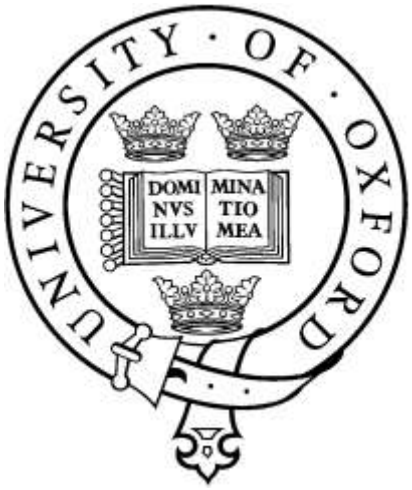


Physical & Theoretical Chemistry Laboratory
University of Oxford

A thesis submitted for the degree of Doctor of Philosophy

-Hilary Term 2011-

THE APPLICATIONS OF ARTIFICIAL INTELLIGENCE TECHNIQUES IN CARCINOGEN CHEMISTRY

Alexander C. Priest
St John's College
Oxford OX1 3JP
United Kingdom

Abstract

The Applications of Artificial Intelligence Techniques in Carcinogen Chemistry

Alexander Priest | St John's College | Doctor of Philosophy | Hilary Term 2011

Computer-based drug design is a vital area of pharmaceutical chemistry; Quantitative Structure-Activity Relationships *(QSARs)*, determined computationally from experimental observations, are crucial in identifying candidate drugs by early screening, saving time on synthesis and *in vivo* testing.

This thesis investigates the viability and the practicalities of using Mass Spectra-based pseudo-molecular descriptors, in comparison with other molecular descriptor systems, to predict the carcinogenicity, mutagenicity and the Cl^- transport inhibiting ability of a variety of molecules, and in the first case, of chemotherapeutic drugs particularly. It does so by identifying a number of QSARs which link the physical properties of chemicals with their concomitant activities in a reliable and mathematical manner.

First, this thesis confirms that carcinogenicity and mutagenicity are indeed predictable using a variety of Artificial Intelligence techniques, both supervised and unsupervised, information germane to pharmaceutical research groups interested in the preliminary screening of candidate anti-cancer drugs.

Secondly, it demonstrates that Mass Spectral intensities possess great descriptive fidelity and shows that reducing the burden of dimensionality is not only important, but imperative; selecting this smaller set of orthogonal descriptors is best achieved using Principal Component Analysis as opposed to the selection of a set of the most frequent fragments, or the use of every peak up to a number determined by the boundaries of supervised learning.

Thirdly, it introduces a novel system of backpropagation and demonstrates that it is more efficient than its principal competitor at monitoring a series of connection weights when applied to this area of research, which requires complex relationships.

Finally, it promulgates some preliminary conclusions about which AI techniques are applicable to certain problem-scenarios, how these techniques might be applied, and the likelihood that that application will result in the identification of series of reliable QSARs.

Table of Contents

Acknowledgments ..... *v*
List of Abbreviations ..... *vi*

Chapter 1 – Aims, Literature and Overview 1
1.1 Overview 2
1.2 Carcinogenesis, Teratogenesis, and Mutagenesis 4
1.3 Classes of Anti-Cancer Agent 5
1.4 Mass Spectrometry 8
1.5 Biological and Artificial Neural Networks 10
1.5.1 The XOR Problem 12
1.6 Support Vector Machines 13
1.7 Quantitative Structure-Activity Relationships 14
1.8 Choice of datasets 15
1.9 Thesis Objectives 17

Chapter 2 – An Introduction to the AI Techniques and the MD Systems 22
2.1 Introduction to Artificial Intelligence Techniques 23
2.1.1 Artificial Neural Networks 23
2.1.2 Support Vector Machines 34
2.1.3 Fuzzy Support Vector Machines 41
2.1.4 Genetic Algorithms 45
2.1.5 Voted Perceptrons/Boosting 48
2.1.6 Self-Organising Maps 50
2.2 Introduction to Molecular Descriptors 55
2.2.1 Principal Component Analysed Mass Spectra 56
2.2.2 Comparative Molecular Field Analysis 58
2.2.3 Comparative Molecular Similarity Indices Analysis 59
2.2.4 Hologram QSAR 60
2.2.5 EVA (EigenVAlue) Descriptors 60
2.2.6 CoDESSA 61
2.2.7 The Molecular Fragment Approach 62

Chapter 3 – The Transformation of Raw Data into Viable MDs 66
3.1 Introduction to Series of Data 67
3.1.1 Iris Data 67
3.1.2 Carcinogenicity (TD_{50}) in rats 67
3.1.3 Human Erythrocyte Cl- transport inhibitors 69
3.1.4 TOXNET Mutagen Data 69
3.2 The Creation of the Molecular Descriptors 71
3.2.1 Creation of the PCA Mass Spectra Descriptors 72
3.2.2 The Creation of the CoMFA Molecular Descriptors 78
3.2.3 The Creation of the CoMSIA Molecular Descriptors 78
3.2.4 The Creation of the HQSAR Molecular Descriptors 78
3.2.5 The Creation of the Eigenvalue Molecular Descriptor 79

Chapter 4 – Processing Methodology and Preliminary Results 83
4.1 Artificial Neural Networks 85
4.1.1 A Novel Backpropagation System 85
4.1.2 Transfer Functions 89
4.1.3 Modes of Computation 91
4.1.4 Data Management 91
4.2 Support Vector Machines 94
4.2.1 Comparison of commercial programs against home-made versions 94
4.2.2 User-defined parameters and architectures 95
4.2.3 Data Management 97
4.3 Fuzzy Support Vector Machines 100
4.4 Voted Perceptrons 104
4.5 Self-Organising Maps 106
4.6 Genetic Algorithms 112
4.7 Leave-three-out Cross-Validation 114
4.8 Java programs associated with this Thesis 115

Chapter 5 – Raw, Graphical, and Visual Results 118
5.1 Summary of Cross-Validation Values by Dataset 120
5.2 Discussion: the AI Techniques 122
5.3 Summary of Cross-Validation Values by Technique 125
5.4 Discussion: the Molecular Descriptors systems 128
5.5 Summary of Cross-Validation Values by Molecular Descriptor System 131
5.6 Detailed results of predicted Carcinogen TD_{50} bands 134
5.5 Detailed results of predicted Cl^- Transport Inhibitor Bands 137
5.6 Detailed results of predicted TOXNET Mutagen Bands 140
5.7 Discussion: misbanding and its implications 142
5.7 Results obtained through the Self-Organising Map 143
5.7.1 Self-Organising Map Clustering of the Fisher Iris Data 144
5.7.2 Self-Organising Map Clustering of the TD_{50} carcinogen Data 144
5.7.3 Self-Organising Map Clustering of Cl^- Transport Inhibitor Data 147
5.7.4 Self-Organising Map Clustering of TOXNET Mutagen Data 150

Chapter 6 – Discussion and Extensions 153
6.1 Dividing Activity Data into Fewer Bands 154
6.2 Is Hydrogen Bonding Information Important? 155
6.3. Practical Extension A: PCAMS Applicability 156
6.4 Practical Extension B: Different Classes of Drug 158

Chapter 7 – Conclusions 162

Acknowledgements

Thanks are due to a great number of supporters, technicians, family and friends, but specific, heartfelt gratitude is expressed to:

Dr Hugh Cartwright, my supervisor, for his patience, support and congenial open-door policy;

My research peers, and especially *Loren* & *Xandy,* for their camaraderie, help and lively high-spirits. I have never worked with a more interesting group of people: it was wonderful;

The North American Institute of Science and Technology;

The Carcinogenic Potency Project, Berkeley University, USA;

The National Toxicology Program, USA;

Dr Robert Wilkins, St Edmund Hall, Oxford;

The Oxford University Chemistry Department, for lab space, servers, and cheap lunches;

The Schools Competition Act Settlement Trust and *St John's College,* for funding;

My tutors, George, Fraser and *Richard,* for their encouragement and tutelage;

Christine Priest, my mother, who was my motivation for this research;

Stephen Priest, Marty, my grandmother, my siblings, their spouses and my nieces and nephew, for their kindness, love and understanding;

Rodney Priest, my own *Uncle Tungsten,* for lasting inspiration from a young age;

"Unless there are slaves to do the ugly, horrible, uninteresting work, culture and contemplation become almost impossible. Human slavery is wrong, insecure, and demoralizing. On mechanical slavery, on the slavery of the machine, the future of the world depends."

-Oscar Wilde, *The Soul of Man Under Socialism*

List of Abbreviations

SVM	Support Vector Machine
AI	Artificial Intelligence
ANN	Artificial Neural Networks
FSVM	Fuzzy Support Vector Machine
MS	Mass Spectrum/a
CML	Chemical Mark-up Language
m/z	Mass-charge ratio
GA	Genetic Algorithm
V	Volume of solution
R	Molar gas constant
T	Temperature of the reaction
Θ	Random number between 0 and 1 / Test Set
χ	Value of a node (i.e. the single input value in the first layer, and the calculated value in subsequent layers) (ANN)
O	Output of node, following transfer function (ANN)
T	Target output, for the purposes of training (ANN)
$\varepsilon = T - O$	Output error (ANN)
ℓ	Learning rate (ANN)
Ψ	Transfer function (ANN)
P	Correction factor / Penalty Constant (SVM)
b	Bias constant (SVM)
Λ	Bipolar sign function (SVM)
ξ	Slack variable (SVM)
m_i	Membership Value (FSVM)
${}^{i}_{a}\mathrm{v}_{x}$	Value of gene x in chromosome i of string [a] (GA)
${}^{i}\vec{\mathrm{V}}$	Vector representing chromosome i (GA)
${}^{i}\mathrm{v}_{max}$	Maximum value of gene on chromosome i (GA)
${}^{i}\mathrm{v}_{min}$	Minimum value of gene on chromosome i (GA)
${}^{i}_{a}\mathrm{I}$	Initial condition of chromosome i in string [a] (GA)
${}_{a}\psi$	Violations produced by string [a] (GA)
f(a)	Fitness of string [a] (GA)
$w_{k,i,j}$	The weight assigned to the connection between two nodes starting in layer k, node i, and layer k+1, node j (GA)
δ	Shortest distance in Euclidian n-dimensional space (SOM)
χ and w	Points in said space (SOM)
$\alpha(t)$	Learning rate (SOM)
$\mathrm{h}(\mathrm{t},\delta_{\mathrm{bj}})$	Neighbourhood function (based on δ_{bj}) (SOM)

Chapter 1

Aims, Overview and

Literature Review

1.1 Overview

Computer-based drug design is a crucially-important area of pharmaceutical chemistry[1,2]. It is worth considering that bringing a drug to the market may take up to 18 years and can cost up to $2 billion[3]. Quantitative Structure-Activity Relationships (QSARs), determined computationally from experimental observations, are widely used to suggest candidate drugs for early screening, reducing both the time and money spent on synthesis and *in vivo* testing[4]. Computational approaches to the design of new drugs have become progressively more significant and sophisticated as computer power has increased, and all major pharmaceutical companies now make extensive use of these techniques[5].

Many properties of a chemical contribute to the decision over whether it is viable as a drug. Apart from the obvious question of efficacy, the harm that a drug itself presents must be sufficiently low that it does not outweigh its therapeutic value. The carcinogenicity, mutagenicity and teratogenicity of a drug comprise an important area in this regard, but modelling these kinds of complex and dynamic systems presents a significant barrier to the advancement of this technology. Chicu, Herrmann and Berking[6] used statistical methods to predict chemicals' toxicity in crustacea and fish, while Williams[7] predicted the carcinogenic effects of ATPase inhibitors using chemical structure. Later, Conolly *et al.*[8] modelled respiratory tract carcinogens in a similar way. This thesis builds on and extends their work.

Artificial Intelligence techniques have, with great success, provided insight into complex data analysis in many scientific and engineering spheres. This suggests that a fruitful area of research would be the application of techniques such as Artificial Neural Networks (ANNs)[9], which can potentially provide an accurate link between molecular descriptors and physical data, to this problem. There is also the potential to apply a newer addition to the canon of AI techniques, the Support Vector Machine, which has begun to show increasing importance in the field.

However, a great number of potential architectures, parameter settings and dimension-reduction techniques exist in Artificial Neural Networks and Support Vector Machines, and the optimal determination of these is an essential step in the use of these methods. Furthermore, it is necessary to represent physical phenomena in a theoretical frame and, consequently, identifying a representative series of molecular descriptors is also imperative. Matter[10] and Hopfinger *et al.*[11] proposed that this could be achieved by creating a series of molecular descriptors which represent the energy-minimised three-dimensional drug, and Tanabe *et al.*[12] subsequently predicted carcinogenicities using three-dimensional geometrical molecular descriptors as a basis for supervised learning.

We describe here an extension of their work using mass spectral data, which is surprisingly effective considering its relative conceptual simplicity.

1.2 Carcinogenesis, Teratogenesis, and Mutagenesis

Whereas carcinogenicity is a well known property of molecules, teratogenicity (which we will also consider) is a little more obscure. Teratogenesis (literally "monster-birth") is the formation of congenital birth defects, such as cleft palettes or limb deformities. These malformations arise from foetal abnormalities, often caused by an adverse environment, such as interaction with teratogens. Similarly, mutagens (literally "the origin of change") are chemicals which alter the genetic make-up (DNA) of an organism. Whilst it is the case that a chemical often may belong to more than one of these distinct types, the confusion between them in the literature is unhelpful.

This thesis provides a holistic, quantitative approach to the applications of AI techniques to carcinogen chemistry, while incorporating novel ideas and techniques, tailored to improve and simplify them. Implicit in this research is the need to compare techniques, descriptors, and parameters, and to create a map which allows the user to navigate through this complex field.

In order to compare and contrast ANNs and SVMs when applied in this field, and set their abilities in a proper context, it will be necessary to compare them to a variety of other, more established AI techniques, as well as to each other. These comparators include Genetic Algorithms, Voted Perceptrons and Self-Organising Maps. In each case it will be necessary to optimise every aspect of the technique, its parameters, and its architecture so that it represents the

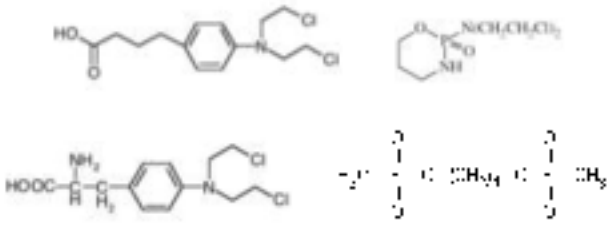
paradigm example when applied to this area; this should ensure a level playing field.

1.3 Classes of Anti-Cancer Agent

The general term *cancer* is applied to many phenomena in the human body, but there exist a wealth of extremely different chemotherapeutic and antineoplastic agents:

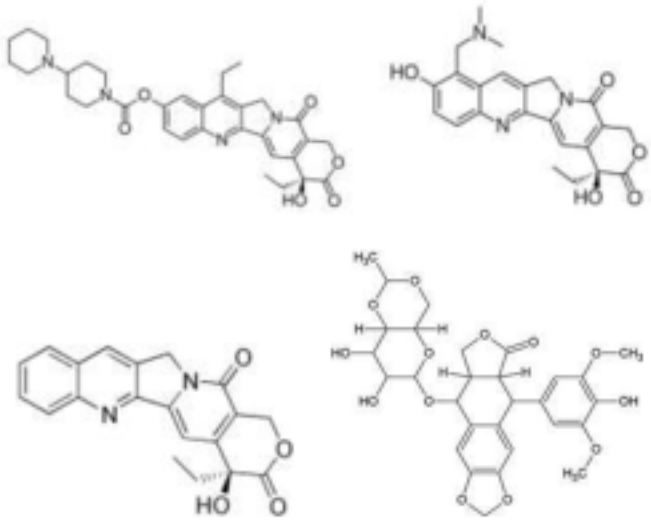
Alkylating Agents: Cancerous cells divide at a rate far greater than other cells in the body. Alkylating agents add an alkyl group to DNA, cross-linking guanine bases, preventing DNA replication and, therefore, stop cell division. Examples include[13] Cyclophosphamide, Ifosfamide, Chlorambucil, Melphalan, Busulfan, Lomustine, Carmustinegiven, Estramustine, Treosulfan, Thiotepa, and Mitobronitol.

Fig 1.1 - Chlorambucil, Cyclophosphamide, Melphalan, Busulfan



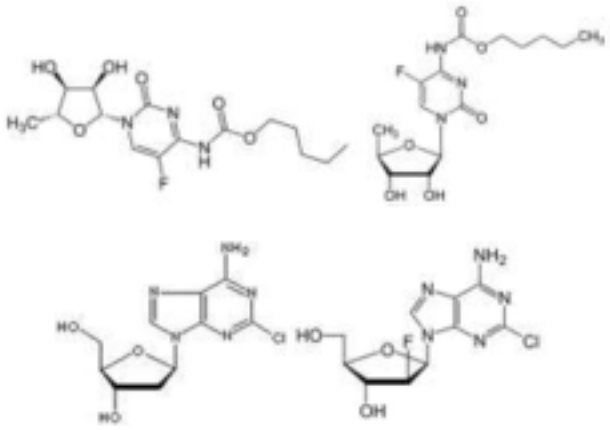
Topoisomerase Inhibitors: Topoisomerase inhibitors prevent DNA replication by interfering with the DNA ligation step in the cell division cycle. Examples include Irinotecan, Topotecab, Camptothecin, Lamellarind, and Etoposide.

Fig 1.2 - Irinotecan, Topotecab, Camptothecin, Etoposide.



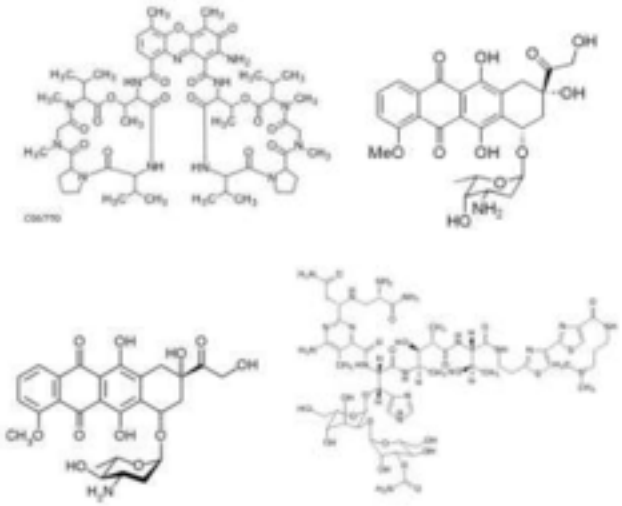
Antimetabolites: Antimetabolites masquerade as nucleobases and so prevent the latter's incorporation into new DNA molecules. They also interfere with RNA transcription. Examples include Azacitidine, Capecitabine, Cladribine, Clofarabine, Cytarabine, Fludarabine Phosphate, Fluorouracil, Gemcitabine, Mercaptopurine, Methotrexate, Nelarabine, Pemetrexed, Raltitrexed, Tegafur with Uracil, and Tioguanine.

Fig 1.3 - Azacitidine, Capecitabine, Cladribine, Clofarabine



Antineoplastics: Anti-tumour antibiotics such as members of the anthracycline family inhibit Topoisomerase II, which prevents DNA from being unwound. These include Dactinomycin, Doxorubicin, Epirubicin, and Bleomycin.

Fig 1.4 - Dactinomycin, Doxorubicin, Epirubicin, and Bleomycin



This is not an exhaustive list – it omits certain classes such as photosensitisers and kinase inhibitors – but is comprehensive enough for procedural decisions to be made. It seems that the classes of drug fall into two broad types, those which interfere with DNA itself (Alkylating Agents and Antimetabolites) and those which interfere with the mechanism of transcription (Topoisomerase Inhibitors and Cytotoxic Agents).

It is also worth noting the sheer variety of shapes and sizes which chemotherapeutics possess: from the small Alkylating Agents [Fig 1.1] to the complex Antineoplastics [Fig 1.4]. Doxorubicin and Epirubicin are extremely similar [Fig 1.4]; Cyclophosphamide and Dactinomycin are extremely different. This variation will require careful modelling, which takes into account the nuances of the structures, as well as the applications for which they are used.

1.4 Mass Spectrometry

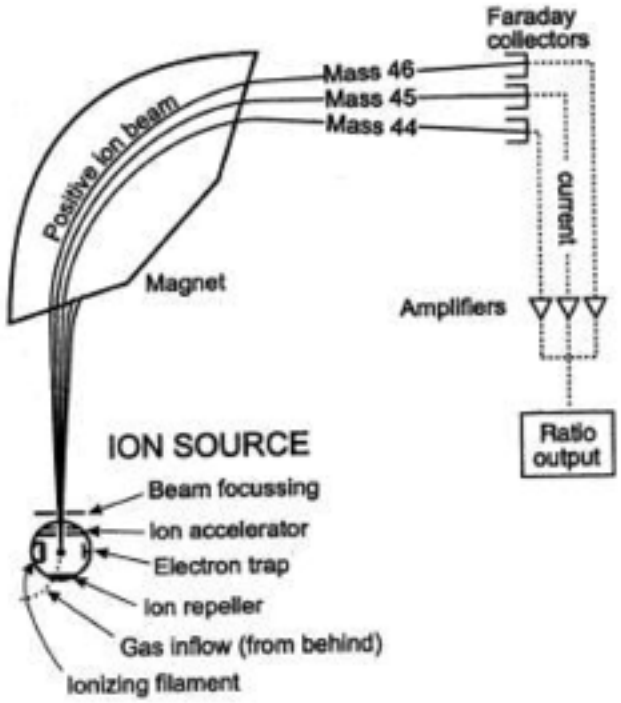
During the course of this thesis, I will be making use of mass spectral data as a novel system of molecular description. In order to understand the power of the technique and the implications of the results it is necessary to understand what mass spectral data is and where it comes from.

