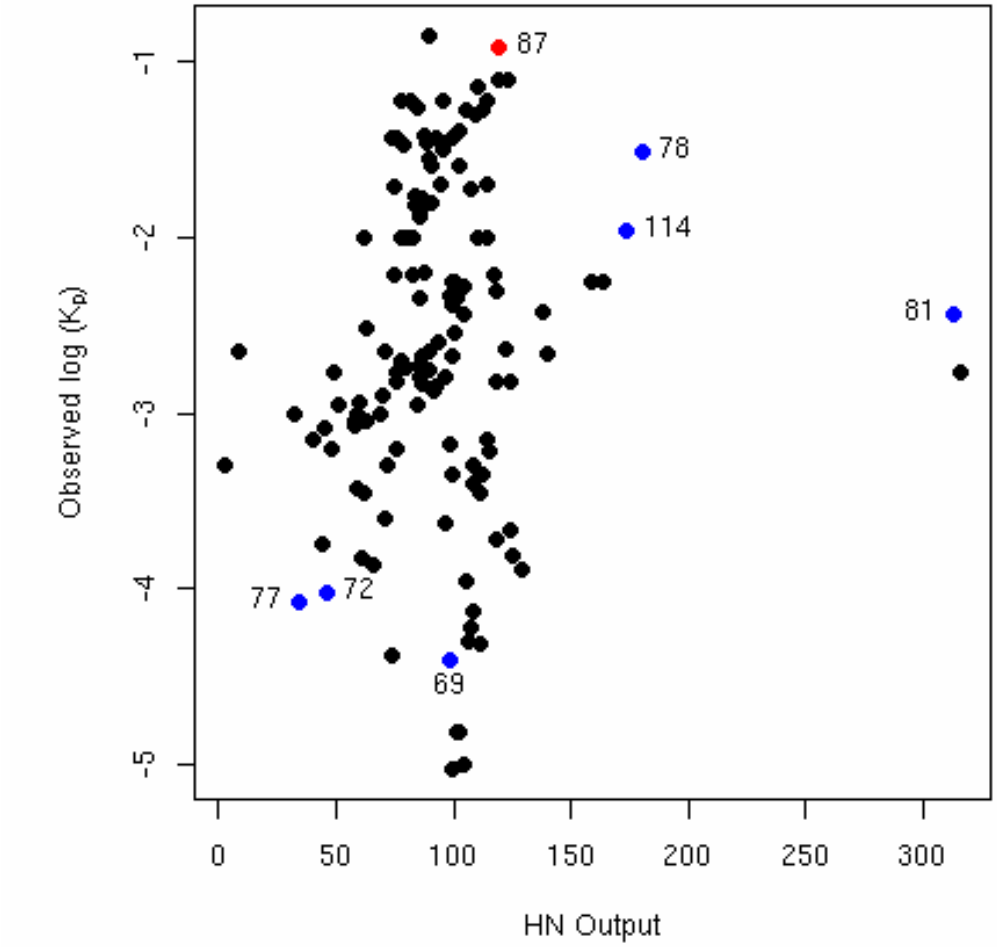


Fig. 9.6. The score plot for the 2nd hidden neuron.Points marked in red are examples of mispredicted molecules. Points colored blue are examples of well predicted molecules.