Simply, Mass Spectrometry is a physical, empirical process for calculating the mass/charge ratios of charged particles, most usefully the elemental composition of a molecule, its bond energies and its stabilities. First a sample of

the chemical is vapourised, charged (ionised, usually, through the use of an electron beam), and accelerated across an Ion Accelerator. When these ions pass near a magnet, their course is bent by an amount reflective of both their mass and their charge. These fragments are then detected based on how far they are off-course [See Figure 1.5].

The intensities of the fragment peaks are then plotted against their mass/charge ratio. The existences and relative intensities of different peaks conceal a great deal of information about the molecule, its substituent parts, and its physical and biological properties. This information is likely to be critical in the establishment of a reliable QSAR for the prediction of carcinogenicity.

Figure 1.5 – A Mass Spectrometer[14]



1.5 Biological and Artificial Neural Networks

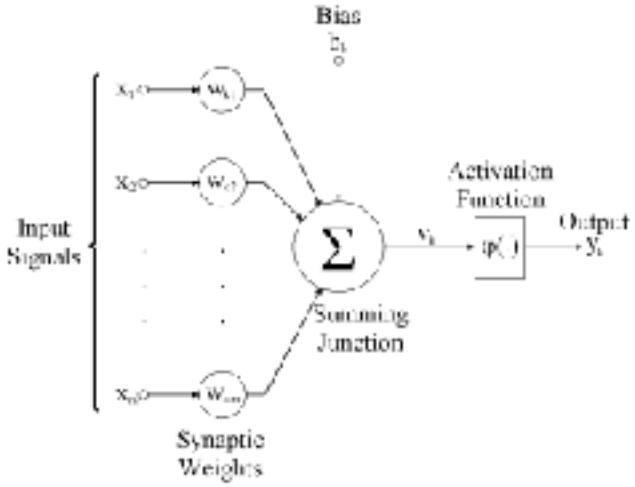
Computational methods, such as statistical analysis, graphical inference, data rationalisation or pattern recognition, have for some time been applied to many aspects of scientific research. A seminal development, made some time ago, but which came to fruition in the 1990s, is the cross-disciplinary field of Artificial Intelligence. The full extent of its applications to science has not even begun to be found, but continuous development is being made. Algebraically insoluble problems are now solved within a short space of time and to an unprecedented degree of accuracy. An important sub-section of this field is that of neural networking, in which programs use an analogue of the learning mechanisms intrinsic in the human brain. Their power lies in their ability to discern and infer complex patterns within data and make accurate predictions about the physical world even when faced with new, dissimilar examples.

Artificial Neural Networks are, in fact, merely an extension of many, more generic, techniques which have been used for many decades, the most obvious precursor being statistical pattern recognition. These statistical foundations are emphasised in a review by Ripley[15], and also in one by Cheng and Titterington[16].

In 1943 McCulloch and Pitts[17] described a biological neuron as a linear threshold unit with several inputs and an output of zero, for an inactive nerve

cell, or an output of one for a cell that had fired. A biological neuron fires when the summed inputs are larger than a certain threshold level.

Fig 1.6 – the Perceptron system developed by McCulloch and Pitts



This precept was used by Rosenblatt[18] when he developed the first artificial neural network in 1959. Based on its biological analogue, an artificial neuron known as the Perceptron, was invented. Its output, which was programmed to be -1 or +1, depended on the sum of various input vectors entering the cell. This system was then further developed and extended by Rosenblatt[19] and others in the 1960s. However, very little progress was made, as interest in the Perceptron declined in the 1970s. This decline was principally because the Perceptron was unable to represent simple, linearly inseparable functions (for example, those bound by the XOR problem[20]).

1.5.1 The XOR Problem

Single neurons are only capable of solving linearly separable problems (see Figure 1.7). This severely limits their usefulness. Simple problems such as the XOR logic function cannot be solved by a single perceptron. As can be seen in Figure 1.8, there is no straight line that can be drawn to separate the 0s from the 1s.

Fig 1.7 - A linearly separable problem.

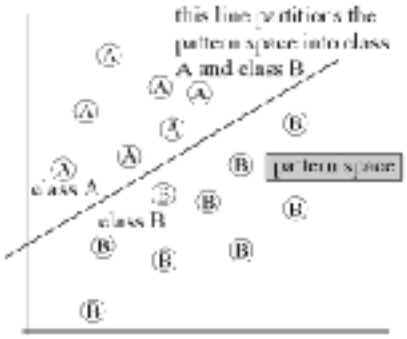


Fig 1.8 - The XOR problem.

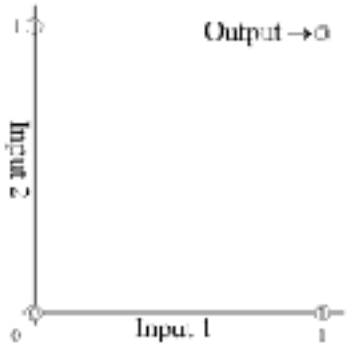
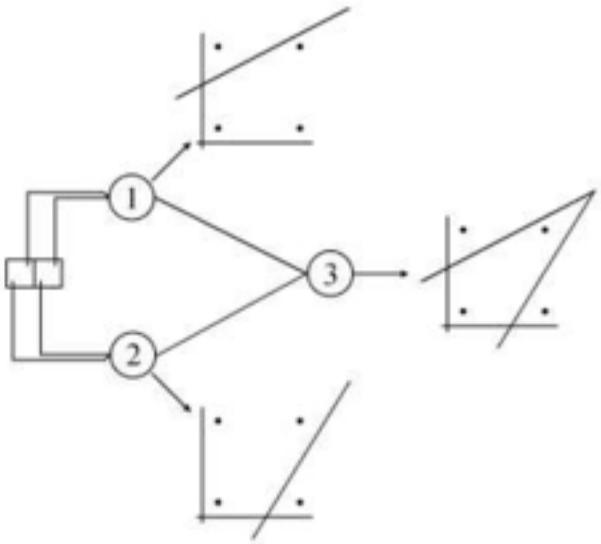


Figure 1.9 – The use of multiple perceptrons which when combined solve the XOR problem



Happily, in 1980s the XOR problem was solved, through the use of families of perceptrons (see Figure 1.9), and the ANN experienced a renaissance.

1.6 Support Vector Machines

Another new technique, developed at AT&T, by Vapnik and his colleagues in 1963, is the Support Vector Machine. Vapnik demonstrated the technique's performance on a number of problems of interest to the machine learning community, and its popularity has grown enormously. Highly mathematical in nature, SVMs rely on a variety of mathematical tricks and theorems to separate classes of data by mapping them into a higher-dimensional feature space[21]. In this way it has become child's-play to analyse the patterns inherent in even the most complex data.

In addition, Vapnik's circle have developed further improvements to his original technique, which allow the use of soft-margins (for non-separable data) (Vapnik's 1995 paper[22]) and Fuzzy Logic (which assigns importance labels to data). These have been used with great success in this field (for example by Wang, Liu, et al.[23]). Various tutorials and extensions to the technique have been published; an excellent example of the former by Bruges[24] was published in 1998.

1.7 Quantitative Structure-Activity Relationships

QSARs[25] represent a system by which the molecular structure and physical properties of a molecule can be used to predict likely activity in given situations, such as chemical and biological reactivities. First we must express the structure of a molecule and some physical, chemical or biological property it possesses numerically. This can then be converted into a mathematical association. The first modern QSAR was developed by Bonchev and Rouvray[26] in 1990: they used it to predict boiling points of alkanes.

Quantitative Property-Property Relationships (QPPRs), while not strictly QSARs, have existed for eighty or so years, for example, the famous Hammett Postulates[27] of 1937, which are experimentally-derived. Mathematical and computational QSARs were developed in the groundbreaking paradigm shift initiated by Leo and Hansch[28] in the 1960s. Leo and Hansch[29,30,31] are generally considered the principal researchers in this area: the QSAR Society declared Hansch the 'father of QSAR' as early as 1990. Hansch and Fujita[32], whose paper was the first to propose using Hammett parameters in biochemical QSARs, quickly found that their paper had become a 'Citation Classic'. Another important SAR, from 1997, is Lipinski's Rule of Five[33], a set of rules concerning the viability of a molecule as an oral drug for use in humans. Subsequently the pace of this type of research has become increasingly frenzied; in the past ten years alone Hansch and Leo's papers have been cited over 120,000 times.

1.8 Choice of datasets

The ambitious goals which I have set for this project make the selection of datasets from the profusion of data relating to the treatment of cancer available doubly important. There are three important criteria which must be considered when selecting sets:

i *Diversity* of types of data is important. It is important that the specific data series to which we apply these techniques represent a variety of types of dataset, so that more general conclusions can be drawn about the techniques, when applied in this broad area.

ii *Accessibility* is also important. It must be possible to access these data freely, whether downloaded online, sourced direct from research groups, or drawn from generic databases.

iii *Reliability* is imperative. The use of unreliable datasets would render this thesis useless. They need to be acquired from peer-reviewed or accredited sources, such as national bodies or known research groups.

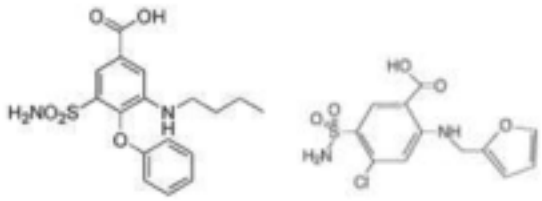
I have therefore selected, other than Fischer Iris testing data, four sources of data to be tested:

i Berkeley University Carcinogenic Potency Database, USA.

ii National Toxicology Program's 11th Report on Carcinogens, USA.

iii Inhibition of KCl transport in Human Erythrocytes, by bumetanide and furosemide analogues[34] (see Figure 1.10), sourced from the Wilkins Group at the Department of Physiology, University of Oxford.

Figure 1.10 - bumatanide and furosemide



iv National Library of Medicine, TOXNET Mutagens, USA

More information which justifies this choice of datasets is given in Chapter 3, but I believe that these four sources meet these important three criteria, whilst also pertaining directly to the goals of this research, which need clearly to be explained.

1.9 Thesis Objectives

The aims of this thesis are to address some of the computational challenges that a user in the field of chemotherapeutic drug design might face when attempting to apply Artificial Intelligence techniques to pattern recognition problems in that field. It will attempt to compare different techniques, concentrating on the application of Artificial Neural Networks and Support Vector Machines, as well some other comparators, Voted Perceptrons, Genetic Algorithms and Self-Organising Maps, to the identification of QSARs for a variety of datasets. It will also seek to test various molecular descriptors, to analyse which ones should be used, and in what order they should be tested.

As well as acting as an example of these techniques to individual problems in the field, this thesis will also attempt to act as a broader indicator of the types of problems that can be solved, and how the user should go about it, as well as a summary of the various options open.

- *Chapter 2* will further elaborate on the AI techniques used and their potential architectures, and will then explain the potential molecular descriptors which could be incorporated into the models to be investigated.

- *Chapter 3* will introduce the raw data, and explain how it will be manipulated into testing data and viable series of molecular descriptors.

- *Chapter 4* will explain exactly which architectures were used for each of the techniques and how these and user-defined parameters were selected.

- *Chapter 5* will display, in summary and in detail, results by cross-validation of each of the techniques, including within the techniques how each molecular descriptor system fared.

- *Chapter 6* will include both a discussion of the results presented in Chapter 5, and also extension experiments which explain anomalies and further test theories.

References

[1] Walters W.P., Stahl M.T., Murcko M.A., Virtual screening - an overview. *Drug Discovery Today* 3 (4) 160-178 (1998).

[2] Jorgensen W.L., The many roles of computation in drug discovery. *Science* 303 (5665) 1813-1818 (2004).

[3] Ivanciuc, O., Weka D., machine learning for predicting the phospholipidosis inducing potential *Current Topics in Medicinal Chemistry* 8 (12) 1691– 1709 (2008).

[4] Agatonovic-Kustrin S., Beresford R., Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. *Pharmacological Biomedical Analysis* 22 (5) 717-727 (2000).

[5] C. A. Ramsden, (ed.), Quantitative Drug Design, Vol. 4, of Comprehensive Medicinal Chemistry, eds. Hansch C., Sammes P.G. , and Taylor J.B., Pergamon, Oxford (1990).

[6] Chicu S., Hermann K., and Berking S., An Approach to Calculate the Toxicity of Simple Organic Molecules on the Basis of QSAR Analysis in Hydraactina echinata, Hydrozoa, Cnidaria. *Quantitative Structure-Activity Relatationships* 19 (2000).

[7] Williams W.R., Relative molecular similarity in selected chemical carcinogens and the nucleoside triphosphate chain. *Pharmacology & Toxicology* 92 (2) 57-63 (2003).

[8] Conolly R.B., Kimbell J.S., Janszen D. et al: Human respiratory tract cancer risks of inhaled formaldehyde: Dose-response predictions derived from biologically-motivated computational modeling of a combined rodent and human dataset. *Toxicological Sciences* 82 (1) 279-296 (2004).

[9] Werbos P., The Roots of Backpropagation, from Ordered Derivatives to Neural Networks and Political Forecasting. Wiley, a reprinting of his Harvard DPhil thesis of 1974 (1994).

[10] Hopfinger A.J., Wang S., Tokarski J.S., et al., Construction of 3D-QSAR models using the 4D-QSAR analysis formalism. *Journal of the American Chemical Society* 119 (43) 10509-10524 (1997).

[11] Matter H., Selecting optimally diverse compounds from structure databases: A validation study of two-dimensional and three-dimensional molecular descriptors, *Journal of Medicinal Chemistry* 40 (8) 1219-1229 (1997).

[12] Tanabe K., Ohmori N., Ono S., Suzuki T.: Neural network based QSARs of chemical carcinogens derived from chemical safety database CAESAR, *Pharmacy.& Pharmacology. 58: A32-A32 Supplement 1* (2006).

[13] British National Formulary: http://www.bnf.org, accessed on 14/03/2010.

[14] Revesz K., Landwehr, J.M., Keybl J., Photograph from Paper: Measurement of ^{13}C and ^{18}O Isotopic Ratios Of $CaCO_3$ using a Thermoquest Finnigan GasBench II Delta Plus XL Continuous Flow Isotope Ratio Mass Spectrometer with Application to Devils Hole Core DH-11 Calcite, *U.S. Geological Survey*, Open-File Report 01-257 (2002).

[15] Ripley, B. D., Neural networks and related methods for classification, *Journal of the Royal Statistical Society* 56 (3), 409-456 (1994).

[16] Cheng B. and Titterington D.M., Neural networks: a review from a statistical perspective. *Statistical Science* 9 (1), 2-54 (1994).

[17] McCulloch, W.S. and Pitts, W., A Logical Calculus of the Ideas Immanent in Nervous Activity, *Bulletin of Mathematical Biophysics* 5 115-133 (1943).

[18] Rosenblatt, F., The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, *Psychological Review* 65 386-408 (1959).

[19] Rosenblatt, F., Principles of Neurodynamics, Spartan Books, New York (1962).

[20] Minsky, M., and Papert, S., Perceptrons, MIT Press, Cambridge, MA (1969).

[21] Boser B.E., Guyon I.M., and Vapnik V.N., A training algorithm for optimal margin classifiers, Haussler D. (ed.), *5th Annual ACM Workshop on COLT*, pp 144-152, ACM Press, Pittsburgh, PA. (1992).

[22] Cortes C. and Vapnik V., Support-Vector Networks, *Machine Learning* 20 (1995).

[23] Wang J., Liu H., et al. Study on the structure-activity relationship of new anti-HIV nucleoside derivatives based on the support vector machine method, *QSAR & Combinatorial Science* 2 (26) 161-172 (2007).

[24] Burges C.J.C., A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery* 2 121-167 (1998).

[25] Kubinyi H., QSAR: Hansch Analysis and Related Approaches, VCH, Weinheim (1993).

[26] Bonchev D. and Rouvray D., Chemical Graph Theory: Introduction and Fundamentals, Gordon and Breach Science Publishers, (1990).

[27] Louis P. and Hammett J., The Effect of Structure upon the Reactions of Organic Compounds. Benzene Derivatives, *Journal of the American Chemical Society* 59 (1) 96-103 (1937).

[28] Leo A. and Hansch C., A method for the correlation of biological activity and chemical structure, *Journal of the American Chemical Society* 86, 5175 (1964).

[29] Hansch C. and Gao H., Comparative QSAR: Radical Reactions of Benzene Derivatives in Chemistry and Biology, *Chemical Reviews* 97, 2995-3059, (1997).

[30] Hansch C., Gao H. and Hoekman D., A Generalized Approach to Comparative QSAR, *Comparative QSAR*, Taylor and Francis, Washington, D.C., J. Devillers, (Ed.), 285-368, (1998).

[31] Leo A. and Hansch, C., Role of Hydrophobic Effects in Mechanistic QSAR, *Perspectives in Drug Discovery and Design* 17, 1-25 (1999).

[32] C. Fujita T., Iwasa J., & Hansch C., A new substituent constant, π ,derived from partition coefficients, *Journal of the American Chemical Society* 86, 5175-5180 (1964).

[33] Lipinski C., Lombardo F., Dominy B., Feeney B., Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings, *Advanced Drug Delivery Reviews* 46, 3-26 (2001).

[34] Culliford S., Ellory C., Lang H.J., Englert H., Staines H., and Wilkins R., Specificity of classical and putative Cl^- transport inhibitors on membrane transport pathways in human erythrocytes, *Cellular Physiology and Biochemistry* 13 181–188 (2003).

Chapter 2

An introduction to the Artificial Intelligence Techniques and the Molecular Descriptor Systems

2.1 Introduction to Artificial Intelligence Techniques

In the course of this thesis a legion of different AI techniques are used and compared. For those where novel systems have been introduced, or which form a large part of this research, good detail is included in this introduction; for more generic techniques, used only for comparison, less detail is necessary.

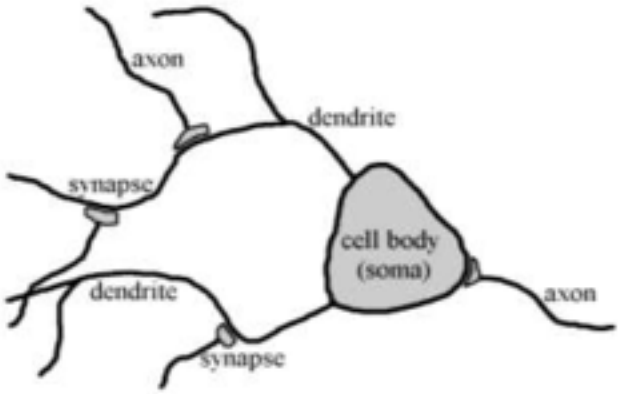
2.1.1 Artificial Neural Networks

Artificial Neural Networks (*ANN*), put simply, provide a computational analogue to the learning processes intrinsic to the mammalian brain. Biological parallels such as this are an important tool in the conception and understanding of Artificial Intelligence methods. Whilst the computational definition of evolution is at times complex, its biological analogue, the theory of evolution, is well understood by most scientists and educated laypeople, making an explanation of the former easier to digest by association. Indeed, until comparatively recently, most methods within the field of Artificial Intelligence could readily be understood by making such associations.

The Human Brain is principally composed of billions of tiny, electrically-stimulated cells known as neurons, which are themselves composed of a cell body (soma), a tree of dendrites, and an axon[35]. Simply, electrical and chemical impulses are received by the dendrites (i.e. the impulses act as an input vector), processed by the cell body and then given out via the axon (the output vector),

which is itself connected to one of the dendrites of another, or many other, neuron(s), so that a large network of neurons is formed. If a connection between neurons is used frequently, the connection becomes strengthened.

Fig 2.1 – the mammalian neural network



In an ANN cells called *nodes* are arranged in various ways so as to emulate the workings of the human brain. A node is the computational equivalent of a neuron, and has its own input and output signals. Networks undergo a regime of training with a particular dataset, after which a correctly-weighted neural network should have the ability to recognise patterns within sets of data, such that given a database of inputs, realistic outputs can be predicted.

The power of neural networks lies in their ability to predict non-linear and even quite complex patterns, given sufficient computational power and time. I shall only consider *linear feed-forward neural networks*. That is to say, networks in

which signals are passed from one *layer* of nodes to the next in one direction only. So, by way of example, one could use a 3-layer, 3-6-1, ANN, as below.

Fig 2.2 – A 3-6-1 ANN.