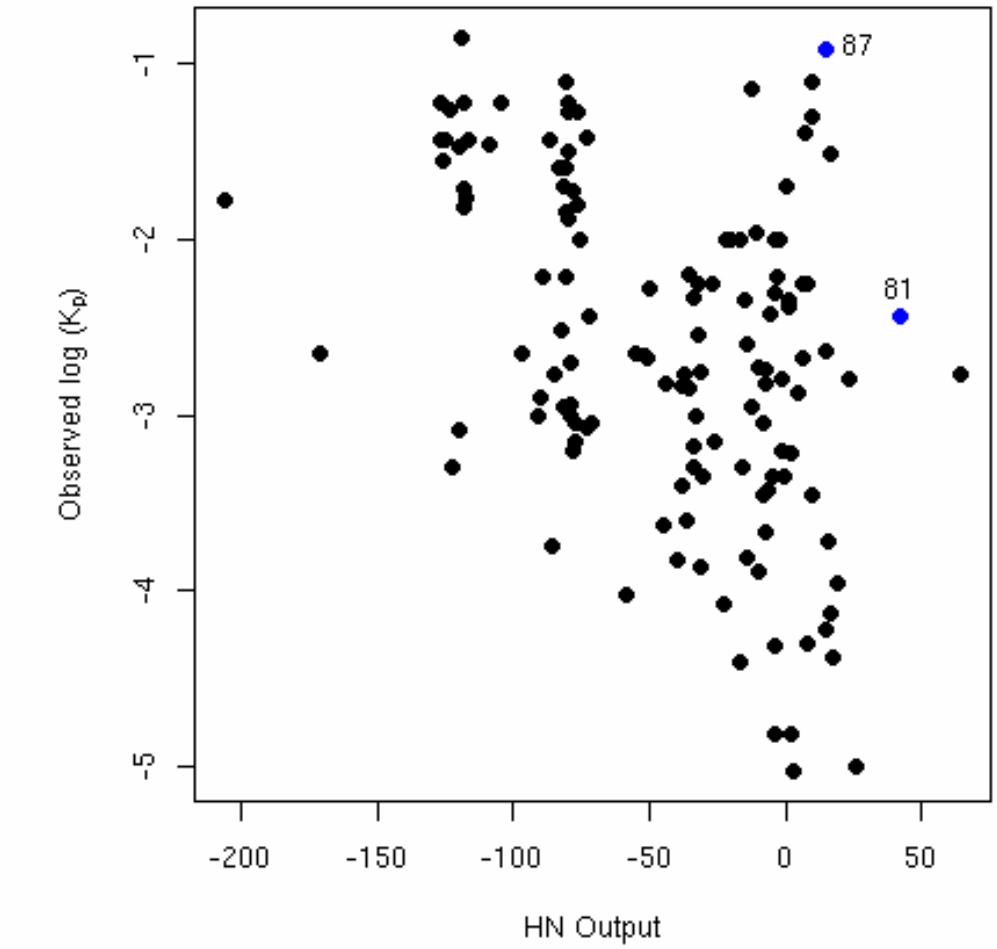


Fig. 9.7. The score plot for the 4^{th} hidden neuron. Points marked in red are examples of mispredicted molecules.

References

[1] Stanton, D. On the Physical Interpretation of QSAR Models. *J. Chem. Inf. Comput Sci.* **2003,** *43,* 1423–1433.

[2] Guha, R.; Jurs, P. C. The Development of Linear, Ensemble and Non-linear Models for the Prediction and Interpretation of the Biological Activity of a Set of PDGFR Inhibitors. *J. Chem. Inf. Comput. Sci.* **2004,** *44,* 2179–2189.

[3] Guha, R.; Jurs, P. C. The Development of QSAR Models to Predict and Interpret the Biological Activity of Artemisinin Analogues. *J. Chem. Inf. Comput. Sci.* **2004,** *44,* 1440–1449.

[4] Breiman, L.; Friedman, J.; Olshen, R.; Stone, C. *Classification and Regresion Trees;* CRC Press: Boca Raton, FL, 1984.

[5] Breiman, L. Random Forests. *Machine Learning* **2001,** *45,* 5–32.

[6] Castro, J.; Mantas, C.; Benitez, J. Interpretation of Artificial Neural Networks by Means of Fuzzy Rules. *IEEE Trans. Neural Networks* **2002,** *13,* 101–116.

[7] Limin, F. Rule Generation from Neural Networks. *IEEE Trans. Systems, Man and Cybernetics* **1994,** *24,* 1114–1124.

[8] Bologna, G. Rule Extraction from Linear Combinations of DIMLP Neural Networks. In *Proceedings of the Sixth Brazilian Symposium on Neural Networks*; IEEE: New York, NY, 2000.

[9] Yao, S.; Wei, C.; He, Z. Evolving Fuzzy Neural Networks for Extracting Rules. In *Fuzzy Systems, Proceedings of the Fifth IEEE International Conference on*, Vol. 1; IEEE: New York, NY, 1996.

[10] Tickle, A.; Golea, M.; Hayward, R.; Diederich, J. The Truth is in There: Current Issues in Extracting Rules from Trained Feedforward Artificial Neural Networks. In *Neural Networks, International Conference on*, Vol. 4; IEEE: New York, NY, 1997.