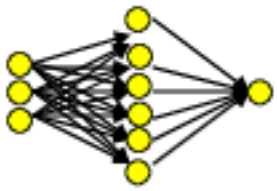
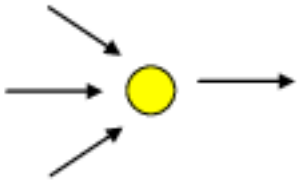


Fig 2.3 – A single hidden layer perceptron.



In Fig. 2.2, the first (i.e. the leftmost) layer is called the input layer, into which data are fed. The middle layer is called *the hidden layer*, and the final layer is called *the output layer*. Fig. 2.3 shows an isolated node from the hidden layer. It receives three pieces of information (in the form of numerical values), processes that information and then reaches a single numerical conclusion. The conclusion reached depends on three factors: the inputs, the *weight* assigned to each input and the *transfer function*.

The weight is a multiplicative factor that is applied to the value of an input entering the node. The weighted inputs are added to arrive at a total net input for each node. That is to say, assuming a linear transfer function (for now), that the output of the node,

$$O = \sum_{i=1}^{N} w_i x_i \qquad \text{when excluding a Transfer Function} \qquad [2.1]$$

where w_i is the weight applied to each input, x_i is its input value, and N is the number of input nodes. The purpose of a weight is to scale each input appropriately to achieve a desired output which gives accurate information. Weights are adjusted by an amount commensurate with the *output error* of the node during the process known as *training* or *supervised learning*.

Whereas we have seen that weights are readily-understandable multiplicative factors representing the 'believability' or trust we should put in the information supplied by a node, the concept of a transfer function is more complex. It does not vary between nodes, or even layers, but is assigned in the programming stage. It is a factor which affects the speed at which the network trains and the quality of that training; therefore, as we shall see, transfer functions are of the utmost importance for the viability of a network. The transfer function is applied by the node to the total input, in order to calculate its output

$$O = \psi\left(\sum_{i=1}^{N}\left(w_i x_i\right)\right) \qquad [2.2]$$

where ψ represents the transfer function acting upon the weighted inputs.

Several transfer functions have been mooted for use in ANNs[36]. Whilst Sigmoidal and Gaussian transfer functions tend to be ubiquitous, more flexibility is achieved by using a Bicentral or Conic system[37], because these more

complex mathematical functions are more suited to complex (e.g. biological) data. However, these tend, being far more complex, to create a greater processing burden. If the standard, *sigmoidal* model were adopted, once cumulative totals had been calculated at each node, the output value would be passed through a sigmoidal or *tanh* curve, with preset variables based on likely maxima and minima.

Training is the process by which, given a set of inputs, along with their optimal output, the connection weights are adjusted to ensure that the output of the network is correct to within a certain margin of error. For the ANN's first training loop, weights are generated randomly, which allows a random starting position, so that no bias exists as to the starting weights of the network. Whereas in the first iteration of the network the weights are generated randomly, in subsequent iterations of supervised learning the network weights are adjusted by backpropagation[38]. Alternatives include unsupervised learning (see SOMs), forward modelling, or Contrastive-Hebbian (Boltzmann) learning[39], which though efficient, and are not nearly so efficient when used in the development of complicated QSARS.

Backpropagation is a system of weight adjustment which propagates output error backwards through the network, adjusting weights in line with their responsibility for the overall error. The error is first calculated by taking the difference between the actual output and the desired output. This error is then

propagated backwards through the network, by adjusting the weight on the connection between every pair of nodes by a proportion of the actual error, related to the input node's value as a proportion of the total input. That is to say that a weight adjustment

$$\Delta w = x\ lO(1-O)\sum_{i=1}^{N} w_i \varepsilon_i \qquad [2.3]$$

where ε_i is the error at the node being processed and l is the learning rate.

In order to increase computational efficiency, that is to say the speed at which the network trains, it is extremely important to ensure that an optimised system of backpropagation is included. Several such systems exist. The most popular is as follows:

Let: χ = value of a node (i.e. the single input value in the first layer, and the calculated value in subsequent layers);

w = the weight assigned between two nodes;

O = output of the node, following application of the transfer function;

T = target output, for the purposes of training;

$\varepsilon = T - O$ = the output error;

ℓ = the learning rate; and

Ψ = a transfer function.

The principle of backpropagation dictates that $\frac{\delta\varepsilon}{\delta w}$ should be found.

Now, $$\varepsilon = \frac{1}{2}\sum(T - O)^2, \qquad [2.4]$$

but if ψ represents the transfer function,

then $$\varepsilon = \frac{1}{2}\sum(T - \psi(w\chi))^2, \qquad [2.5]$$

but since the summation is independent of the layer,

$$\frac{\delta\varepsilon}{\delta w\chi} = \frac{\delta}{\delta w\chi}\frac{1}{2}(T - O)^2$$

$$= -(T - O)\frac{\delta O}{\delta w\chi}$$

$$= -(T - O)\frac{\delta}{\delta w\chi}\psi(w\chi)$$

$$= -(T - O)\psi(w\chi)(1 - \psi(w\chi))$$

$$= -(T - O)(1 - O)(O)$$

Therefore, $$\Delta w_i = -\ell\frac{\delta\varepsilon}{\delta w} = \ell\frac{\delta\varepsilon}{\delta w\chi}\chi \qquad [2.6]$$

Having derived this and thought about its implications, it becomes clear that this system would work extremely well for simple data in layers adjacent to the input or output layers, but that were the data complicated, or intermediate hidden layers used, then the curve of error against weight would no longer be parabolic, as it is always in feed-forward ANNs, but rather a *polynodal* sinusoidal curve, which could result in suboptimal local maxima. Therefore, a novel system is proposed in Chapter 4.

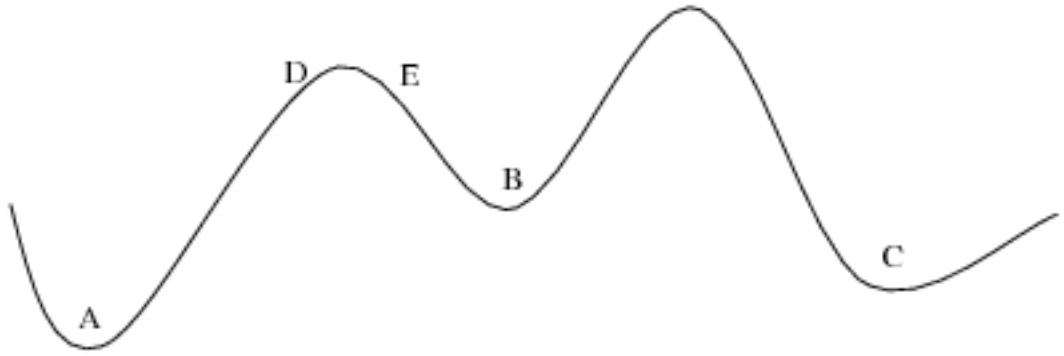
This adjustment is also subject to a multiplier, the *learning rate*, which sets the rate of change of the connection weights; this value is high at the beginning of the training process to allow large changes to the network weights, but becomes lower towards the end when errors are small.

Pareto equilibria, a grave problem for Artificial Intelligence, occurs when neural networks find a local minimum in the data, but the learning rate, *l*, is too small compared to the step required to escape from it: the Neural Network trains to a minimum, but not necessarily its lowest one.

This concept is best explained using the hill-walker analogy, shown in Figure 2.4. Owing to this problem, it is quite possible that when the network starts with random weights it will fall into a local minimum from which it would never escape. In order to lessen the likelihood of this, neural networks can run

parallel *threads* (contemporaneous instances of the same program), each with its own semi-random jump, which restart if no ground has been gained.

Figure 2.4– The hill-climbing algorithm.



Were a hillwalker to be at point B, in Fig 2.4, in this contoured structure (which represents a cross-section of the multi-dimensional surface under investigation), he would declare point B to be the deepest valley, because he cannot see valleys A or C. Similarly, were he 'dropped' at E, he would move towards B, using gradient descent. However, were he randomly dropped at D, he would find the true minimum at A.

Another, more chemical, example would be the local energy minimum which exists if the two methyl groups are adjacent (at 30°) in butane (see Fig. 2.5). A lower energy is possible if they are 180° apart, but in order to reach this conformer, the molecule is required to rotate past localised maxima (H and Me eclipsed).

Figure 2.5 – the Newman projection of butane



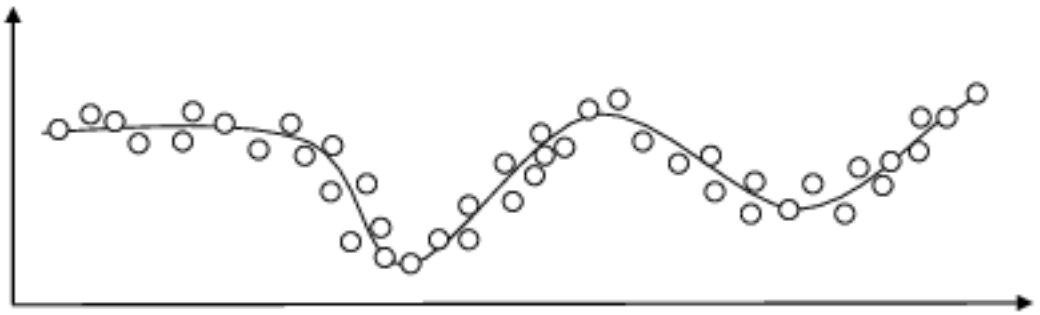
Artificial Neural Networks can be applied in three broad ways. First, in the classification of data, including pattern and sequential analysis, in novel item detection and in sequential decision making processes.

Fig 2.6– Data Classification



Secondly, for function approximation or regression analysis. And, thirdly in the processing of data, including filtering and compression.

Fig 2.7 – Function Approximation



These different types of task can relate to process identification and control (e.g. vehicle control), game-playing and decision making (e.g. boardgames), sequence recognition (be it gesture, speech, handwriting recognition), pattern recognition (e.g. radar systems, face identification, object recognition, for example a tank in the jungle), medical diagnosis, data mining, graphical visualisation or e-mail spam filtering.

This plethora of applications has ensured the continued popularity of ANNs in all branches of science, especially in Chemistry. A list of examples to show the diversity of neural networks in science would surely include studies into:

1. Plant ecosystems
2. Plant disease propagation[40]
3. Electricity load predictions[41]
4. River flow and flooding management
5. Chemicals in the environment and their effects on it
6. The environmental effects of various types of pollution[42]
7. Chemical QSARs
8. Groundwater Modelling[43]
9. Food engineering[44]
10. Economic Predictions[45]

Wherever data exist in sufficient quantities, Artificial Neural Networks will continue to be used to analyse this data and infer patterns, allowing the prediction of a whole host of physical, biological and chemical phenomena.

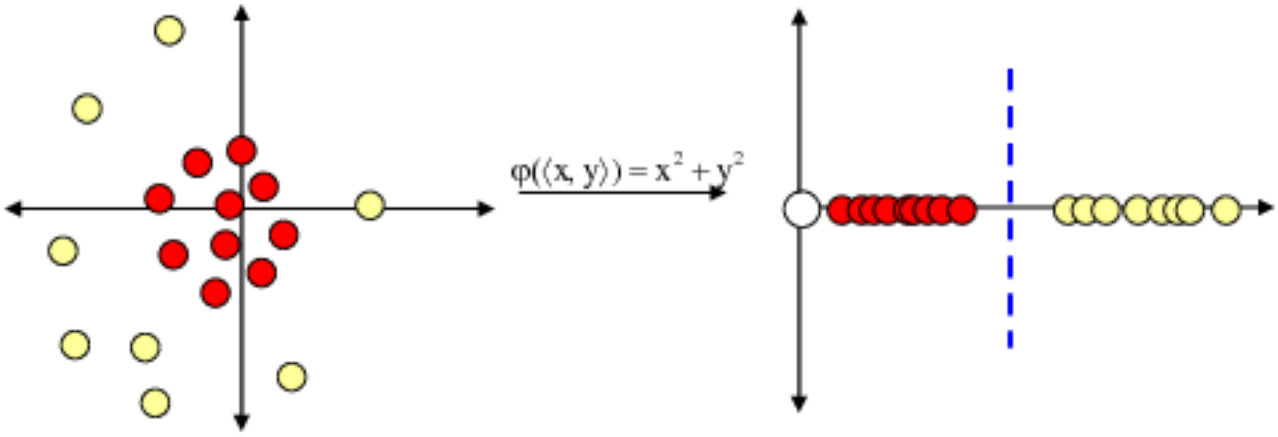
2.1.2 Support Vector Machines

Support Vector Machines (SVMs) are the latest in a host of methods of complex data analysis emerging from Computer Science and Mathematics, but how effective a tool are they for analysis of the kind of data generated by Chemistry? They currently have a high computation burden and are, as yet, underused in this field. On the one hand, studies have shown than in some cases SVMs are only 1% better discriminators than Voted Peceptrons or traditional Neural Networks[46]; on the other, as the increase in computational power continues exponentially, the run-time of more complex systems of Voted Perceptrons will decrease, and more complex models will be used; their efficiency may then rally to levels at which their importance will become undoubted.

This interface between Chemistry, Mathematics and Computer Science represents a key growth area for the fields of Chemistry and Biochemistry. Crucial leaps, such as the *Screensaver Lifesaver*[47] project at Oxford, have been made over the past twenty years. These allow Analytical Chemists to identify, in the computational sphere, reactions and chemicals most likely to develop attractive properties, before time and money are wasted in developing them *in vivo* and *in vitro*.

Support Vector Machines are a machine learning mechanism used for linear classification. They simultaneously minimise the classification error while maximising the margin between the delineation and the closest node (the margin). In order for this technique to work, the classes must be linearly separable. This can easily be achieved using the *Kernel Trick*, a method by which non-linearly separable data, having been passed through a kernel into a higher-dimensional feature space, become separable.

Fig 2.8 – The principle used in The Kernel Trick
(which maps non-linear data to a higher-dimensional feature space)
represented pictorially, showing a map to a lower-dimensional space.



Basically, Support Vector Machines are classifiers based on structural risk minimization whose purpose is to find the hyperplane that maximises the margin between classes. SVMs are often introduced through a two-class classification problem. Let us assume two classes, class one and class two, with class labels $y_1 = -1$ and $y_2 = +1$ respectively.

For a training set, Θ, of N data points,

$$\Theta = \{x_i, y_i\}_{i=1}^{N} \quad [2.7]$$

where each x_i belongs to one of the two classes ($y_i = \pm 1$). Training, then, finds an optimal hyperplane

$$w\phi(x) + b \quad [2.8]$$

which maximises the separation of two classes of training samples, where $\phi(x)$ is some nonlinear function which maps the input space into a higher dimensional feature space, w is a weight vector, and b is a bias constant.

x_i can then be assigned its concomitant class label and the classifier expressed as

$$g(x_i) = \Lambda\left(w\phi(x) + b\right) \quad [2.9]$$

where Λ is the bipolar sign function. If the classes are separable, a weight vector and a bias function can be found such that each example, x_i, satisfies either

$$w\phi(x_i)+b>+1 \text{ for } y_i=+1 \quad [2.10]$$

or $$w\phi(x_i)+b<-1 \text{ for } y_i=-1 \quad [2.11]$$

These equations can be simplified to

$$y_i\left(w\phi(x_i)+b\right)\geq+1 \text{ where } i=1,2,3...N \quad [2.12]$$

In this (separable) example, it is easy to discern the hyperplane which maximises the margin of separation between the classes. However, in a non-separable example, such a hyperplane does not exist. In this case, it is necessary to introduce a series of slack variables (ξ) as follows:

$$y_i\left(w\phi(x_i)+b\right)\geq 1-\xi_i \text{ where } i=1,2,3...N \text{ and } \xi_i\geq 0 \quad [2.13]$$

This modification allows more flexibility when dealing with non-separable cases, by allowing the system to build in degrees of dissent and error.

Therefore, in order to find the optimal hyperplane, it is necessary to solve the quadratic programming (QP) problem

$$\min L(w,\xi_i) = \frac{1}{2}w^2 + P\sum_{i=1}^{N}\xi_i \text{ subject to Equation 2.13} \qquad [2.14]$$

where *P* is a penalty constant which defines how much empirical errors are penalised in training. Rather than solving this QP optimisation in the primal space, a series of Lagrange multipliers can be substituted into Equation 2.13, giving

$$\max_{\alpha_i} Q(\alpha_i) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \phi(x_i)\phi(x_j) \qquad [2.15]$$

where $0 \geq \alpha_i \geq P$ and $\sum_{i=1}^{N}\alpha_i y_i = 0$

The Karush-Kuhn-Tucker conditions[48], which allow inequality constraints, generalise the method of Lagrange multipliers, which only allow equality constraints. These constrain the solution to Equation 2.15 into satisfying the following conditions:

$$\alpha_i\left(y_i\left(w\phi(x_i)+b\right)\geq 1-\xi_i\right)=0 \qquad \text{and} \qquad [2.16]$$

$$(P-\alpha_i)\xi_i = 0 \qquad [2.17]$$

Any points with $\alpha_i > 0$ are support vectors. There are two types of such points: those where α_i is between 0 and *P* lie on a margin. A point where α_i is equal to

P represents a misclassified datum.

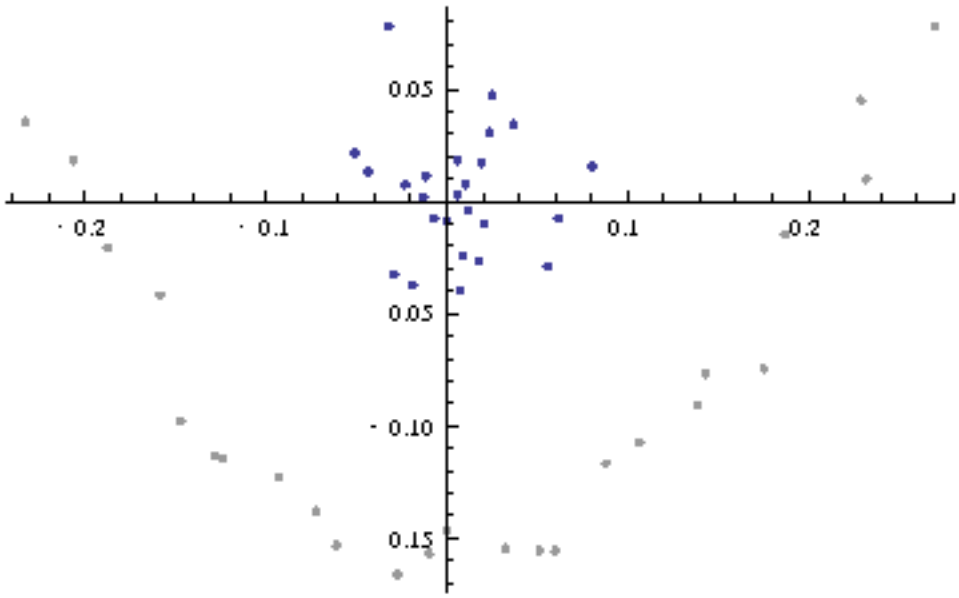
Practically, Support Vector Machines are reasonably simple to operate, once user-defined parameters have been put in place. Suppose we were to generate some noisy data and then analyse it.

```
len =20;
X =Join[RandomReal [NormalDistribution [-2,1],(len
/2,2}],RandomReal[ NormalDistribution[ 2,1],(len /2,2}]];
y= Join [Table[ 1,(len /2}],Table [-1,(len /2}]];
```

This semi-random data could be represented thus:

```
SVMDataPlot[X, y]
```

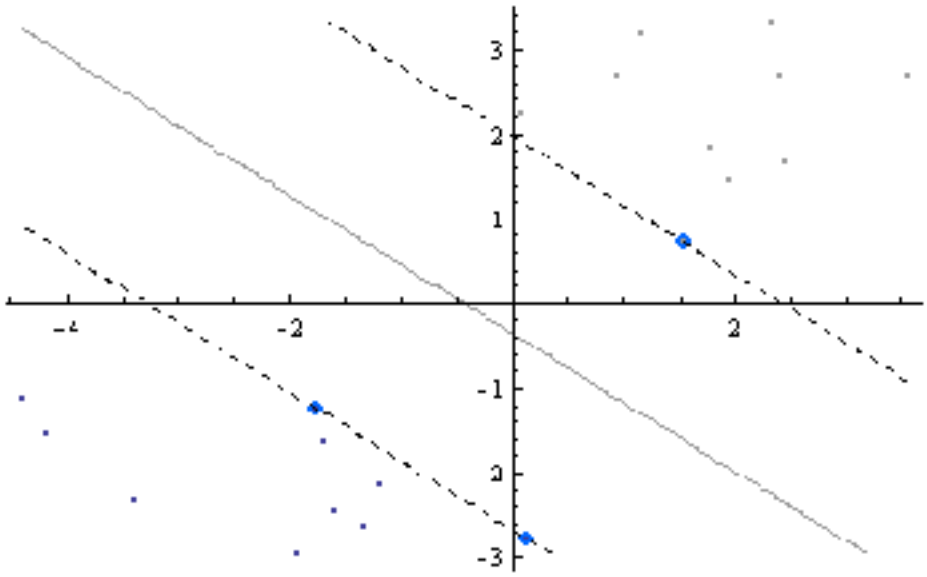
Fig. 2.9 – A plot of generated initial testing data base on a normal distribution.