[11] Sato, M.; Tsukimoto, H. Rule Extraction from Neural Networks Via Decision Tree Induction. In *Neural Networks, International Joint Conference on*, Vol. 3; IEEE: New York, NY, 2001.

[12] Gupta, A.; Park, S.; Lam, S. Generalized Analytic Rule Extraction for Feedforward Neural Networks. *IEEE Transactions on Knowledge and Data Engineering* **1999,** *11,* 985–991.

[13] Chastrette, M.; Zakarya, D.; Peyraud, J. Structure-Musk Odor Relationships for Tetralins and Indans Using Neural Networks (On the Contribution of Descriptors to the Classification). *Eur. J. Med. Chem.* **1994,** *29,* 343–348.

[14] Hervas, C.; Silva, M.; Serrano, J. M.; Orejuela, E. Heuristic Extraction of Rules in Pruned Artificial Neural Network Models Used for Quantifying Highly Overlapping Chromatographic Peaks. *J. Chem. Inf. Comput. Sci.* **2004,** *44,* 1576–1584.

[15] Mak, B.; Blanning, R. An Empirical Measure of Element Contribution in Neural Networks. *IEEE Trans. Systems, Man and Cybernetics C* **1998,** *28,* 561–564.

[16] So, S.-S.; Karplus, M. Evolutionary Optimization in Quantitative Structure-Activity Relationship: An Application of Genetic Neural Networks. *J. Med. Chem.* **1996,** *39,* 1521–1530.

[17] Haykin, S. *Neural Networks;* Peason Education: Singapore, 2nd ed.; 2001.

[18] Hornik, K. Some New Results on Neural Network Approximation. *Neural Networks* **1993,** *6,* 1069–1072.

[19] Garson, D. Interpreting Neural Network Connection Strengths. *AI Expert* **1991,** 47–51.

[20] Yoon, Y.; Guimaraes, T.; Swales, G. Integrating Artificial Neural Networks with Rule-Based Expert Systems. *Decision Support Sys.* **1994,** *11,* 497–507.

[21] Metropolis, N.; Rosenbluth, A.; Rosenbluth, M.; Teller, A.; Teller, E. Equation of State Calculations by Fast Computing Machines. *J. Chem. Phys.* **1953,** *21,* 1087–1092.

[22] Sutter, J.; Dixon, S.; Jurs, P. Automated Descriptor Selection for Quantitative Structure-Activity Relationships Using Generalized Simulated Annealing. *J. Chem. Inf. Comput. Sci.* **1995,** *35,* 77–84.

[23] Goll, E.; Jurs, P. Prediction of the Normal Boiling Points of Organic Compounds from Molecular Structures with A Computational Neural Network Model. *J. Chem. Inf. Comput. Sci.* **1999,** *39,* 974–983.

[24] Stanton, D.; Mattioni, B. E.; Knittel, J.; Jurs, P. Development and Use of Hydrophobic Surface Area (HSA) Descriptors for Computer Assisted Quantitative Structure-Activity and Structure-Property Relationships. *J. Chem. Inf. Comput. Sci.* **2004,** *44,* 1010–1023.

[25] Patel, H.; Berge, W. t.; Cronin, M. Quantitative Structure-Activity Relationships (QSARs) for the Prediction of Skin Permeation of Exogenous Chemicals.. *Chemosphere* **2002,** *48,* 603–613.

[26] Mattioni, B. E. *The Development of Quantitative Structure-Activity Relationship Models for Physical Property and Biological Activity Prediction of Organic Compounds,* PhD thesis, Pennsylvania State University, 2003.

[27] Audus, K.; Chikhale, P.; Miller, D.; Thompson, S.; Borchardt, R. Brain Uptake of Drugs: The Influence of Chemical and Biological Factors. *Adv. Drug Res.* **1992,** *23,* 1–64.

[28] Gratten, J.; Abraham, M.; Bradbury, M.; Chadha, H. Molecular Factors Influencing Drug Transfer Across the Blood Brain Barrier. *J. Pharm. Pharmacol.* **1997,** *49,* 1211–1216.