It is clear to the human brain that this data is, indeed, separable. It can be separated either by mapping into a higher-dimensional space,

```
SVMPlot[a,X,y]
```

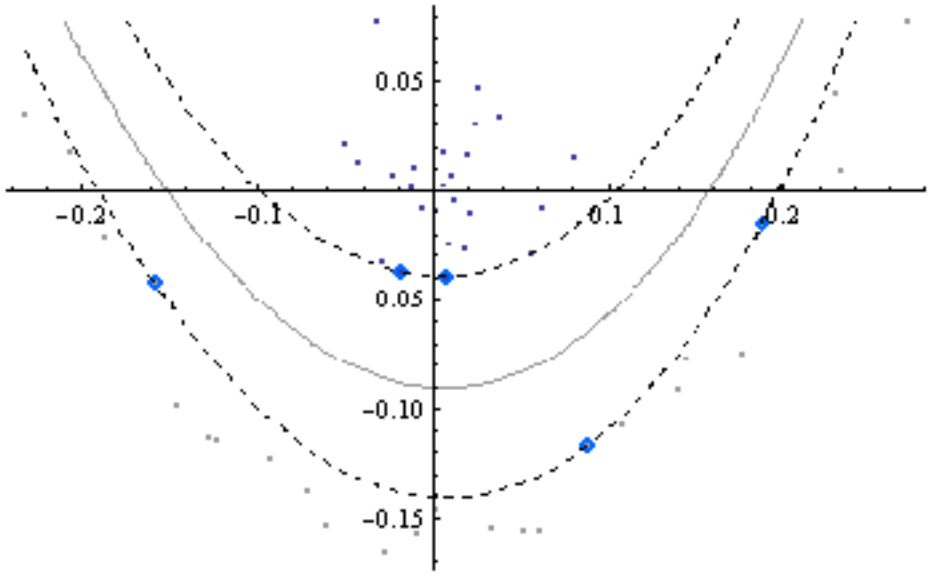
Fig. 2.10 – A plot of the separated initial testing data in higher dimension.



or, perhaps, more elegantly by using a polynomial kernel representing a parabola.

```
p=PolynomialKernel[#1,#2,2]&;
SVMPlot[a,X,y,KernelFunction->p]
```

Fig. 2.11 – A plot of separated initial testing data using a polynomial kernel.



The dotted lines in Fig 2.11 represents the support vectors, the solid grey line the separating hyperplane, and the dark blue points the innermost points of each label, through which the support vectors pass.

2.1.3 Fuzzy Support Vector Machines

The Support Vector Machine is a powerful tool, but without extension it has its limitations, principally that it is extremely sensitive to outlying points and so it is extremely reliant on the user assigning suitable values of P[49]. On the one hand, a higher P imposes a larger penalty on misclassified data and so creates fewer support vectors and a smaller separating region. On the other a lower P

sets a smaller penalty for the same error and creates a larger margin. Points are all regarded as being equally important. However, in many datasets some points are actually more important than others. How then do we discriminate between meaningful points and noisy or outlying ones? In Fuzzy Support Vector Machines (FSVMs) we assign a membership value to each.

For a training set, Θ_{fuzzy}, of N data points,

$$\Theta = \{x_i, y_i, m_i\}_{i=1}^{N} \qquad [2.18]$$

where each x_i belongs to one of the two classes ($y_i = \pm 1$) and has a membership value of m_i, between 0 and 1. Correspondingly, $m_i \xi_i$ represents the error.

If we follow the same pathway as for SVMs, we arrive at the optimal hyperplane,

$$\min L(w, \xi_i) = \frac{1}{2} w^2 + P \sum_{i=1}^{N} m_i \xi_i \qquad [2.19]$$

subject to

$$y_i\left(w\phi(x_i) + b\right) \geq 1 - \xi_i \text{ where } i = 1,2,3...N \text{ and } \xi_i \geq 0 \qquad [2.20]$$

where, again, P is a user-defined penalty constant. In this approach, however, a larger m_i results in a more influential ξ_i; i.e. point x_i is considered to be more meaningful.

The Lagrangian dual optimisation problem now becomes (through the introduction of α as a Lagrange multiplier)

$$\max_{\alpha_i} Q(\alpha_i) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \phi(x_i) \phi(x_j) \qquad [2.21]$$

$$\text{where } 0 \geq \alpha_i \geq m_i P \text{ and } \sum_{i=1}^{N} \alpha_i y_i = 0$$

and so the Karush-Kuhn-Tucker conditions are

$$\alpha_i \left(y_i \left(w\phi(x_i) + b \right) \geq 1 - \xi_i \right) = 0 \qquad [2.22]$$

$$(m_i P - \alpha_i) \xi_i = 0 \qquad [2.23]$$

As m_i, the membership value, has been incorporated into the bounds of α_i we can now separate outliers from meaningful anomalies, by inspecting whether they are either the result of the mathematical mapping, or some important characteristic of the dataset. The two types of support vector are analogous with SVM (lying on a margin, $0 \leq \alpha_i < m_i P$; misclassified $\alpha_i = m_i P$).

Therefore, from the values of α_i we can work backwards to discover

$$w_0 = \sum_{i=1}^{N_v} \alpha_i y_i \phi(x_i)$$

where N_V is the number of support vectors, and thence b_0 from Equation 2.22. Therefore, the decision function in the higher-dimensional feature space is

$$g(x_i) = \Lambda\left(\alpha_i y_i \phi(x_i)\phi(x) + b_0\right) \qquad [2.24]$$

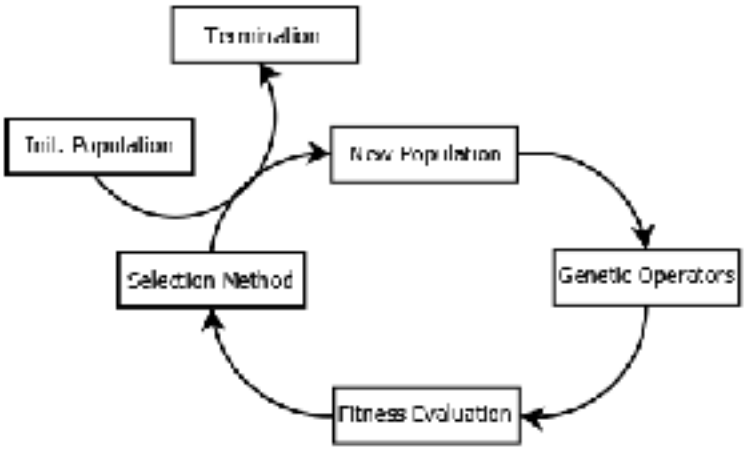
Mercer's Theorem[50]: The relative simplicity of SVMs relies on the Kernel Trick. Implicit in the above calculations is the need to compute $\phi(x_i)$, which could be somewhat problematic. However, if we make use of a kernel which obeys Mercer's Theorem this is not necessary. The Mercer theorem allows us instead to replace every dot product [such as $\phi(x_i)\phi(x_j)$ or $\phi(x_i)\phi(x)$] with a non-linear kernel function [$K(x_i,x_j)$ and $K(x_i,x)$ respectively], which can be obtained in the lower-dimensional primal space. Furthermore, this allows the decision function to be expressed as

$$g(x_i) = \Lambda\left(\alpha_i y_i K(x_i, x) + b_0\right) \qquad [2.25]$$

2.1.4 Genetic Algorithms

A Genetic Algorithm (GA)[51], comprises an array or population of strings (genotypes) each of which encodes a potential solution (phenotype) to the problem; this population evolves over successive cycles to represent a better and better solution. The system is based on the Building Block Hypothesis[52], which indicates that the components of a solution can be separated into blocks, and then combined or crossed over with other possible solutions in order to find the best solution. This process mimics biological selection. In order to process a GA, we must define a *fitness function* through the use of which the posited solutions are compared, and a *solution domain* (the inputs).

Fig 2.12 - Schematic of a GA



In our biological analogy, in each cell there is a set of chromosomes. Chromosomes are strings of DNA which act as a representative of the whole

system. A chromosome consists of genes, building-blocks of DNA, each of which encodes a protein (*viz.* each gene represents a characteristic, e.g. hair colour); potential options for characteristics (e.g. red, brown) are known as alleles. The complete sum of all of the chromosomes is the genome, and a particular series of genes in a genome is a genotype. The genotype gives rise to the organism's phenotype, its physical traits.

Initially the artificial genotypes (usually binary) are randomly selected. After each cycle the individual phenotypes are evaluated and those which have the greatest fitness are selected stochastically; these are then spliced and/or mutated to create a new population. This process terminates at a pre-defined number of cycles, or when the fitness has reached a certain level. This method is obviously an artificial analogue of the principal of evolution, and survival of the fittest.

At the beginning of each cycle *breeding* involves the simple cross-over of information. This is the artificial analogue of DNA cross-over during the Pachytene Stage of Prophase I in Meiosis, in which two chromosomes line up and swap sections of code. Some solutions are *mutated* (only on a point mutation basis; c.f. biological mutation). This is achieved through the selection of a random variable, which is then altered by a small, random amount. If this mutation is successful it may be retained, through *elitism*. Nonetheless, mutations in most instances will be retained whether or not they are beneficial.

It is obviously necessary to decide which of the chromosomes will be bred or mutated. There are numerous potential methods by which this selection can take place. For example:

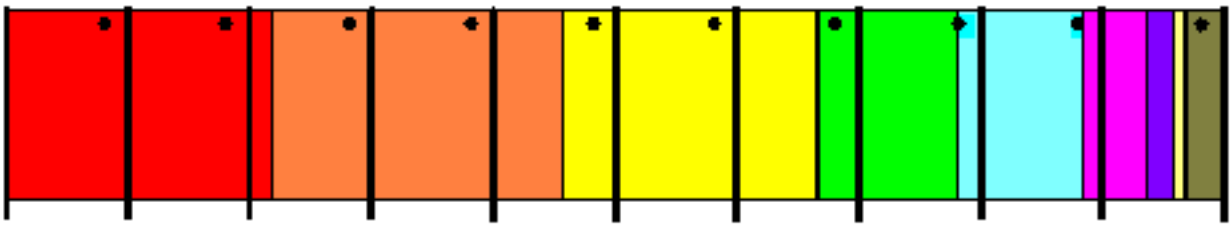
Truncation[53] allows only those chromosomes which have performed well to proceed to the next generation, purging the unfittest. However, the loss of the weaker solutions can, counter-intuitively, lead to a slower search for the solution; an overall weak solution might contain an important grain of truth, which would be lost, lowering the diversity of the gene pool. This system obviously has parallels with that of *Survival of the Fittest* outlined by Darwin.

The *Roulette Wheel*[54] technique passes through only those chromosomes which, under a probability function, are shown to aid random selection. However, this process is noisy (the fitness does not always get better in the short term). This makes it less accurate at breeder selection.

Fitness Decay, a novel technique used within my Group, is a mild alteration of the Roulette Wheel technique, so that the probability of a parent or chromosome being selected is proportional to the integral of an exponentially decaying curve, where the difference between the best and actual fitness is plotted against the probability of selection.

First proposed by Baker[55], *Stochastic Universal Sampling* is used as it has no bias. The fitness difference of each pair of neighbouring chromosomes is used to assign imaginary lines, which are then cut into the same number of segments as chromosomes to be selected. A chromosome in the first segment is randomly selected; another chromosome is selected from the same position in each additional segment of the fitness landscape.

Figure 2.13 - The Stochastic Universal Sampling of a fitness landscape. The dot, once randomly generated in the first segment, is replicated in the identical position in additional segments.



2.1.5 Voted Perceptrons/Boosting

Voting represents the simplest way of combining multiple classifiers. Schapire[56] combined three weak learners to generate a strong learner, with an overall reduced error rate. However this method required a very large training sample, so in 1996 Fruend and Schapire[57] proposed Adaboost, which re-uses data, and so the set no longer needs to be large. In 1998 Freund and Schapire claimed that the improved accuracy was due to a wider margin (see SVMs below). Finally,

Freund and Schapire, in 1999, proposed the Voted Perceptron[58], which is an improved version of the Perceptron model proposed by Rosenblatt in 1958[59].

With a kernel-based Voted Perceptron system, in principle, if we have inputs

$$\langle (x_1, y_1) \ldots . (x_m, y_m) \rangle$$

where we desire a list of weighted perceptrons, in number of cycles *T*, represented by

$$\langle (v_1, c_1) \ldots . (v_k, c_k) \rangle \qquad [2.26]$$

then training takes place as follows

$$k, v_1, c_1 = 0$$

For $i = 1$ to *m*:

compute prediction, $\hat{y} = \text{sign}(v_k \cdot x_i)$ [2.27]

where $v_k \cdot x_i \sum_{j=1}^{k} y_v, K(x_{vj}, x_i);$

if: $\hat{y} = y$ then $c_k = c_k + 1$

else: $v_{k+1} = i_i$

$c_{k+1} = 1$

$k = k + 1$

Repeat *T* times.

In order to make a prediction in an unlabelled instance *x*, given a list of weighted Perceptrons

$$\langle (v_1, c_1) \ldots . (v_k, c_k) \rangle$$

$$\text{predicted label, } \hat{y} = \text{sign}(s) \qquad [2.28]$$

$$\text{where } s = \text{sign}\left(\sum_{i=1}^{k} c_i \, \text{sign}(v_i \cdot x) \right) \qquad [2.29]$$

$$\text{and } v_i \cdot x = \sum_{j=1}^{i} y_v, K(x_{v\,j}, x) \qquad [2.30]$$

VPs have shown excellent promise in the field[60].

2.1.6 Self-Organising Maps

A self-organising map (SOM), also known as the Kohonen[61,62,63] Map, is an ANN trained using unsupervised learning (*viz.* it is not necessary to know the correlation between the input and output variables beforehand) to produce a low-dimensional (usually two), discrete representation of the high-dimensional input space of an array of training samples. The SOM is a subclass of neural networks, and should be considered in order to produce a comprehensive comparison. However, SOMs are different from other ANNs in that they use a neighbourhood function to determine the course and quality of training.

Unsupervised training creates two-dimensional representations of the training data, and so is useful for visualising multi-dimensional data. They are formed from an array of components called nodes, associated with each of which is a weight vector and a position in space. This space, or map, is usually formed of regularly-spaced nodes on a hexagonal or rectangular grid, and describes nodes which have been mapped from a higher-dimensional input space to a lower-dimensional space. The method of allocating places for a vector from data space onto the map is to place it with those nodes which have the closest weight vector.

For a visual representation see Fig 2.14, in which hexagonal cells are used. Each input neuron is connected to all output neurons, with each connection being assigned a weight. One area in which SOM excels is the reduction of dimensionality[84]; therefore, there are always many more input vectors than output.

To continue our analogy of artificial intelligence to the mammalian brain, SOMs are an analogue of separate parts of the cerebral cortex in the human brain which handle sensory information. The training aim in self-organising maps is to associate different areas of the SOM to certain input patterns. Random weights having been assigned to the whole map as an initial step, a training sample is given to the SOM. Its ordinary (or Euclidian) distances to the weight vectors are calculated, using

$$\delta = \sqrt{\sum_{i=1}^{n} (\chi_i - w_i)^2} \qquad [2.31]$$

in which δ is the shortest distance in Euclidian n-dimensional space; χ and w are points in said space, with each component represented by subscript index.

The node which produces the smallest distance is tracked and called the BMU, or Best Matching Unit, and it and its neighbours' weights are adjusted by competitive learning. The cells' weights can interact using a variety of different neighbour functions; often Gaussian.

Fig 2.14 – A hexagonal self-organising map. Similar colours represent similar vectors, which cluster together, following dimensionality-reduction.

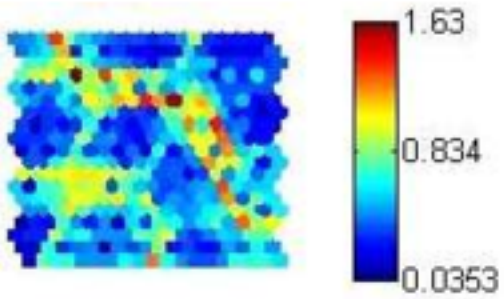
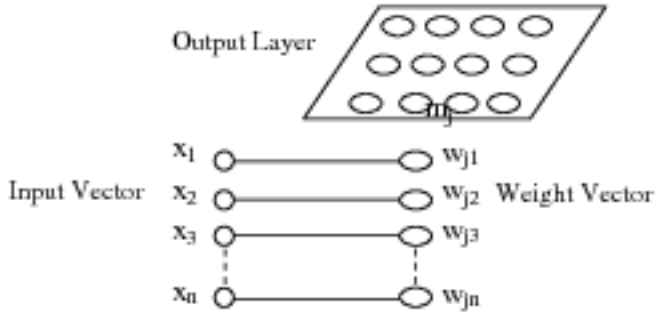


Figure 2.15 – The Structure of a Self-Organising Map, showing the connection between the n-dimensional input vector, *x*, and the weight vector *wj* of the *j*-th map unit *m*.



In common with other neural networks, SOMs include two processes: training and mapping.

It is necessary to define quite a variety of user-set variables to best use SOM. These are:

i. *The size and shape of map.* Maps are either a hexagonal or rectangular lattice of neurons, and the width and height of this lattice must be defined. Final resolution can vary wildly depending on the size of the map: if too small the map will be inflexible; if too large it will take too long to calculate and the generalisation ability will fade as the map grows larger. This, naturally, makes interpretation more difficult;

ii. *The learning rate function.* Many possible functions exist, for example:

Linear	$\alpha(t) = \alpha_0(1 - t/T)$
Exponential	$\alpha(t) = \alpha_0(k/\alpha_0)^{t/T}$
Reciprocal	$\alpha(t) = \alpha_0/(1 + kt/T)$

where $\alpha(t)$ is the learning rate at time, *t*, *T* is the training length and α_0 is the initial learning rate;

iii. *The initial learning rate.* If the learning rate (between 0 and 1) is too low the network will take too long to classify the training data; too high, and the training weights will oscillate and fail to settle;

iv. *The neighbourhood radius* is the distance from the best neuron over which other neurons are affected and influenced to cluster;

v. *The number of training cycles;*

vi. *The neighbourhood function* (analogue of transfer function in ANNs). The possibilities are:

Gaussian	$h(t,d_{bj}) = e^{-\delta_{bj}^2 / 2\sigma_t^2}$
Cut Gaussian	$h(t,d_{bj}) = e^{-\delta_{bj}^2 / 2\sigma_t^2} S(y)(\sigma_t - \delta_{bj})$
Bubble	$h(t,d_{bj}) = S(y)(\sigma_t - \delta_{bj})$
Epanechicov	$h(t,d_{bj}) = \max\{0, 1-(\sigma_t - \delta_{bj})^2\}$

where σ_t is the neighbourhood radius at time, *t*, *S(y)* is the step function [*S(y)* = 0 if $y < 0$; *S(y)* = 1 if $y \geq 0$].

2.2 Introduction to Molecular Descriptors

A molecular descriptor specifies some feature of a molecule, such as its dipole moment, size, or electron distribution, in an unambiguous way; the complete description of a molecule thus requires many descriptors[65]. Since we can anticipate that the effectiveness of drugs must be related in some way to properties at the molecular level, it is evident that connections exist between a suitably-chosen set of molecular descriptors and the activity of a drug.

Geometric descriptors are calculated from 3-dimensional structures and are dependent both on stereochemistry and conformation. Two of the most common such descriptors are volume and surface area, which both provide information on a molecule's stereochemistry. Molecular refractivity, $\pi = \frac{M}{\rho} \times \frac{(n^2 - 1)}{(n^2 + 2)}$, where M is the RMM, n is the index of refraction and ρ the density, can also be used; refractivity is related to the Van der Waals forces that are involved in the interaction with drug-receptors. A typical QSAR, log P, the partition coefficient for water-octanol, is associated with hydrophilicity; biological activity is often associated with log P.

Topological descriptors represent sequences of atoms and how they are connected. It can be said that the success of this type of descriptor is because of relationships between a molecule's topological make-up and its intermolecular

interactions. The descriptors I use tend to be a representation of a complex combination of these two types.

2.2.1 Principal Component Analysed Mass Spectra

Principal Component Analysed Mass Spectra (PCAMS) is one of the novel approaches created as part of this research. Although the most widely used molecular descriptors are a direct expression of properties of the molecule itself, such as its dipole moment or molecular volume, there are many examples where some derivable property of the molecule, such as its infrared spectrum, has been used as a descriptor: this is a standard approach in Computational QSAR analysis. I have therefore investigated whether the mass spectra of candidate drugs might constitute suitable proxy molecular descriptors. As mass spectra contain substantial amounts of information about a sample, it is reasonable to suppose that a single mass spectrum might be able to replace a substantial number of more conventional descriptors.

However, these data alone have an extremely high dimensionality and are extremely sparse. Under such conditions, most Artificial Intelligence techniques will be unworkable; it is therefore necessary to reduce the dimensionality of the data severely to prevent over-fitting or memory overloading. One way of achieving this result, without sacrificing too much of the variance of the original data, is to perform a Principal Component Analysis (PCA)[66] on the peak data.