[29] Ishibuchi, H.; Nii, M.; Tanaka, K. Fuzzy-Arithmetic-Based Approach for Extracting Positive and Negative Linguistic Rules from Trained Neural Networks. In *Fuzzy Systems, Proceedings of the IEEE International Conference on*, Vol. 3; IEEE: New York, NY, 1999.

[30] Chen, P.; Mills, J. Modeling of Neural Networks in Feedback Systems Using Describing Functions. In *Neural Networks, International Conference on*, Vol. 2; IEEE: New York, NY, 1997.

[31] Siu, K.-Y.; Roychowdhury, V.; Kailath, T. Rational Approximation, Harmonic Analysis and Neural Networks. In *Neural Networks, International Joint Conference on*, Vol. 1; IEEE: New York, NY, 1992.

[32] Fu, X.; Wang, L. Rule Extraction By Genetic Algorithms Based on a Simplified RBF Neural Network. In *Evolutionary Computation, Proceedings of the 2001 Congress on*, Vol. 2; IEEE: New York, NY, 2001.

[33] Stanton, D.; Jurs, P. Development and Use of Charged Partial Surface Area Structural Descriptors in Computer Assissted Quantitative Structure Property Relationship Studies. *Anal. Chem.* **1990,** *62,* 2323–2329.

[34] Balaban, A. Higly Discriminating Distance Based Topological Index. *Chem. Phys. Lett.* **1982,** *89,* 399–404.

[35] Kier, L.; Hall, L. *Molecular Connectivity in Chemistry & Drug Research;* Academic Press: New York, 1976.

[36] Kier, L.; Hall, L.; Murray, W. Molecular Connectivity I: Relationship to Local Anesthesia.. *J. Pharm. Sci.* **1975,** *64,*.

[37] Kier, L.; Hall, L. *Molecular Connectivity in Structure Activity Analysis.;* John Wiley & Sons: Hertfordshire, England, 1986.

[38] Kier, L.; Hall, L. Molecular Connectivity VII: Specific Treatment to Heteroatoms. *J. Pharm. Sci.* **1976,** *65,* 1806–1809.

[39] Randic, M. On Molecular Idenitification Numbers. *J. Chem. Inf. Comput. Sci.* **1984,** *24,* 164–175.

Chapter 10

Summary

The focus of this thesis has been the validation and interpretation of QSAR models. The preceding chapters cover examples of the application of interpretation method applied to a variety of models as well as the development of interpretation and validation methods.

Chapters 2 and 3 provide a brief introduction to the modeling techniques used in this work and the general methodology used to develop predictive QSAR models using the ADAPT software package. Though the descriptions have focused on quantitative models, the principles underlying the development of these types of models are also applicable to the development of classification models. As described in Chapter 3, the model building procedure involves a number of steps. Aspects such as feature selection have been studied extensively in the statistical literature, and developments in that field have yielded important insights into the methods by which good subsets of descriptors are selected. The model building process is also fundamentally a statistical problem and advances in the fields of data mining and pattern recognition have allowed the field of QSAR modeling to build models with increasing predictive ability and reliability.

Chapters 4 and 5 focus on two specific steps of the QSAR modeling process. Chapter 4 describes a method that was developed to create representative QSAR sets. One of the initial steps in building a QSAR model is to divide the dataset into a number of subsets, collectively termed QSAR sets. The training set is used to build the model, and the prediction set is used to test the final model for its predictive ability. In the case of a neural network an additional subset, namely the cross-validation set, is also created which is used to prevent overfitting. A number of methods exist to create these sets but one requirement is that each set be representative of the composition of the whole dataset. That is, features that are present in the overall dataset should also be present in these subsets, otherwise different phases of the model development process may be biased. This aspect of set creation is not addressed at all by the random selection method. The activity binning method does solve the problem to some extent but mainly concentrates on representing the range of activities in the whole dataset in the individual subsets. The