Principal Component Analysis finds the eigenvalues of the co-variance matrices, providing new molecular descriptors which although fewer in number represent (by linear weighted combination) a large proportion of the variation in the data. An orthogonal linear transformation, PCA maps data to another co-ordinate system so that the greatest variance from any data projection is represented by the first principal component, the second greatest by the second component, and so on.

First the dataset is organised into matrices, the empirical mean is found, as are deviations from it. These allow a matrix of the co-variance to be compiled.

The Co-variance Matrix,

$$C = E(X - \mu\sigma) \otimes (X - \mu\sigma) = E(X - \mu\sigma).(X - \mu\sigma)^*,$$

where E is the expected value operator, μ = the mean, σ = the deviation and X is the matrix of data. If X is real, then a conjugate transposed matrix is the same as the original. The eigenvalues are then found and used as basis vectors for mapping into a new dimension.

Once all of the principal components have been found, some less important ones can be discarded. This process loses some data: the usually acceptable amount of loss is approximately 5%[67]. As a rough guideline it is not sensible for the number of molecular descriptors to exceed the population of the training

set. Therefore, as matrix diagonalisation is iterative in this approach, it can process indefinitely, becoming an increasingly good representation. It, therefore, makes sense to invoke a success criterion to decide when the data are ready to use: this is often set as when the same number of vectors as the population of the training set represents at least 95% of the variance of the original data. Occasionally however, it is necessary to use fewer components as descriptors, when there are insufficient training examples, and where, if too many descriptors were to be used, then the Artificial Neural Network would over-fit the data, because there would be many more connections between the nodes than examples by which to train their weights.

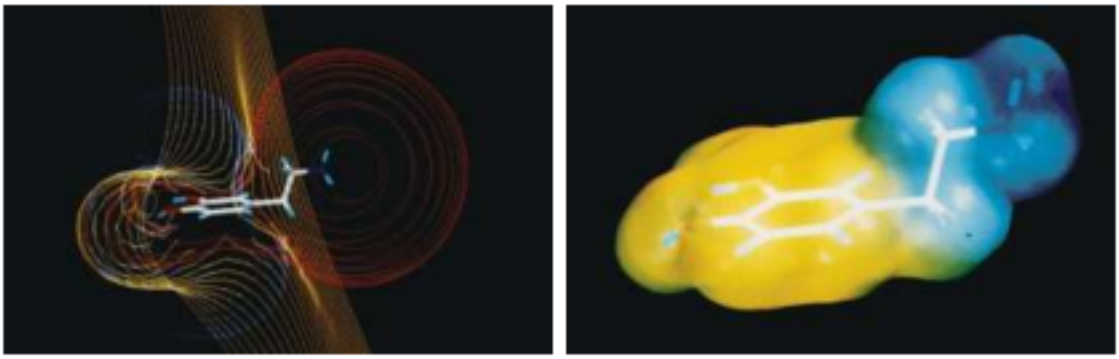
2.2.2 Comparative Molecular Field Analysis

CoMFA[68] is a molecular descriptor used for many years in QSAR discovery. It collates steric and electrostatic values as well as Clog*P* values (hydrophobic parameters[69]). In CoMFA, the molecules are mapped into a 3D lattice and the steric and electrostatic fields at the various grid points are calculated. This array of values is then analysed by Partial Least Squares.

While Classical QSAR equates biological activities of drugs with physical properties which represent certain structural features[70], in general QSAR analyses consider neither the 3D structure of a drug nor its chirality[71]. An active site at a receptor when interacting with a ligand often does not behave as this model implies. First, it would experience the electrostatic potential of the

molecule (see Figure 2.16) and then, when closer, the body of the molecule with its distribution of charge patterns. Cramer and Milne (1979)[72] aligned molecules in space and then mapped their molecular fields to a 3D grid. This provides a better representation of both the model, and the interactions of its active site with potential ligands.

Fig 2.16 – (L) Electrostatic potential contour lines of dopamine; (R) surface of the positively charged form of dopamine [Cramer and Milne[72]]



2.2.3 Comparative Molecular Similarity Indices Analysis

Comparative Molecular Similarity Indices Analysis (CoMSIA)[73] is less well-known and was developed at BASF Ludwigshafen, in Germany. CoMSIA is similar to CoMFA, except that as well as steric and electrostatic features, hydrogen bond donor, hydrogen bond acceptor and hydrophobic values at grid points are also considered. The 3D method gives the conformation/spatial orientation of molecules. Similarity indices are calculated using:

$$A^q F,K,(j) = -\sum W_{probk} W_{ik} e^{-\alpha r/2 s_{ij}} \quad [2.32]$$

where *A* is the similarity index at grid point (*q*), summed over all atoms (*i*) of the molecule (*j*). $W_{probe,k}$ is the probe atom. W_{ik} is the real value of the physical property (*k*) of atom *i*. r_{ij} is the distance between the probe atom and atom *i* of the test molecule. α is the attenuation factor.

2.2.4 Hologram QSAR

Hologram QSAR[74], developed by Tripos, is another popular QSAR descriptor, principally because it does not require the exact 3D information for the molecules. Molecular holograms are calculated by generating linear and branched fragments which can be distinguished by atoms, bonds, connections, number of hydrogen atoms and chirality. HQSAR works by discerning the patterns in the fragments related to activity in sets of molecules. The cytotoxicity of each fragment gives a prediction of the overall cytotoxicity of the molecule.

2.2.5 EVA (EigenVAlue) Descriptors

EVA data, which were first developed at Shell Research Ltd., are derived from the fundamental vibrations of a molecule and resemble an Infrared spectrum. Fundamental vibrations are calculated through the use of normal coordinate analysis derived from the fully-optimised geometry of the molecule. Eigenvalues (f_i) of the normal co-ordinates reflect vibrational frequencies;

eigenvectors atomic displacements. However, the asymmetrical molecules in which we are interested have (3*N*−6) vibrational degrees of freedom. Therefore, the eigenvalues of the normal co-ordinates would be ambiguous since the descriptor would not be of a consistent length. To overcome this, eigenvalues can be mapped onto a frequency scale with boundaries carefully selected to represent the complete potential range of frequencies. All lines can then be replaced by Gaussian curves, their mean at f_i, with a fixed Standard Deviation, σ, and unit area. The summation of the curves can then be expressed as:

$$EVA(x) = \sum_{i=1}^{3N-6} \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{1}{2}\frac{(x-f_i)^2}{\sigma^2}\right) \qquad [2.33]$$

where x represents the range of vibrations.

EVA descriptors are rotationally and translationally invariant: every example in the descriptor system has a standard meaning (if σ and L are constant) and produces a unique code.

2.2.6 CoDESSA

CODESSA[75] (Comprehensive Descriptors for Structural & Statistical Analysis) is a system which derives a variety of descriptors from molecules, integrated within a series of mathematical tools. However, CoDESSA is computationally time-consuming, similar to other techniques, and in preliminary experiments

seemed to perform no better than other molecular descriptor systems. It will not, therefore, form part of the research in this thesis.

2.2.7 The Molecular Fragment Approach

MFA discovers the common fragments in a limited set of molecules and translates these into binary numerical outputs. However, one of the principal disadvantages of adopting this final approach is its reliance on common molecular fragments, which reduces the pool of potential candidates dramatically: they must all be reasonably similar molecules. This method has therefore been discounted as crude and ill-suited to this approach.

References

[35] Stryer L., Biochemistry, 4th Edition, WH Freeman and Company, New York (1995).

[36] Arbib M.A., Handbook of Brain Theory and Neural Networks, 2nd Edition, Bradford Book, The MIT Press (2003).

[37] Anderson, J.A., An Introduction to Neural Networks, Bradford Book, The MIT Press (1995).

[38] Liang J., Gupta, M.M. Stable dynamic backpropagation learning in recurrent neural networks, *IEEE Transactions on Neural Networks*, 10 (6) 1321 - 1334 (1999).

[39] Xie, X., and Seung, H. S., Equivalence of backpropagation and contrastive Hebbian learning in a layered network, *Neural Computation* 15, 143–163 (2003).

[40] De Wolf, E. and Francl, L., Neural Networks that distinguish infection periods of wheat tan spot in an outdoor environment, *Phytopathology* 87 (83) (1997).

[41] Rayudu R. and Samarasinge S., A network of neural nets to model power system fault diagnosis, *The Proceedings of the Fourth International Conference on Neural Information Processing*, Springer, Berlin, 12 (1), 1939–1944 (1997).

[42] Jiang D. et al., Progress in developing an ANN model for air pollution index forecasting, *Atmospheric Environment* 38, 7055 (2004).

[43] Rajanayake C. et al., Solving the inverse problem in stochastic groundwater modelling with artificial neural networks, *Proceedings of 1st Biennial Congress of the International Environmental Modelling Society* (2002).

[44] Chandraratne M.R. et al., Prediction of lamb tenderness using image surface texture analysis, *Journal of Food Engineering* (2005).

[45] Limsombunchai V. and Samarasinge S., House price prediction: hedonic price model vs artificial neural networks, *Kasetsar University Journal of Economics* (2005).

[46] Cristianini N., Shawe-Taylor J., Support Vector Machines and other kernel-based learning methods, Cambridge (2000).

[47] The Screensaver Lifesaver Project, University of Oxford: http://www.chem.ox.ac.uk/curecancer.html, accessed 01/02/2010.

[48] Muller K., Mika S., Riitsch G., Tsuda K., and Schölkopf B., An Introduction to Kernel-Based Learning Algorithms, *IEEE Transactions on Neural Networks* 12, 181-201 (2001).

[49] Lin C.F. and Wang S-D., Fuzzy support vector machines, *IEEE Transactions on Neural Networks* 13 464–471 (2002).

[50] Carmeli C., De Vito A., and Toigo E., Reproducing kernel Hilbert spaces and the Mercer theorem, *Analysis & Applications* (2006).

[51] Musser D.R. and Stepanov A.A., Generic programming, *The Proceedings of the 1st International Joint Conference ISSAC 88 and AAEC-6, Lecture Notes Computer Science 358*, Springer, Berlin 13–25 (1989).

[52] Wu A.S., and Lindsay R.K., A Comparison of the Fixed and Floating Building Block Representation in the Genetic Algorithm, *Evolutionary Computation*, 4 (2), 169-193 (1996).

[53] Muehlenbein H. and Schlierkamp-Vosen D., Predictive Models for the Breeder Genetic Algorithm, I. Continuous Parameter Optimizations, *Evolutionary Computation*

1 (1), 25-49. (1993).

[54] Goldberg D. and Deb K., A comparative analysis of selection schemes used in genetic algorithms. *Foundations of Genetic Algorithms*, Rawlins G. (Ed.), Morgan Kaufmann, San Mateo, CA, 69-93 (1991).

[55] Baker, J.E. Reducing Bias and Inefficiency in the Selection Algorithm, *Proceedings of the Second International Conference on Genetic Algorithms and their Application (Hillsdale)*, Springer, Berlin, 4-21 (1987).

[56] Schapire R.E., The Strength of Weak Learnability, *Machine Learning* 5 197-227 (1990).

[57] Freund Y. and Schapire R.E., Experiments with a New Boosting Algorithm, *Proceedings of the Thirteenth International Conference on Machine Learning* L Saitta (Ed.), 148-156 (1996).

[58] Freund Y., Schapire R.E., Large Margin Classification Using the Perceptron Algorithm, *Machine Learning* 373, 277-296 (1999).

[59] Rosenblatt, F. , The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* 65, 386-407 (Reprinted in *Neurocomputing*, MIT Press, 1988) (1958).

[60] Wachman, G., Kernel Methods and Their Application to Structured Data, PhD thesis, Tufts University (2009).

[61] Kohonen, T., Self-organized formation of topologically correct feature maps, *Biological Cybernetics* 43, 59-69 (1982).

[62] Kohonen T., Self-Organization and Associative Memory, Springer, Berlin (1984).

[63] Kohonen, T., Self-Organizing Maps, Third, extended edition, Springer, Berlin (2001).

[64] Smith A.J., Applications of the self-organising map to reinforcement learning, *Neural Networks* 15 (8-9), 1107-1124 (2002).

[65] Hansch C. and Leo A., Exploring QSAR. Fundamentals and Applications in Chemistry and Biology, *Journal of the American Chemical Society* (1995).

[66] Wold S., Esbensen K., and Geladi P., Principal Component Analysis, *Chemometrics and Intelligent Lab. Systems* 2 (37) (1987).

[67] Matter, H., Selecting optimally diverse compounds from structure databases: a validation study of two-dimensional and three-dimensional descriptors, TRIPOS GmbH, München, Germany, *Journal of Medicinal Chemistry* 40 (1997).

[68] Kubinyi H., Comparative Molecular Field Analysis (CoMFA), BASF AG, D-67056 Ludwigshafen, Germany, Handbook of Chemoinformatics: From Data to Knowledge in 4 Volumes, Wiley, New York (2008).

[69] Leo, A., Jow, P. Y. C., Silipo, C., Hansch, C., Calculation of hydrophobic constant (log P) from pi and f constants, *Journal of Medicinal Chemistry* 18, 865–868 (1975).

[70] Kubinyi H., Burger's Medicinal Chemistry, Vol. I, Wolff M.E. (Ed.), 5th edition, Wiley, New York, 497-571 (1995).

[71] van de Waterbeemd H. (Ed.), Structure Property Correlations in Drug Research, Academic Press, Austin, TX, (1996).

[72] Cramer, R. D. III and Milne M., Comparative Molecular Field Analysis (CoMFA), *Journal of the American Chemical Society*, Computer Chemistry Section, 44 (1979)

[73] G. Klebe, U. Abraham, and T. Mietzner, Molecularity similarity indices in a comparative analysis (CoMSIA) of drug molecules by comparative field analysis, *Journal of Medicinal Chemistry* 37, 4130-4146 (1994).

[74] Tripos. Inc., HQSAR. Version 1.0, St. Louis, MO (1999).

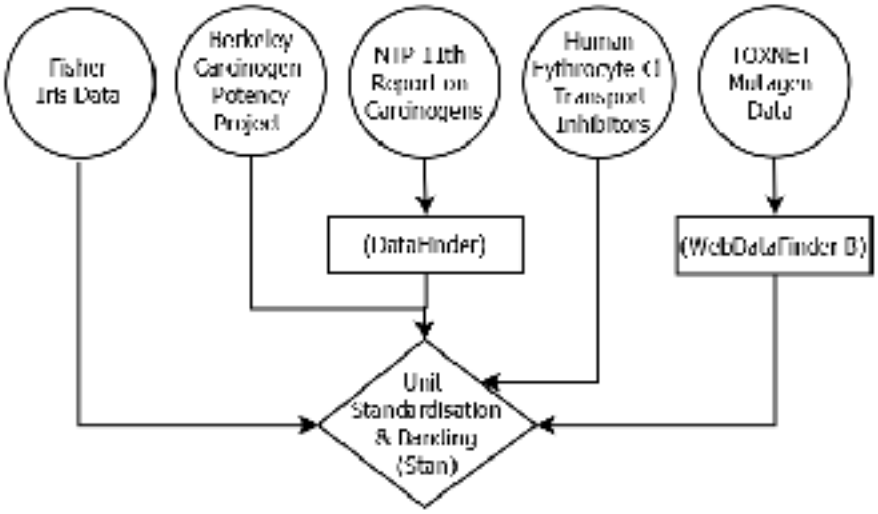
[75] Katritzky, A. R., Lobanov, V. S., and Karelson, M. CODESSA, Version 5.3, University of Florida, Gainesville, FL (1994).

Chapter 3

The Transformation of Raw Data into Viable Molecular Descriptors

3.1 Introduction to Series of Data

Figure 3.1 – A data flow schematic for input testing data



3.1.1 Iris Data

Fisher's Iris dataset[76] is a multivariate dataset used for testing in machine learning. Introduced by Ronald Fisher in 1936, the dataset consists of 50 samples from each of three species of Iris flowers. The values of four features of examples of each species were measured. Based on the combination of the four features, Fisher developed a linear discriminant model to distinguish between the Irises[77]. These data are useful for preliminary testing and optimisation of programs and techniques.

3.1.2 Carcinogenicity (TD_{50}) in rats

Despite some concerns[78], it is generally accepted in the literature that the effect of a carcinogen on a rat is broadly similar to the analogous effect on a human.

Oral TD_{50} data for adult rats (*viz.* the dosage necessary to kill 50% of the population when taken orally) was sourced from the Berkeley University Carcinogenic Potency Database (CPDB)[79], August 2007 release. These data pertain specifically to carcinogenicity, it having been established after the death of the rat that a tumour was indeed the cause. Provided by academic researchers, supposedly without discrimination or bias, these data are also rigidly verified and made available for public scrutiny. Once acquired they were further concatenated and standardised with equivalent data from the National Toxicology Program's 11th Report on Carcinogens[80], to form a testing dataset which comprised 538 examples, catalogued according to CAS Number.

Dosage values were banded into logarithmic bands, which were base-shifted so that Band 1 represented the least carcinogenic chemicals and Band 9 represented the most carcinogenic. This was done by taking the logarithm of the TD_{50} value from the literature, rounding to the nearest integer and adding 5.

$$\text{i.e. band,b} = \text{rnd}(\log(\chi)) + 5$$

where χ equals the TD_{50} value of carcinogenicity or teratogenicity. I carried out this data manipulation, concatenation, and standardisation through the use of various Java programs, which process was rigidly and independently checked against the original datasets to ensure that the foundation for this research was sound and accurate.

3.1.3 Human Erythrocyte Cl^- Transport Inhibitors

Data regarding the inhibition of KCl transport in Human Erythrocytes, by bumetanide and furosemide analogues[81] were donated by the Wilkins Group of the Department of Physiology, Anatomy and Genetics, at the University of Oxford[82]. These data are important in the treatment of various cancers. The identification of individual transport inhibitors is an important aim when characterising discrete membrane transport systems; these pathways are of interest as chemotherapeutic targets. This data was again banded logarithmically into 5 bands, representing the lowest concentration at which significant inhibition was detected: 1 was a low concentration, and 5 was a high concentration. Appropriate CAS numbers were found for each of the chemicals concerned, so that CML and Mass Spectrometry Data could be matched to them.

3.1.4 TOXNET Mutagen Data

TOXNET (TOXicology Data NETwork)[83] is a cluster of databases covering toxicology, carcinogenicity, teratogenicity, and mutagenicity data, created by the Toxicology and Environmental Health Information Program (TEHIP) at the National Library of Medicine (NLM) in the USA. It principally acts as a concatenation of many databases and data sources. These data include:

i *HSDB (Hazardous Substances Data Bank)*[84]

A database of the toxicologies of over 5,000 potentially hazardous

chemicals. The data are fully referenced and peer-reviewed by a Scientific Review Panel.

ii *IRIS (Integrated Risk Information System)*[85]

A database from the USA's Environmental Protection Agency (EPA) containing carcinogenic data on more than 500 chemicals. IRIS risk assessment data have been scientifically reviewed by EPA scientists.

iii *CCRIS (Chemical Carcinogenesis Research Information System)*[86]

A data bank, developed by the USA's National Cancer Institute (NCI), containing over 9,000 records of carcinogenicity, mutagenicity, tumour promotion, and tumour inhibition data. Test results have been reviewed.

iv GENE-TOX (Genetic Toxicology)[87]

A database developed by the USA Environmental Protection Agency (EPA) containing genetic toxicology test results on over 3,200 chemicals.

The TOXNET "Multiple Databases" system allows querying of the HSDB, IRIS, CCRIS, and GENE-TOX databases. A dataset was created by CAS number from this dataset. These data were downloaded from the TOXNET web site, through the use of a Java program which queried each chemical by CAS number. Some manual oversight was necessary to standardise data.

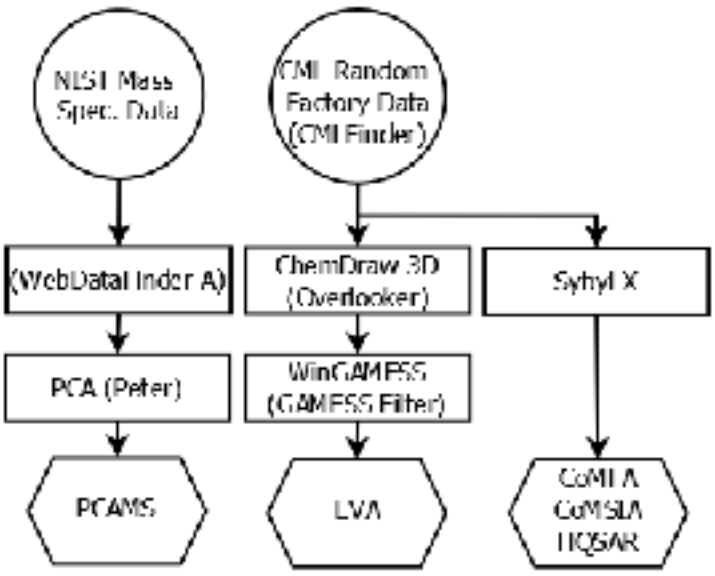
Possible bands of mutagen were:

- Band A: *"Mutagenic to Humans,"*
- Band B: *"Likely to Be Mutagenic to Humans,"*
- Band C: *"Suggestive Evidence of Mutagenic Potential,"*
- Band D: *"Inadequate Information to Assess Mutagenic Potential,"*
- Band E: *"Not Likely to Be Mutagenic to Humans."*

It was important to disregard band D, deleting them from the database, as there was too little information on the chemical for any classification to be meaningful. This resulted in four potential bands.