method described in Chapter 4 utilizes a self-organizing map (SOM), together with a set of holistic descriptors, to classify the dataset into two classes. The QSAR sets are then created such that the distribution of molecules in the two classes of the original dataset is maintained in each of the subsets. The assumption underlying this procedure is that models built with these QSAR sets will be able to capture the representative features of the dataset during training and will also exhibit better predictive ability compared to models built with QSAR sets in which the features of the dataset are not evenly distributed. The method was tested on a dataset of dihydrofolate reductase inhibitors which had been studied by Mattioni et al.[1] The best CNN model developed using this method had a simpler architecture and exhibited statistics similar to the previously reported model. An interesting observation regarding this model was that the statistics for the individual QSAR sets were consistent. That is, the RMSE and R^2 values were similar to each other for the training, cross-validation and prediction sets. This was not the case for the original model developed by Mattioni.[1] We believe that this indicates that the model developed using representative QSAR sets was better trained and had better generalizability than the reported model. The SOM method was also compared to the sphere exclusion method of generating QSAR sets using the diversity index as the metric. The results indicated that the latter method did not lead to better QSAR models than the SOM based method. Finally, we investigated a number of holistic descriptor combinations for use in the classification stage and employed a diversity metric to study the diversity of the QSAR sets created using the SOM and different descriptor types.

After a QSAR model has been built it must be validated. The goal of validation is to test whether a model has been overfit or is due to chance factors. In addition, the validation step provides a measure of the model's predictive ability. The traditional QSAR methodology uses scrambling tests and the prediction set to perform validation. However, these methods are not able to answer the question of a how a given model will perform when faced with data that it has not seen before. Though the prediction set statistics provide some indication of the model's generalizability, it is restricted, due to the fact that it has features in common with the dataset it was taken from. Furthermore, the prediction set statistics do not provide us with a measure of confidence for predictions made for molecules that were not present in the original dataset. The technique described in Chapter 5 allows us to understand whether a new molecule will be well predicted by a model or not and thus is more general than the statistics of the prediction set. In addition the method also provides a quantitative measure of the models predictive ability for a new molecule. That is, it provides a measure of confidence

in the models's prediction. Though confidence measures can be evaluated for specific types of models (confidence bounds for linear models, frequency scores for random forest models and so on), the method described in this work is quite general so that it can be applied to any type of regression model. We first investigated an approach which tried to correlate similarity to model quality, since one would expect that a new molecule that is similar to the training set should be predicted well. However, this did not yield any conclusive results. An alternative method was based on a classification approach. The residuals from a regression model were divided into two classes, good and bad, by choosing a cutoff value. Once the residuals were divided into two classes a classifier was trained, using these class assignments. We investigated both linear and nonlinear classifiers, and a nonlinear neural network classifier exhibited the best results. The input to the neural network was the descriptor set that was used to build the original model. Once the neural network was trained, it was then used to predict the class of the residual for a new compound. We tested the method on three datasets covering both biological activities and physical properties. The results indicated that the neural network classifier was able to correctly predict the class of the residual for a new molecule 80% to 90% of the time. The neural network classifier was also able to provide a probability of class membership, which can be viewed as a confidence measure. Plots of probability of membership versus residual were created to visualize the results of the technique for the training sets. Given that similarity should play an important role in the prediction of new compounds, we also attempted to include similarity measures in the classification model. However, the results indicated that their inclusion did not significantly improve results.

Chapters 6 and 7 focused on applications of the QSAR methodology and interpretation of linear models. The first study developed linear regression and neural network models to predict and interpret the anti-malarial activity of a set of artemisinin analogs. The dataset had been studied by Avery et al.,[2] who developed a set of models using the CoMFA technique. This method is a 3-D QSAR method and is dependent on accurate alignments. We attempted to build predictive models using the 2-D ADAPT methodology. The results indicated that the models we developed compared reasonably with those reported in the original work. The R^2 for the training and prediction sets were 0.68 and 0.77 respectively, compared to R^2 values ranging from 0.82 to 0.88 for the original models. However, the neural network model that was developed showed significantly better performance with R^2 values of 0.96, 0.94 and 0.88 for the training, cross-validation and prediction sets, respectively. The linear regression model was subsequently interpreted

using the PLS technique which was able to explain how each descriptor in the model correlated to the predicted activity. The conclusions from the interpretation corresponded well with the established mode of action of the artemisinin group of anti-malarials.

The second study (Chapter 7) focused on modeling the inhibitory activity of a set of PDGFR inhibitors. This class of compounds have been studied for their role in cell signal transduction pathways. The dataset considered in this work was obtained from Pandey et al.[3] who investigated the biological activity a set of 79 piperazinylquinazolines using a phosphorylation assay. The original dataset was studied in the presence and absence of human plasma. The latter data were modeled by Khadikar et al.,[4] who restricted themselves to the use of topological descriptors and linear regression models. The study described in Chapter 7 used the data from the assays carried out in the presence of human plasma and built linear regression and neural network models using the full suite of ADAPT descriptors. The best linear model exhibited a R^2 value of 0.84 and a RMSE of 0.24. The statistics for the neural network were significantly better than for the linear model. The R^2 for the training, cross-validation and prediction sets was 0.94, 0.90 and 0.61, respectively. The study also provided an interpretation of the structure-activity trends characterized by the linear model. The main conclusions that were drawn, namely, the presence of bulky hydrophobic groups and nitrogen centers increase inhibitory activity, matched closely to observations made by Pandey and other workers for this class of compounds. The study also developed a random forest model to determine descriptor importance. The ranking generated by the random forest model corresponded closely to the descriptors present in the best linear and nonlinear models, exhibiting the ability of the feature selection algorithms to select information rich descriptor subsets.

The last two chapters described approaches to the interpretation of neural network models. Chapter 8 described a method to provide a broad interpretation of a neural network model similar in manner to the descriptor importance measures for random forest models. The method described in this work was essentially a sensitivity analysis of the network and consisted of scrambling individual descriptors and making new predictions for the training set. The result of scrambling a descriptor was to increase the RMSE of the predictions. Furthermore, the more important a descriptor was to the model's predictive ability, the larger the difference between the RMSE of the original predictions and the RMSE for the predictions obtained after scrambling that descriptor. This procedure allowed us to rank the descriptors in the model in order of importance. By analogy with the descriptor importance plots that can be created for

random forest models, the method was also able to create importance plots which were used to visualize the relative importance of descriptors in a neural network model. This method was applied to neural network models built for three datasets using the ADAPT methodology. Linear models had been previously developed for these datasets. Each linear model was interpreted using the PLS technique. The results indicated that the descriptors that were highly ranked in the neural network models were very similar in nature (and in some cases identical) to the descriptors deemed the most important from the PLS analysis of the linear models. Assuming that both types of models captured similar structure-property trends in each dataset, these results indicate that the broad interpretation method correctly identifies the descriptors that play an important role in the neural network model's predictive ability.

Though this method provides some insight into the working of a neural network model, it does not help us to understand the role played by a specific descriptor in the model. In other words, how does the network model the relationship between a given descriptor and the predicted output? This problem was addressed in Chapter 9 which described a method to provide a detailed interpretation of a neural network model, similar to the type of interpretation that is possible for linear regression models using the PLS technique. The method used only the weights and biases of the trained network and did not require the training set to develop an interpretation. The core of the method involved the linearization of the network by defining effective weights. A second feature was to order the hidden neurons using the effective weights. By considering the hidden neurons as latent variables, the interpretation used the effective weights to provide a detailed breakdown of the roles played by each descriptor in the model's predictive ability. By evaluating the contribution of each hidden neuron to the output neuron, the method allowed the user to focus on the most important hidden neurons and the most important descriptors within them. The effective weights were also used, in conjunction with the training set data, to develop score plots, by analogy with the PLS interpretation technique. These plots allowed for the easy visualization of the behavior of each hidden neuron and resulted in a very detailed, compound-wise interpretation of the structure-property trends present in the dataset, as encoded by the neural network. The technique was tested on three datasets covering biological and physical properties. First, linear models were developed for each dataset using the ADAPT methodology and then interpreted using the PLS method. Next, the descriptor subsets from the linear models were used to build neural network models. The assumption was, that given the same datasets and descriptor subsets, both linear and nonlinear models would capture the same

structure-property trends. The results indicated that the neural network interpretation matched very closely (and in some case was identical to) the interpretation of the linear models. This indicates that the interpretation method was able to extract, in a detailed fashion, the structure-property trends encoded by the neural network models for the datasets studied.