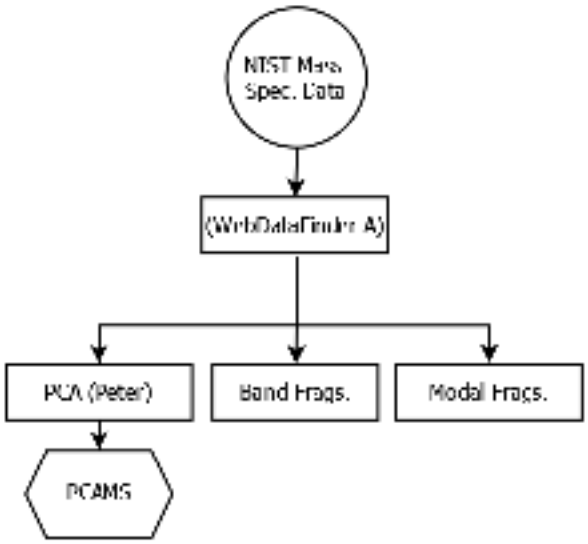
3.2 The Creation of the Molecular Descriptors

Figure 3.2 – A data flow diagram for the creation of the molecular descriptors



3.2.1 Creation of the Principal Component Analysed Mass Spectra Descriptors

Figure 3.3 – A data flow diagram for the creation of and preliminary tests on Mass Spectral molecular descriptors



While it is true that there are many programs which can artificially generate mass spectrometry data[88], accurate experimental data is far superior for use in molecular descriptors. The *National Institute of Science and Technology* (*NIST*), project *WebBook*[89] provides physical data about chemicals arranged by CAS Number.

This data, supplied by a USA Government Department was found to be reliable when cross-referenced against equivalent mass spectra taken in Oxford. A series of Java programs formatted the data into datasets.

However, these data represented the intensities of 610 molecular fragments; the

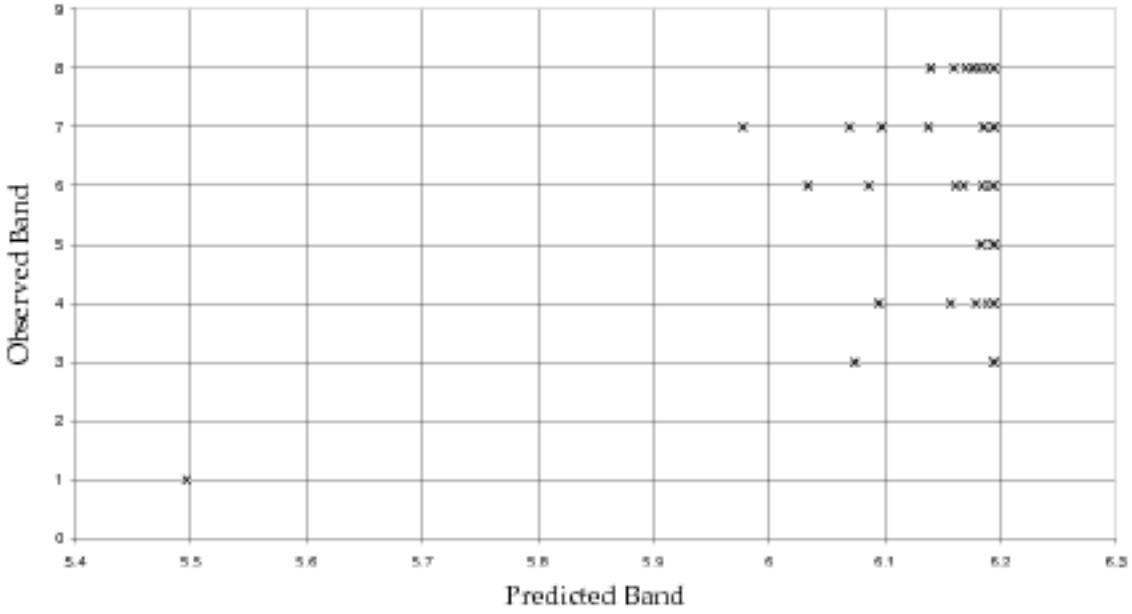
use of the whole would cause inefficiencies in the Neural Networks (sparse data merely takes extra time to process and makes no real contribution to the ANN) and memory problems in the Support Vector Machines. It was, therefore, necessary to reduce the dimensionality of these descriptors.

Various initial experiments were undertaken to test the fidelity of three types of dimensionality reduction. I tried banding the fragments into ranges of values, and using their cumulative intensities. I also tried using the intensities of the most popular fragments as molecular descriptors.

Neither of these methods proved successful – they both often performed as worse predictors of, for example the TD_{50} Carcinogenicity Data, than prediction by random selection! It makes sense that the individual fragments concerned would be important to identify this sort of complex, biological activity, and that these indicative fragments might not be those which are most popular within the dataset.

An example of the results which came from using the intensities of the first 242 peaks as molecular descriptors can be seen in Fig 3.4. This is likely to be because the truncation or banding of the Mass Spectral data would lead to the loss of critical information about the molecule.

Figure 3.4 – The intensities of the first 242 mass spectral peaks as molecular descriptors for the prediction of a chemical's rat TD_{50} carcinogenicity band, using an ANN.



The most efficient, and accurate way to resolve a series of molecular descriptors which accurately represented the data, but in fewer dimensions, was to perform a Principal Component Analysis (*PCA*) on the peak data.

Principal Component Analysis is a dimension orthogonalisation tool, which can also be used for dimension reduction (with commensurate loss of information).

If the average arithmetic mean of a series of data is

$$\bar{x} = \frac{\sum_{i}^{N} x_i}{N}$$

and, the co-variance of a series of data from a population is

$$\sigma_A \sigma_B = \frac{\sum_i^N (a_i - \bar{a})(b_i - \bar{b})}{N-1} \qquad [3.1]$$

then Principal Component Analysis using the eigenvector process involves placing the data into a matrix, and subtracting the average value of each vector. Subsequently, a matrix is made of the co-variances and its eigenvectors are calculated. The eigenvalues can be used to calculate the amount of "blame" for the variance which is held by each of the component vectors.

The iterative Jacobi method calculates *every* eigenvalue and eigenvector in a matrix, which is important in principal component analysis. A non-analytical procedure for the identification of eigenvalues and eigenvectors of big matrices is imperative because of the difficulty of calculating roots of high-order polynomials. The Jacobi system rotates the highest off-diagonal element's dimensions to zero, and repeats this process until all off-diagonals are below a certain value.

An eigenvector square matrix Z with the same dimensions as the co-variance matrix is created with elements

$$Z_{ij} = \delta_{ij}$$

The eigenvalue matrix *Y* is one single column of the co-variance matrix, with the diagonal values of the latter as elements.

An iteration proceeds like this.

1. The largest off-diagonal C_{jk} is selected;
2. The rotation angle of the dimensions is

$$\theta = \frac{\arctan\left(\frac{2 \times C_{jk}}{C_{jj} - C_{kk}}\right)}{2}$$

3. In a rotation about 2 dimensions, *j* and *k*, by angle θ a rotation matrix R would be

$$R_{kk} = \cos\theta \qquad R_{jj} = \cos\theta \qquad R_{jk} = \sin\theta \qquad R_{kj} = -\sin\theta$$

where

$$i \neq j,k \qquad a,b \neq j,k \qquad R_{ii} = 1 \qquad R_{a\,b} = 0$$

4. The operation of rotation is

$$C_{new} = R^{-1} C_{old} R$$
$$E_{new} = R^{-1} Z_{old} R$$
$$Y_{i\,0} = C_{ii}$$

This process was then applied to the dataset of all 532 molecules, which contained intensity data for all molecules from peaks of mass 1 to 610.

Most publicly-available programs for performing Principal Component Analysis, such as SAS[90], WinIDAMS[91], or Excel plug-ins, can only deal with approximately two hundred and fifty data points. Therefore, it was necessary to write a Jacobi rotation matrix program which could deal with this scale of dimensionality: a Java program called *Peter*.

Peter processed the data supplied to it, producing a series of 146 weighted component vectors which described 95% of the information contained in the original input vectors.

Table 3.1 - 146 Principal Components representing 95% of the variance of the original Mass Spec. data.

Transformed Data	*Cumulative Responsibility*
PC 1	0.0709
PC 2	0.1303
PC 3	0.1738
PC 4	0.2076
PC 5	0.2355
PC 6	0.2596
PC 7	0.2823
…	…
PC 140	0.9460
PC 141	0.9468
PC 142	0.9476
PC 143	0.9483
PC 144	0.9491
PC 145	0.9498
PC 146	>0.95

3.2.2 The Creation of the CoMFA Molecular Descriptors

A series of molecular descriptors was created using Tripos's Sybyl-X[92] 3D modelling program. This process was speedy and automated. Data were first converted from Random Factory's CML Data (see 3.2.5 below), into a more useable format using OpenBabel[93], a chemical data format converter. Sybyl then analysed the molecules, calculated the various descriptors and produced a database by CAS Number. This produced 162 descriptors.

3.2.3 The Creation of the CoMSIA Molecular Descriptors

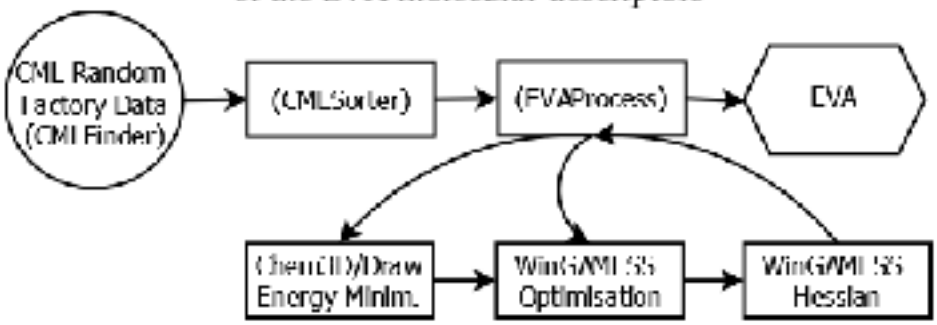
A series of molecular descriptors were created using Tripos's Sybyl 3D modelling program, using the same method as for 3.2.2 above. This produced 157 descriptors, organised by CAS Number.

3.2.4 The Creation of the HQSAR Molecular Descriptors

A series of molecular descriptors were created using Tripos's Sybyl 3D modelling program, using the same method as for 3.2.2 above. This produced 123 descriptors, organised by CAS Number.

3.2.5 The Creation of the Eigenvalue Molecular Descriptor

Figure 3.5 – A data flow diagram for the creation of the EVA molecular descriptors



In order to make a comparison of the PCAMS system of the molecular descriptors against another tried-and-tested system, Eigenvalue (EVA) descriptors, it was necessary to create those descriptors for the same series of molecules used in the training series. First, I acquired CML data for the molecules from *The Random Factory CML Reference Collection*, who supply reliable CML datasets for a cost.

It was then necessary to produce energy-minimised geometries for each of the molecules, so that eigenvalues of their normal co-ordinates, which represent vibrational frequencies, could be calculated. This was achieved, automatically using a Java program, first, by minimising the energies of twenty random conformations of each molecule using Chem3D/Draw[94], selecting the lowest, and then further minimising it through the use of WinGAMESS[95].

WinGAMESS "is a program for *ab initio* molecular quantum chemistry... Computation of the energy Hessian permits prediction of vibrational frequencies, with IR or Raman intensities. Solvent effects may be modeled by the discrete Effective Fragment potentials, or continuum models such as the polarizable Continuum Model. Numerous relativistic computations are available, including third order Douglas-Kroll scalar corrections, and various spin-orbit coupling options. The Fragment Molecular Orbital method permits use of many of these sophisticated treatments to be used on very large systems, by dividing the computation into small fragments. Nuclear wavefunctions can also be computed, in VSCF, or with explicit treatment of nuclear orbitals by the NEO code."[96]

This excellent program supplied free-of-charge by the Gordon Group of Iowa State University, USA, further refined and minimised, through a series of treatments, the molecules' conformation energies, and then calculated the relevant vibrational frequencies, to create the series of molecular descriptors used in EVA. The program did suffer some virtual memory problems on the system, and some processes failed during the optimisation stage; a Java program (GamessFilter) was created to check for errors and repeat cycles where necessary.

These descriptors were then converted into pseudo-descriptors using PCA. This is because EVA descriptors have 800 descriptors, which is too many for the

various techniques to train from usefully. This process produced 103 descriptors.

References

[76] Fisher, R.A., The Use of Multiple Measurements in Taxonomic Problems, *Annals of Eugenics* 7 179–188 (1936).

[77] McLachlan G., Discriminant Analysis and Statistical Pattern Recognition, Wiley, New York (1992).

[78] Rosenkranz H., Cunningham A., Lack of predictivity of the rat lethality (LD_{50}) test for ecological and human health effects, *Alternatives to Laboratory Animals* 33 (1) (2005).

[79] Berkley University Carcinogenic Potency Database: http://potency.berkeley.edu/, accessed 07/01/2010.

[80] National Toxicology Program, Report on Carcinogens: http://ntp.niehs.nih.gov/, accessed 21/02/2010.

[81] Culliford S., Ellory C., Lang H.J., Englert H., Staines H., and Wilkins R., Specificity of classical and putative Cl^- transport inhibitors on membrane transport pathways in human erythrocytes, *Cellular Physiology & Biochemistry* 13, 181–188 (2003).

[82] Cartilage Cellular Physiology Group, Dr Robert Wilkins: http://www.physiol.ox.ac.uk/Research_Groups/Cartilage_Cellular/, accessed 04/11/09.

[83] TOXNET Toxicology Data Network: http://toxnet.nlm.nih.gov/, accessed 08/01/2010.

[84] HSDB (Hazardous Substances Data Bank): http://www.nlm.nih.gov/pubs/factsheets/hsdbfs.html, accessed 08/01/2010.

[85]IRIS (Integrated Risk Information System): http://www.epa.gov/iris/, accessed 08/01/2010.

[86] CCRIS (Chemical Carcinogenesis Research Information System): http://www.nlm.nih.gov/pubs/factsheets/ccrisfs.html, accessed 08/01/10.

[87] GENE-TOX (Genetic Toxicology):
http://www.nlm.nih.gov/pubs/factsheets/genetxfs.html, accessed 08/01/2010.

[88] Drahos L. and Vékey K., Masskenetics, *Journal of Mass Spectrometry* 36, 237 (2001).

[89] NIST WebBook:
http://webbook.nist.gov/chemistry/, accessed 08/01/2010.

[90] SAS Institute, SAS:
http://www.sas.com/software/sas9/, accessed 08/01/2010.

[91] UNESCO WinIDAMS:
http://portal.unesco.org/ci/en/ev.php-URL_ID=2070&URL_DO=DO_TOPIC&URL_SECTION=201.html, accessed 08/01/2010.

[92] Tripos's Sybyl-X:
http://tripos.com/index.php?family=modules,SimplePage,,,&page=SYBYL-X, accessed 08/01/2010.

[93] Open Babel, The Open Source Chemistry Toolbox:
http://openbabel.org/wiki/Main_Page, accessed 08/01/2010.

[94] CambridgeSoft, ChemDraw:
http://www.cambridgesoft.com/software/ChemDraw/

[95] WinGAMESS: http://www.msg.chem.iastate.edu/gamess/win.html, accessed 08/01/2010.

[96] GAMESS: Summary of Capabilities:
http://www.msg.chem.iastate.edu/gamess/capabilities.html, accessed 21/01/2010.

Chapter 4

Processing Methodology and Preliminary Grid-search Results

A Data Flow Chart showing the techniques necessary to ensure a holistic approach
NIST Mass Spec. Data
CMI Ranukon Factory Data (CMI Tasks)
Fisher Int. Data
Berkeley Carcinogen Potency Project
NTP 11th Report on Carcinogens
Human Erythrocyte (I) Transport Inhibitors
IOOMEI Mutagen Data
(WebDataFinder A)
PCA (Peter)
(WebDataFinder B)
ChemDraw 3D (ChemBroker)
WinGAMESS (GAMESS Files)
Sybyl-X
(DataTasks)
EVA
MOPAC
CoMFA CoMSIA HQSAR
Unit Standardisation & Banding (Stan)
(Canocalculator)
Training
Cross validation
Testing
ANN (Ann A)
SVM (Chanon A) Multiclassifica
PSVM (Chanon B) Multiclassifica
Voted Perceptron (GRA)
SOM (Ann B)
GA (Darwin)

1.2 Carcinogenesis, Teratogenesis, and Mutagenesis

Whereas carcinogenicity is a well known property of molecules, teratogenicity (which we will also consider) is a little more obscure. Teratogenesis (literally "monster-birth") is the formation of congenital birth defects, such as cleft palettes or limb deformities. These malformations arise from foetal abnormalities, often caused by an adverse environment, such as interaction with teratogens. Similarly, mutagens (literally "the origin of change") are chemicals which alter the genetic make-up (DNA) of an organism. Whilst it is the case that a chemical often may belong to more than one of these distinct types, the confusion between them in the literature is unhelpful.

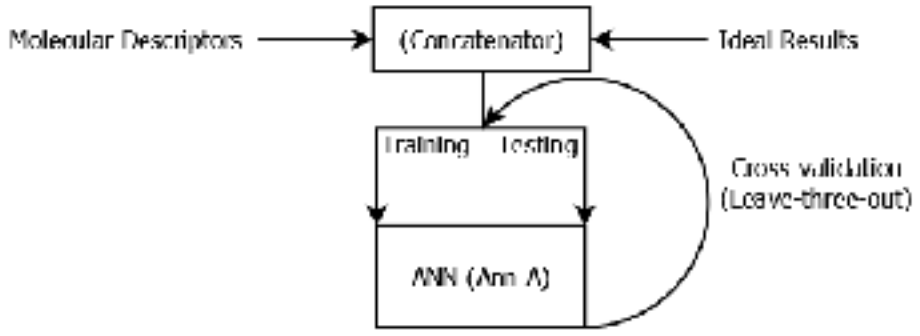
This thesis provides a holistic, quantitative approach to the applications of AI techniques to carcinogen chemistry, while incorporating novel ideas and techniques, tailored to improve and simplify them. Implicit in this research is the need to compare techniques, descriptors, and parameters, and to create a map which allows the user to navigate through this complex field.

In order to compare and contrast ANNs and SVMs when applied in this field, and set their abilities in a proper context, it will be necessary to compare them to a variety of other, more established AI techniques, as well as to each other. These comparators include Genetic Algorithms, Voted Perceptrons and Self-Organising Maps. In each case it will be necessary to optimise every aspect of the technique, its parameters, and its architecture so that it represents the

This Chapter explains the methods and architectures used, and the parameters selected by initial gridsearches. First, we consider ANNs.

4.1 Artificial Neural Networks

Figure 4.1 – A data flow diagram for the Artificial Neural Network



4.1.1 A Novel Backpropagation System[97]

Standard feed-forward Artificial Neural Networks, using a sigmoidal transfer function and a novel backpropagation system (designed to guard against Pareto equilibria, when a Neural Network falls into a false minimum, from which it is unable to escape), were applied to the various series of data. Where there was no further improvement to be found, the weights were saved, a randomisation of the weights occurred, and processing began again. The reasoning for the new system stems from the necessity, with a more complicated data set, to invoke a similar change in error during backpropagation, which will prevent overcorrection. This decreases the time a system takes to arrive at a minimum.

Let $w_{k,i,j}$ = the weight assigned to the connection between two nodes starting in layer k, node i, and layer k+1, node j;

O = output value of a node, following application of the transfer function;

$\varepsilon = T - O$ = the overall error (where T is the target value);

ψ^{-1} = the inverse transfer function;

l = the learning rate;

N_k = the number of nodes in layer k; and

ζ = the inverse gradient of the transfer function.

Then the error of the output node

$$\varepsilon_{k,j} = \psi^{-1}(O_{kj}) - \psi^{-1}(T_{kj}) \qquad [4.1]$$

where $T_{output,j}$ = the ideal output, and the inverse transfer function (reverse of the TF),

$$\psi^{-1}(t) = 100.\log\left(\frac{1+\frac{t}{2000}}{1-\frac{t}{2000}}\right) \qquad [4.2]$$

Therefore, the change in connection weight,

$$\Delta w_{k,i,j} = \zeta \ell \varepsilon_{k+1,j} \frac{O_{k,i} w_{k,i,j}}{\sum_{z} O_{k,z} w_{k,z,j}} \qquad [4.3]$$

This renders the target output of any non-output layer,

$$T_{kj} = \frac{\sum_{i} \frac{\psi^{-1}(T_{k+1,i}).O_{k,j}.w_{k,j,i}}{\sum_{z} O_{k,z} w_{k,z,j}}}{N_{k+1}} \qquad [4.4]$$

where N_k = number of nodes in layer k. Thereafter, the system works backwards through the network.

This system differs by calculating the *inverse* error gradient[a] at each node and then uses this to calculate the amount by which the weights should be altered in the subsequent iteration. This allows the network to spring out of Pareto equilibria when the amount by which each weight changes in each iteration is high, while not adversely affecting the final equilibrium because the change increment is small towards the end. This system was found to reduce typical processing time by 11% as fewer iterations are necessary to achieve the same equilibrium[b].

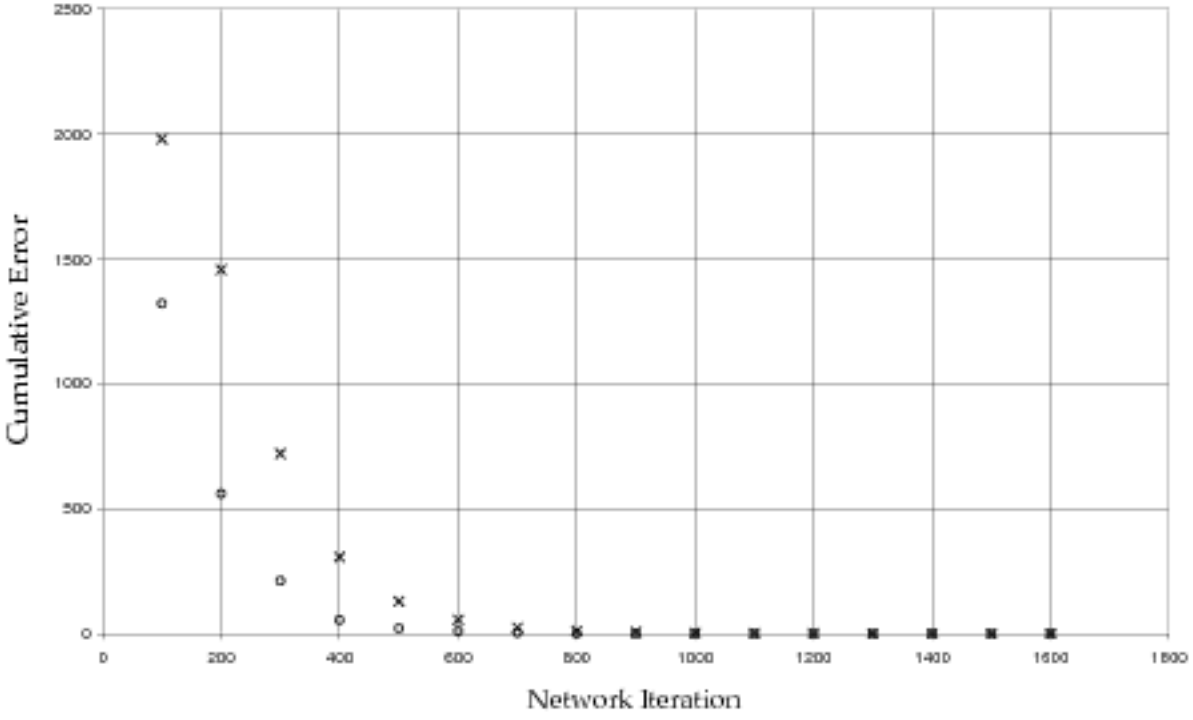
[a] The inverse error gradient is the reciprocal of the error gradient, ie $(\delta\varepsilon/\delta w_j)|w_{j=i}$

[b] That is to say that over 112 program runs using different carcinogen/mutagen data sets, the novel system of backpropagation arrived at the minimum error on average 11% faster than its predecessor.

In order to test the efficiency of the new proposed system of backpropagation, two networks, one using standard backpropagation and one using the proposed form, were compared using the same test data (Iris), where a mild increase in efficiency was apparent. More importantly, a comparison plot of averaged error over 112 runs, against iteration number is shown in Figure 4.2 using the TD_{50} carcinogen data. In this case, there is an important increase in network efficiency, which reduces processing times.

This system is quicker and more versatile to train: it can therefore deal with complex data and hidden layers more quickly. It is clear that the relationship between output and input is important in backpropagation. Therefore the transfer function has to be considered in the selection of the backpropagation system: the two must be considered a paired system.

Fig 4.2 – The variation of average error as a function of iteration number using the standard (crosses) and proposed (circles) backpropagation systems.



4.1.2 Transfer Functions

In choosing a transfer function, it is necessary to balance the two aspects of training which can influence its efficiency. On the one hand, it is important to use a transfer function which quickly adjusts the weights to minimise the error. However, on the other hand, it is equally important to realise that several of the transfer functions discussed in Chapter 2 make use of exponential functions, which are expensive computationally to evaluate. It should also be borne in mind that as well as requiring the use of a transfer function in processing each layer in the network, in each instance of backpropagation it is necessary to reverse that effect. That is to say for each exponential function used as part of a

transfer function there must be a complementary logarithmic function used in backpropagation. These also require a high computational expense.

Short *ad hoc* experiments were conducted to decide which of the three recommended transfer functions should be used. Linear transfer functions require no exponentials to be calculated, sigmoidal transfer functions require four exponentials to be calculated, and bicentral ones require sixteen!

The result of these short trials [See Table 4.1] was that, in the case of the Oral Rat TD_{50} data and the TOXNET mutagen data, though the bicentral transfer function was mildly better than the sigmoidal transfer function, the latter was found to provide the better trade-off with time. In the case of the Cl^- Transport Inhibition data, a sigmoidal transfer function performed best in the initial gridsearches. In subsequent tables, the underlined figures indicate the parameter/architecture used. Figure 4.3, for interest, compares threshold and sigmoidal transfer functions.

Fig 4.3 – a threshold and a sigmoidal transfer function.

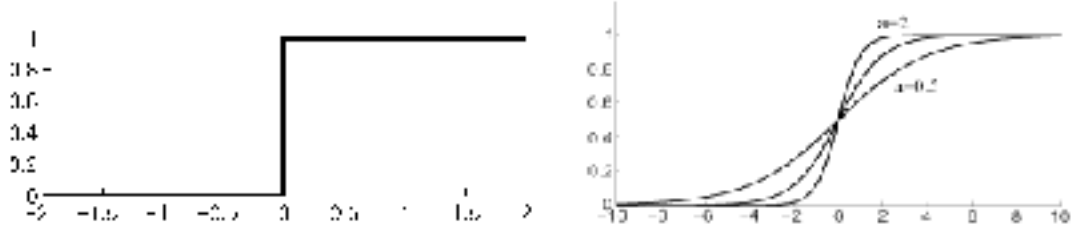


Table 4.1 – q^2 leave-20%-out cross-validation values for different transfer functions after 1,000 cycles

Transfer Fn.	*Iris Data*	*Oral Rat TD50 data*	*Cl^- Transport Inhibition Data*	*TOXNET Mutagen Data*
Linear	0.97	0.12	0.21	0.11
RBF (γ=1)	0.98	0.24	0.31	0.17
RBF (γ=10)	0.97	0.24	0.35	0.19
RBF (γ=50)	0.94	0.25	0.31	0.26
RBF (γ=100)	0.84	0.25	0.28	0.21
Polynomial (2)	0.97	0.20	0.25	0.26
Polynomial (4)	0.97	0.31	0.34	0.27
Polynomial (6)	0.97	0.32	0.36	0.24
Sigmoidal	0.98	0.36	0.45	0.32
Bicentral	0.98	0.37	0.43	0.35

where q^2 = number of correctly-banded data points/ total number of data points

4.1.3 Modes of Computation

There are four ways in which the *ANN* written for the purposes of this thesis may be used to analyse data:

1) Train a neural network from scratch with data generated mathematically.
2) Train a neural network from scratch with data lifted from a file.
3) Resume training a neural network from previously saved weights.
4) Test a series of data from a file and save the relevant outputs.

A simple menu system meeting these needs was included.

4.1.4 Data Management

It was decided that input data would be lifted from text files produced in the data management stage. These files would include a name, the number of

layers, the number of nodes in each layer, the number of training samples and the data themselves (each series of inputs, followed by a series of outputs). These files were given the suffix ".para". When satisfactory weights, giving a low average error, were found, they should be saved similarly in text files, given the suffix ".data". The network saves data as it processes to prevent their loss. Following completion of one percent of the chosen maximum number of iterations, whether or not the weights have become better is ascertained. If the weights have improved then they are saved. Testing data are saved in text files, similar to training data, but these have the suffix ".test".

It was decided that a system of reporting needed to be implemented to allow the user to gauge the success or otherwise of the network: every one percent of chosen maximum number of iterations to be performed seemed to suit the time periods involved.

Short *ad hoc* experiments were also used to set arbitrary variables contained within the transfer function. In order to prevent number errors in the ANN, such as dividing by zero or taking the logarithm of zero, it was necessary to place limits on the values which could be present: between -500 and +500, and data, if necessary, were scaled to within these limits.

It was decided to incorporate a system where, if an error were not decreasing over a series of reports (provided every 1% of the chosen maximum number of

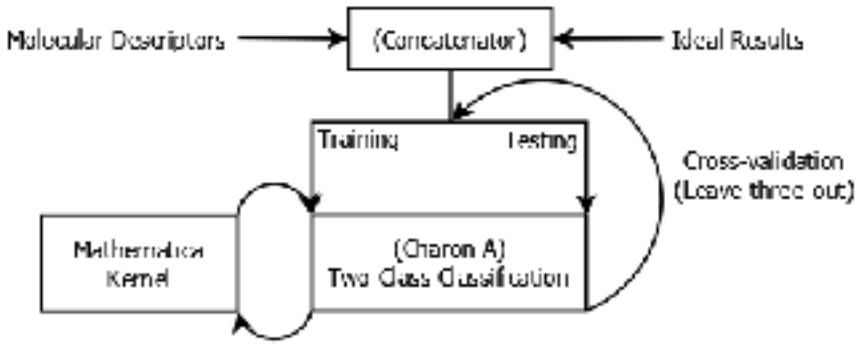
iterations to be processed), then the system would restart automatically, randomising the weights, but keeping the lowest average error as the benchmark by which subsequent networks are judged. Each time an automatic restart occurred extra latitude would be given to each network to come to a superior value of the output error.

However, in practice random restarts were found to occur even when the network seemed to be improving at a rate likely to beat the best average error and so an extra criterion was added which forced the learning rate (or leniency) to be less than a certain value, which occurs as the network reaches the bottom of a local minimum.

In conclusion, architecture of an ANN is extremely important, and there are many options to choose from. This is less true in Support Vector Machines, where fewer choices are necessary.

4.2 Support Vector Machines

Figure 4.4 – A data flow diagram for the Support Vector Machines



The molecular descriptors were likewise used in a series of Support Vector Machines, to discriminate between each carcinogen band and the other bands, in turn. In the case of the TD_{50} carcinogen data, nine SVMs were used running simultaneously (on parallel threads) on a quad-core machine, to improve processing times (one thread for the comparisons of each band).

4.2.1 Comparison of commercial programs against home-made versions.

Home-made SVMs were all outperformed by Wolfram Research's Mathematica[98] (v 7.0.1.0), enabled with a specially-adapted SVM Toolkit[99] However, Mathematica routinely suffered virtual memory obstacles and terminated, so I wrote a Java program (SVMOverseer), which fed data direct into the Mathematica kernel through another bespoke Java program (Charon A2). This allowed multiple Support Vector Machines to run simultaneously,

while SVMOverseer optimised the user-set parameters to better reflect the network architecture.

Table 4.2 – Time required (s) to process Support Vectors by program

Seconds	*Iris Data*	*Oral Rat TD50 data*	*Cl⁻ Transport Inhibition Data*
Charon A 1.1	60	14437	15241
Mathematica	25	12784	11247

4.2.2 User-defined parameters and architectures

In Non-Separable Support Vector Machines (which involve complex data) there are three broad parameters or architecture paradigms which have to be proposed by the user[100]. In setting them, we have to some extent assumed that they are independent of one other, a supported[101] assumption, and so only a one-dimensional grid search for each parameter was necessary.

First, *P* is a penalty constant which defines how much empirical errors are penalised in training. On the one hand, a higher *P* imposes a larger penalty on misclassified data and so creates fewer support vectors and a smaller separating region. On the other a lower *P* sets a smaller penalty for the same error and creates a larger margin. It is desirable to acquire as wide a separating margin as possible to reduce classification error, however it is necessary to do so without sacrificing accuracy or unreasonably increasing the processing time. Setting this value can severely affect the efficacy of the resulting model, and is the subject of

a short study for each series of data (see Table 4.3). On the basis of these studies a value of 2 was selected for Iris and Oral Rat TD_{50} data, a value of 1 was selected for Cl^- Transport Inhibition Data, and a value of 10 for the TOXNET Mutagen Data.

Table 4.3 – q^2 leave-20%-out cross-validation values for different P

P	*Iris Data*	*Oral Rat TD_{50} data*	*Cl^- Transport Inhibition Data*	*TOXNET Mutagen Data*
0.1	0.71	0.11	0.37	0.31
0.5	0.80	0.12	0.51	0.45
1	0.81	0.21	0.61	0.47
2	0.94	0.42	0.49	0.52
5	0.93	0.23	0.31	0.51
10	0.91	0.11	0.31	0.63
30	0.84	0.11	0.24	0.61

Secondly, ε. For SVM, ε is the ε-insensitive loss function. ε affects the smoothness of the margins and the number of support vectors: therefore, both the complexity and the generalisation capacity of the network depend on ε. There is also a connection between the noise observed in the training data and the ε-value. On the basis of the grid-search in Table 4.4, an ε of 0.1 was selected for the Iris Data and the Cl^- Transport Inhibition Data, 1 for the Oral Rat TD_{50} data, and 10 for the TOXNET Mutagen Data.

Table 4.4 – q^2 leave-20%-out cross-validation values for different ε

ε	*Iris Data*	*Oral Rat TD_{50} data*	*Cl^- Transport Inhibition Data*	*TOXNET Mutagen Data*
0.001	0.90	0.34	0.60	0.46
0.1	0.94	0.37	0.61	0.60
1	0.84	0.42	0.61	0.62
10	0.81	0.41	0.60	0.63

Finally, it is necessary to decide which Kernel Function to select in order to map the data into a higher-dimensional space where they are (hopefully) separable. Here again, other than the identity kernel there are two broad choices: a Radial Basis Function (RBF) or a Polynomial Function. The function which will most ably approximate the data in a higher-dimensional feature space might be drawn from either of these types, and a better Kernel Function results in a better separation of hyperplanes. This too was the basis of a short study for each series of data [see Table 4.5]. On the basis of this, a quadratic polynomial kernel was selected for the Iris Data, a 6th degree polynomial kernel for the Oral Rat TD_{50} data and the TOXNET Mutagen Data, and a 4th degree polynomial for the Cl^- Transport Inhibition Data.

Table 4.5 – q^2 leave-20%-out cross-validation values for different kernels

Kernel	*Iris Data*	*Oral Rat TD_{50} data*	*Cl^- Transport Inhibition Data*	*TOXNET Mutagen Data*
RBF (γ=1)	0.74	0.12	0.41	0.27
RBF (γ=10)	0.82	0.14	0.45	0.29
RBF (γ=50)	0.84	0.23	0.52	0.30
RBF (γ=100)	0.84	0.29	0.56	0.39
Polynomial (2)	0.94	0.21	0.45	0.41
Polynomial (4)	0.94	0.36	0.61	0.52
Polynomial (6)	0.94	0.42	0.61	0.63

4.2.3 Data Management

In the case of, for example, the Fisher Iris Data, descriptors are fed into the Mathematica Kernel in the following format:

```
X={{4.6,3.1,1.5,0.2},{5,3.6,1.4,0.2},{5.4,3.9,1.7,0.4},{4.6,3.4,
1.4,0.3},{5,3.4,1.5,0.2},{4.4,2.9,1.4,0.2},{4.9,3.1,1.5,0.1},{5.
```

```
4,3.7,1.5,0.2},{4.8,3.4,1.6,0.2},{4.8,3,1.4,0.1},{4.3,3,1.1,0.1}
,{5.8,4,1.2,0.2},
...
{6.1,3,4.6,1.4},{5.8,2.6,4,1.2},{5,2.3,3.3,1},{5.6,2.7,4.2,1.3},
{5.7,3,4.2,1.2},{5.7,2.9,4.2,1.3}};
```

Iris species, as follows:

```
Y={1,1,1,1,1,1,1,1,1,1,1,1... -1,-1,-1}
```

A kernel is selected, and a τ value. The Separable SVM is then activated.

```
a=SeparableSVM[X,Y,τ]
t=0.05;
p=PolynomialKernel[#1,#2,2]&;
```

Once the SVM has been processed, it is possible for it to classify a new data point

```
SVMClassify[p,X,a,Y,{5.7,2.8,4.1,1.3}]
```

as

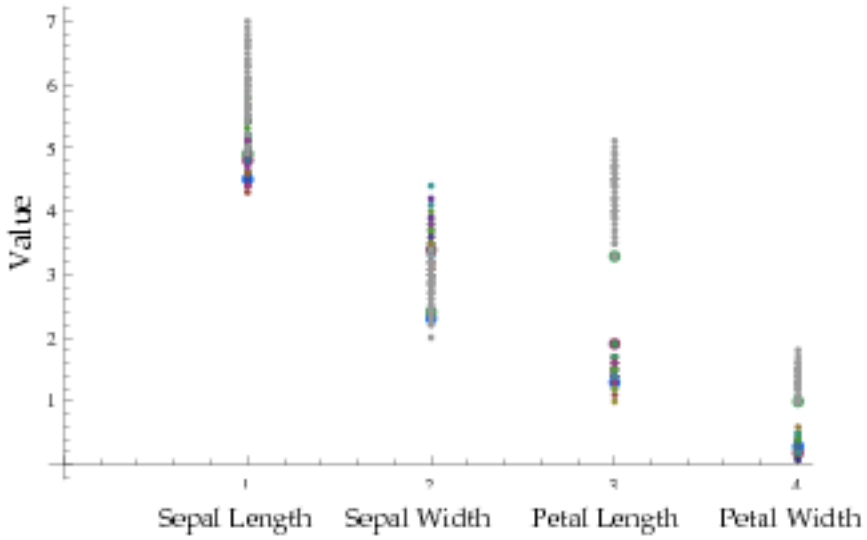
```
-1
```

Indeed the data for these iris varieties can then be plotted

```
SVMPlot[a,X,y,KernelFunction->p]
```

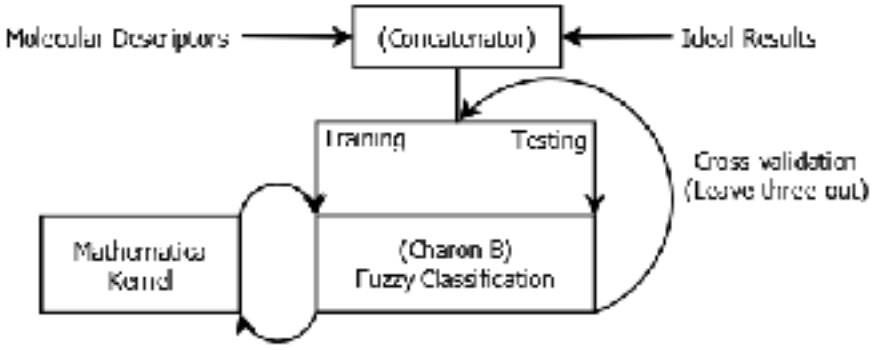
Figure 4.5 is an excellent visualisation of how the Iris data is complex, and cannot be solved by a simple classifier.

Fig. 4.5 – SVM Plot of Fisher Iris Data



4.3 Fuzzy Support Vector Machines

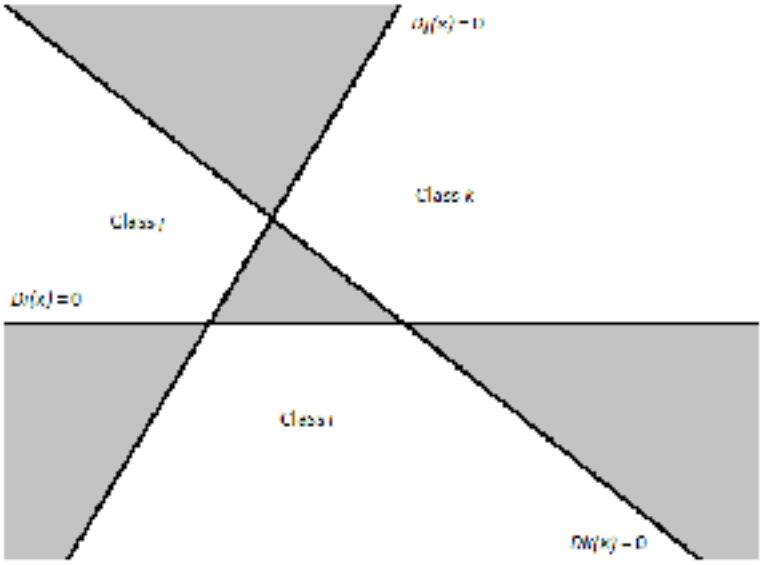
Figure 4.6 – A data flow diagram for the Fuzzy Support Vector Machines



Support Vector Machines obviously possess one critical flaw: they are designed to be binary, two-class separators. There is, therefore, a problem in extending a two-class technique to n-class problems. Conventionally, an n-class problem is divided into n two-class problems. The decision function of the ith problem, $D_i(x)$, divides class i from the other classes. Datum x is placed in class i when $D_i(x) > 0$.