In summary, this thesis has focused on specific steps in the QSAR model development process, some of which have not been considered in detail previously. The SOM method illustrates a way to create QSAR models utilizing as much information as is available to the modeler. The problem of model performance when faced with new compounds has not been studied in detail in the cheminformatics or QSAR literature and the method described in this work represents a generalized, quantitative approach to this problem. As described in this thesis, interpretation of QSAR models greatly improves their usability. For the case of linear models, the PLS approach allows us to understand, in detail, the various structure-property trends encoded in the model. The two studies presented in this work exemplify the ease with which linear models can be dissected using this technique. In the case of neural network QSAR models, interpretability has been lacking, resulting in their reputation as black box models. The two methods discussed in this work have have been successful in alleviating this problem. The broad interpretation method has been shown to be a useful method to quickly summarize the roles played by individual descriptors in a neural network model. For a more in-depth view of the relationship between input descriptors and network output, the detailed interpretation method has been shown to be able to provide a comprehensive view of the structure-property trends encoded in the model. Together with score plots, this method allows for a detailed, compound specific analysis of the model. The method thus places neural network models on par with linear regression models in terms of interpretability. As a result of this method, 2-D QSAR studies involving both linear regression and neural network models are expected to be more comprehensive, leading to better understanding of structure-property relationships. The studies presented in this thesis have been shown to improve the quality, reliability and interpretability of the QSAR modeling process.

References

[1] Mattioni, B. E.; Jurs, P. C. Prediction of Dihydrofolate Reductase Inhibition and Selectivity Using Computational Neural Networks and Linear Discriminant Analysis. *J. Mol. Graph. Model.* **2003,** *21,* 391–419.

[2] Avery, M.; Gao, F.; Wesley, C.; Mehrotra, S.; Milhous, W. Structure-Activity Relationships of the Antimalarial Agent Artemisinin. 1. Synthesis and Comparative Molecular Field Analysis of C-9 Analogs of Artemisinin and 10-Deoxoartemisinin. *J. Med. Chem.* **1993,** *36,* 4264–4275.

[3] Pandey, A.; Volkots, D. L.; Seroogy, J. M.; Rose, J. W.; Yu, J.-C.; Lambing, J. L.; Hutchaleelaha, A.; Hollenbach, S. J.; Abe, K.; Giese, N. A.; Scarborough, R. M. Identification of Orally Active, Potent, and Selective 4-Piperazinylquinazolines as Antagonists of the Platelet-Derived Growth Factor Receptor Tyrosine Kinase Family. *J. Med. Chem.* **2002,** *45,* 3772–3793.

[4] Khadikar, P. V.; Shrivastava, A.; Agrawal, V. K.; Srivastava, S. Topological Designing of 4-Perazinylquinazolines as Anatagonists of PDGFR Tyrosine Kinase Family. *Bioorg. Med. Chem. Lett.* **2003,** *13,* 3009–3014.

Vita

Rajarshi Guha

Date & Place of Birth

1st February, 1977. Benghazi, Libya

Education

- M.Sc. – Indian Institute of Technology, Kharagpur, India, 2001
- B.Sc. (Hons) in Chemistry – Presidency College, Calcutta, India, 1995

Selected Distinctions

- Dalalian Fellowship, Pennsylvania State University, 2004
- ACS COMP Division CCG Excellence Award, 2004
- Braddock Graduate Fellowship, Pennsylvania State University, 2002
- Roberts Graduate Fellowship, Pennsylvania State University, 2001
- Institute Proficiency Prize, IIT Kharagpur, 2001

Selected Publications

- **Guha, R.**; Stanton, D.T; Jurs, P.C., "Interpreting Computational Neural Network QSAR Models: A Detailed Interpretation of the Weights and Biases", *J. Chem. Inf. Model*, **2005**, *in press*
- **Guha, R.**; Jurs, P.C., "Interpreting Computational Neural Network QSAR Models: A Measure of Descriptor Importance", *J. Chem. Inf. Model*, **2005**, *45*(3), 800–806
- **Guha, R.**; Jurs, P.C, "Determining the Validity of a QSAR Model – A Classification Approach", *J. Chem. Inf. Model*, **2005**, *45*(1), 65–73
- **Guha, R.**; Jurs, P.C., "The Development of QSAR Models To Predict and Interpret the Biological Activity of Artemisinin Analogues", *J. Chem. Inf. Comput. Sci.*, **2004**, *44*(4), 1440–1449
- **Guha, R.**; Serra, J.R.; Jurs, P.C., "Generation of QSAR Sets with a Self-Organizing Map", *J. Mol. Graph. Model.*, **2004**, *23*(1), 1–14