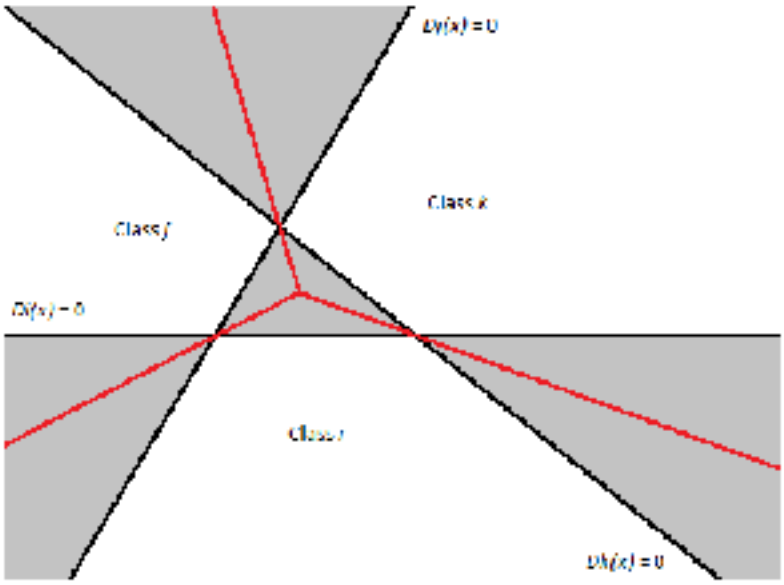
However, how should we deal with cases where no $D_i(x)$ is positive, or, indeed, where more than one is (*viz.* the datum is claimed by no class or by more than one)? This problem was solved in the normal Support Vector Machine through the use of the system propounded in Kreßel[102], where a pairwise system, which determines $n(n-1)/2$ two-class decision functions, is used. However, this system still produces unclassifiable regions, where several $D(x_i)$ have the same values.

Figure 4.7 – Some regions cannot be classified using the two-class classifier.



This renders the consideration of Fuzzy Support Vector Machines inescapable. Therefore a series of Fuzzy Support Vector Machines were created, with polyhedral pyramid as a membership function, as proposed in Abe[103].

Figure 4.8 – The class boundaries with polyhedral pyramidal membership functions (red).



It is then easy to classify new data, through the use of the following rules:

1. If $D_i(x) > 0$ is satisfied for only one class, that is the class of x.
2. If $D_i(x) > 0$ is satisfied for more than one class, x is a member of whichever class has the largest $D_i(x)$.
3. If $D_i(x) > 0$ is satisfied for no classes, x is a member of whichever class has the smallest absolute value of $D_i(x)$.

The SVMs were once again processed using a modification of Charon, known as Charon B (see Appendix), which interacted with the Mathematica Kernel to process the Support Vectors. These results were compared independently using Vanderbei's[104] code. I made the reasonable assumption that such complex data (all three sets) would not be linearly separable. As with normal Support Vector Machines, it was necessary to undertake grid search tests to ensure that parameters and architectures were selected that ensured speedy and accurate classification. I thought that it would be important to find these anew rather than rely on the same values as for the normal SVM in case there were substantial changes to be made.

On the basis of Table 4.6, I selected a value of P of 2 for the Iris Data, of 3for the Oral Rat TD_{50} data, of 3 for the Cl^- Transport Inhibition Data, and 3 for the TOXNET Mutagen Data.

Table 4.6 – q^2 leave-20%-out cross-validation values for different values of P.

P	*Iris Data*	*Oral Rat TD₅₀ data*	*Cl⁻ Transport Inhibition Data*	*TOXNET Mutagen Data*
0.5	0.81	0.15	0.63	0.42
1	0.83	0.17	0.65	0.48
2	0.83	0.26	0.67	0.49
3	0.80	0.30	0.67	0.51
4	0.80	0.30	0.61	0.45
5	0.80	0.29	0.54	0.42
10	0.78	0.14	0.51	0.40

Likewise, on the basis of Table 4.7, I used a quadratic polynomial kernel for the Iris Data, a sextic polynomial for the Oral Rat TD_{50} data, a quartic polynomial for the Cl^- Transport Inhibition Data, and an sextic polynomial for the TOXNET Mutagen Data.

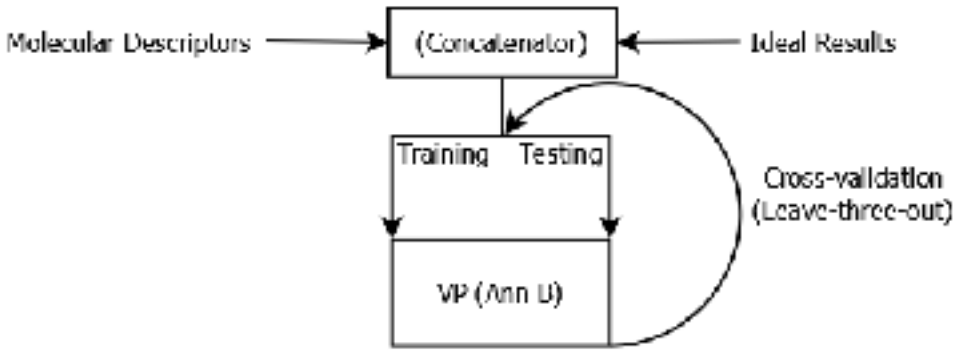
Table 4.7 – q^2 leave-20%-out cross-validation values for different kernels.

Kernel	*Iris Data*	*Oral Rat TD₅₀ data*	*Cl⁻ Transport Inhibition Data*	*TOXNET Mutagen Data*
Polynomial (2)	0.83	0.19	0.45	0.29
Polynomial (3)	0.83	0.22	0.65	0.31
Polynomial (4)	0.83	0.27	0.67	0.35
Polynomial (6)	0.84	0.30	0.68	0.51

In order to properly compare and assess these three principal techniques, it was necessary to pit them against a few tried-and-tested methods of pattern classification.

4.4 Voted Perceptrons

Figure 4.9 – A data flow diagram for the Voted Peceptron System



A system of Voted Perceptrons was created with a polynomial kernel function. This function was found to model the data best, with a degree of 3 for the Iris data and the Cl^- Transport Inhibition Data, and 4 for the Oral Rat TD_{50} data and the TOXNET Mutagen Data [see Table 4.8].

Table 4.8 – q^2 leave-20%-out cross-validation values for different polynomial kernels

index	*Iris Data*	*Oral Rat TD_{50} data*	*Cl^- Transport Inhibition Data*	*TOXNET Mutagen Data*
No kernel	0.41	0.14	0.31	0.13
2	0.62	0.21	0.51	0.20
3	0.95	0.31	0.57	0.29
4	0.95	0.35	0.57	0.42
5	0.95	0.35	0.57	0.43
6	0.95	0.35	0.57	0.43
7	0.95	0.36	0.57	0.43

Table 4.9 – q^2 leave-20%-out cross-validation values for different values of T

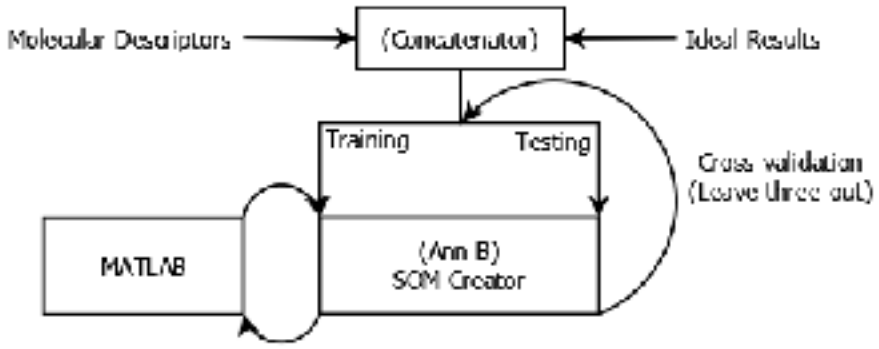
T	*Iris Data*	*Oral Rat TD_{50} data*	*Cl^- Transport Inhibition Data*	*TOXNET Mutagen Data*
100	0.94	0.32	0.51	0.27
500	0.95	0.35	0.57	0.31
1000	0.95	0.46	0.58	0.33
2000	0.95	0.47	0.59	0.33
3000	0.95	0.47	0.59	0.35
4000	0.95	0.47	0.59	0.39
6000	0.95	0.47	0.59	0.43
10000	0.95	0.47	0.59	0.44

The selection of T, the number of epochs, was also the subject of an optimisation test [see Table 4.9]. On the basis of this, T was set at 1000 for the Iris data, 2000 for the Oral Rat TD_{50} data and the Cl^- Transport Inhibition Data, and 6000 for and the TOXNET Mutagen Data.

The Voted Perceptron Data Management, Menu Systems, and Data Formats were often identical to that of the ANN, which formed the basis of the VP, following modification to the new architecture. As many of the facilities, problems and solutions were identical to that of the ANN, I will not repeat them here, not least as this is only a comparator technique.

4.5 Self-Organising Maps

Figure 4.10 – A data flow diagram for the Self-Organising Map



It was also important, when using supervised learning, to see whether unsupervised learning might be another, beneficial comparator to the ANN; so, a system of SOMs were also introduced to the problems.

In common with the ANN, the map was initialised with random weights, and training was performed by feeding the map a random input vector, calculating its Euclidian distance from each output vector and declaring the closest the best. That output vector's weight was then adjusted so it more closely resembled that of the input vector. Weights of those vectors around the best output vector were also adjusted to a lesser extent the further they were from the best output vector. The formula which was used to update the weights was

$$w_i(t+1) = w_i(t) + \alpha(t)h(t,d_{bj})[x_i(t) - w_i(t)] \qquad [4.5]$$

where $w_j(t+1)$ is the weight vector for the next cycle, $\alpha(t)$ is the learning rate (in common with supervised learning, the learning rate decreases over the life of the run), and $h(t,d_{bj})$ is the neighbourhood function (based on δ_{bj}, the Euclidean distance between the winning vector and the current vector). This process was then repeated until the map altered little between cycles.

The SOMs were written in MATLAB (The MathWorks Inc), using the SOM toolbox (MATLAB is the simplest way to code SOMs; its code is terse and the SOM toolbox is excellent) and trained sequentially. Optimisation of the user-defined parameters was completed using the *qe* (the Quantisation Error),

$$qe = \frac{1}{m}\sum_{j=1}^{m}\left|x_j - w(x_j)\right| \qquad [4.6]$$

where m is the number of inputs and $w(x_j)$ is the best output for input x_j.

There are several important user-defined parameters which were optimised:

i. The size and shape of map,

ii. The learning rate function,

iii. The initial learning rate,

iv. The neighbourhood radius,

v. The number of training cycles,

vi. The neighbourhood function (analogue of transfer function).

Neurons can be arranged on a map either as a rectangular or hexagonal lattice. I have used a hexagonal lattice as it does not favour those cells immediately adjacent to it; all six neurons closest to the best output are equidistant. Map size is important as if there are too few, then it will not be flexible enough to identify a wide range of solutions; if there are too many it will take too long to process and the generalisation ability will decrease. Generally we use a SOM with three times the number of map units as training samples.

Grid-searches were undertaken to assign user-defined parameters, and the results can be seen in Tables 4.10 to 4.16. In every case, the SOM was initialised with the same weights, which had been selected after trials of ten sets of random weights, to ensure a fair comparison. On the basis of this maps were generated of size 19x20 for the Iris data, 41x42 for the Oral Rat TD_{50} data, 24x24 for the Cl^- Transport Inhibition Data, and 25x25 for the TOXNET Mutagen data.

Table 4.10 - *qe* values for different map sizes

Size	*Iris Data*
19x19	1.27×10^{-6}
19x20	$\underline{1.18\times10^{-6}}$
19x21	1.21×10^{-6}
20x20	1.47×10^{-6}
20x21	1.23×10^{-6}
21x21	1.45×10^{-6}

Size	*Oral Rat TD_{50} data*
27x28	4.18×10^{-4}
28x28	4.14×10^{-4}
28x29	4.15×10^{-4}
29x29	4.13×10^{-4}
29x30	$\underline{4.12\times10^{-4}}$
30x30	4.14×10^{-4}

Size	*Cl^- Transport Inhibition Data*
22x23	5.91×10^{-6}
23x23	5.78×10^{-6}
23x24	5.79×10^{-6}
24x24	$\underline{5.73\times10^{-6}}$
24x25	5.78×10^{-6}
25x25	6.02×10^{-6}

Size	*TOXNET Mutagen Data*
24x25	0.414
25x25	<u>0.409</u>
25x26	0.457
26x26	0.481
26x27	0.465

Three different learning rate functions can be used in MATLAB: Linear, Reciprocal, and Exponential. As you can see from Table 4.11, Linear learning rate functions have the best effect for each data set.

Table 4.11 - *qe* for different learning rate functions with an initial learning rate of 0.5.

Learning Rate Fn.	*Iris Data*	*Oral Rat TD₅₀ data*	*Cl⁻ Transport Inhibition Data*	*TOXNET Mutagen Data*
Linear	1.14×10^{-6}	4.12×10^{-4}	5.73×10^{-6}	0.409
Reciprocal	1.75×10^{-5}	7.40×10^{-4}	5.41×10^{-4}	0.956
Exponential	0.237	0.412	1.62×10^{-3}	5.415

In line with the findings of Tan and George[105] the neighbourhood size should be approximately half the value of the map diameter. It and the initial learning rate, α_0, should be considered part of a paired system, and so must be selected together. On this basis a pairing (Tables 4.12-4.15), respectively of initial learning rate to initial radius, of 0.5 and 9 were selected for the Iris data, 0.3 and 12 for the Oral Rat TD_{50} Data, 0.7 and 9 for the Cl^- Transport Inhibition Data, and 0.3 and 9 for the TOXNET Mutagen Data.

Table 4.12 - *qe* for different initial neighbourhood radii and learning rates for Iris Data

↓Initial Radius	*Initial Learning Rate→*	*0.1*	*0.3*	*0.5*	*0.7*	*0.9*
8		17.5	21.4	1.76×10^{-6}	1.26×10^{-4}	8.91×10^{-5}
9		9.45	5.02×10^{-3}	1.64×10^{-6}	7.21×10^{-5}	2.54×10^{-4}
10		0.158	2.09×10^{-4}	5.45×10^{-6}	8.16×10^{-5}	9.02×10^{-5}
11		24.5	$2.71.10^{-3}$	1.71×10^{-4}	0.169	4.12×10^{-5}

Table 4.13 - *qe* for different initial neighbourhood radii and learning rates for Oral Rat TD_{50} Data

↓*Initial Radius*	*Initial Learning Rate*→	*0.1*	*0.3*	*0.5*	*0.7*	*0.9*
10		1.12×10^{-3}	5.54×10^{-4}	7.14×10^{-4}	6.41×10^{-3}	8.45×10^{-3}
11		6.45×10^{-4}	4.76×10^{-4}	7.23×10^{-4}	9.54×10^{-3}	9.14×10^{-3}
12		7.19×10^{-4}	4.12×10^{-4}	9.87×10^{-4}	4.69×10^{-3}	9.05×10^{-3}
13		8.64×10^{-4}	7.68×10^{-4}	2.38×10^{-3}	6.94×10^{-3}	4.54×10^{-3}

Table 4.14 - *qe* for different initial neighbourhood radii and learning rates for Cl^- Transport Inhibition Data

↓*Initial Radius*	*Initial Learning Rate*→	*0.1*	*0.3*	*0.5*	*0.7*	*0.9*
8		2.45×10^{-4}	7.12×10^{-5}	4.36×10^{-5}	2.89×10^{-5}	5.45×10^{-5}
9		1.84×10^{-4}	6.54×10^{-5}	9.23×10^{-6}	5.73×10^{-6}	6.54×10^{-6}
10		3.56×10^{-4}	4.23×10^{-5}	5.62×10^{-5}	5.45×10^{-5}	7.91×10^{-5}
11		5.41×10^{-4}	9.85×10^{-5}	7.98×10^{-5}	8.64×10^{-5}	1.21×10^{-4}

Table 4.15 - *qe* for different initial neighbourhood radii and learning rates for TOXNET Mutagen Data

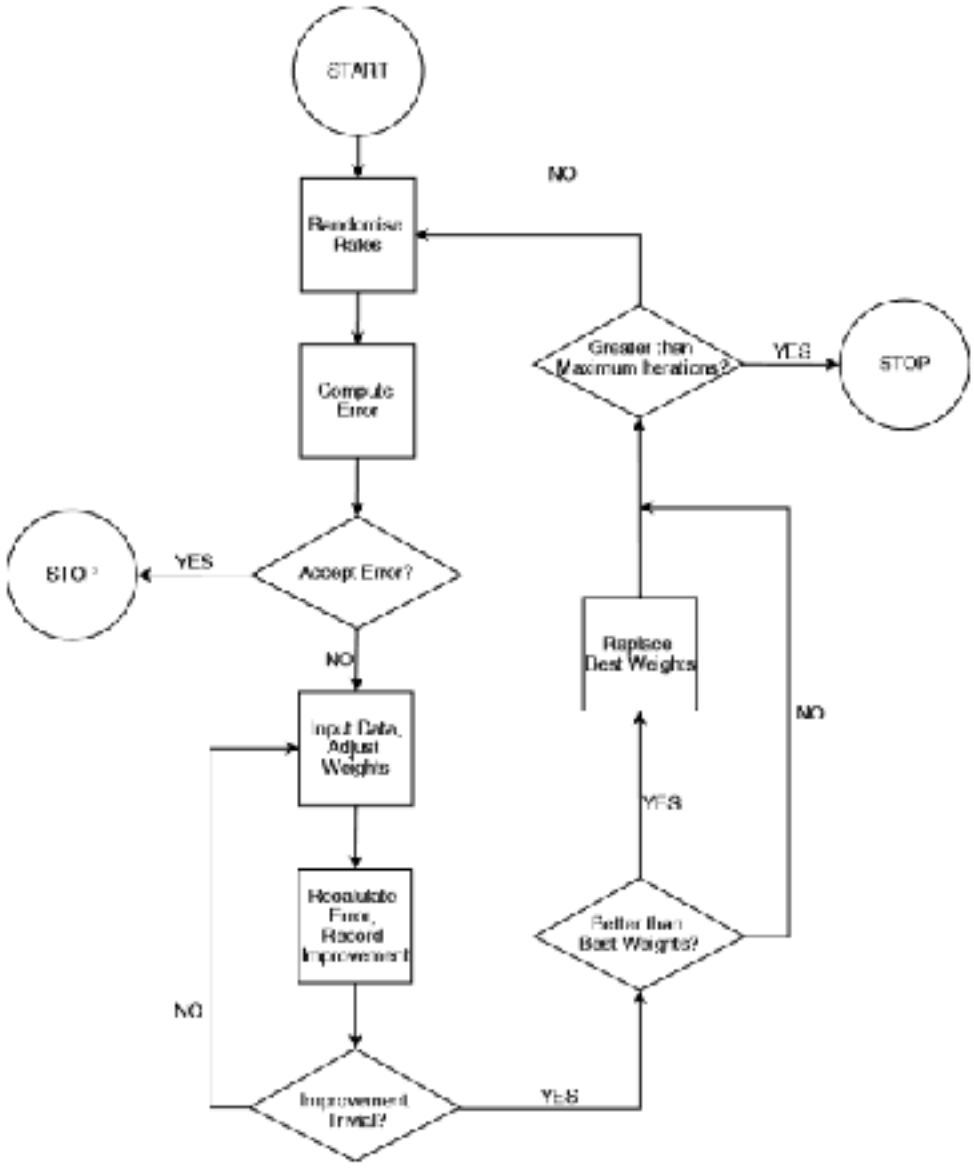
↓*Initial Radius*	*Initial Learning Rate*→	*0.1*	*0.3*	*0.5*	*0.7*	*0.9*
8		3.54	0.954	4.22	8.73	21.6
9		1.69	0.409	1.69	4.41	15.4
10		9.65	2.15	7.16	10.23	29.6
11		12.2	7.45	14.8	19.5	22.4

In line with Kohonen[106], the number of training cycles should be set at 500 times the number of the map units to produce a good map. Four neighbourhood functions are available in MATLAB: Gaussian, Cut Gaussian, Bubble and Epanechicov; the last of these was used for Iris and Oral Rat TD_{50} data and the TOXNET mutagen data, whereas Bubble showed the most promise for the Cl^- Transport Inhibition data (Table 4.16).

Table 4.16 - *qe* values for different neighbourhood functions

Neighbourhood Fn	*Iris Data*	*Oral Rat TD50 data*	*Cl⁻ Transport Inhibition Data*	*TOXNET Mutagen Data*
Gaussian	5.01	15.4	7.16×10^{-3}	2.59
Cut Gaussian	0.231	0.132	4.62×10^{-5}	2.41
Bubble	1.41×10^{-6}	3.56×10^{-8}	7.16×10^{-6}	1.68
Epanechicov	1.23×10^{-6}	9.45×10^{-9}	4.62×10^{-5}	0.409

Fig 4.11 – The decision tree for the random-restart SOM.

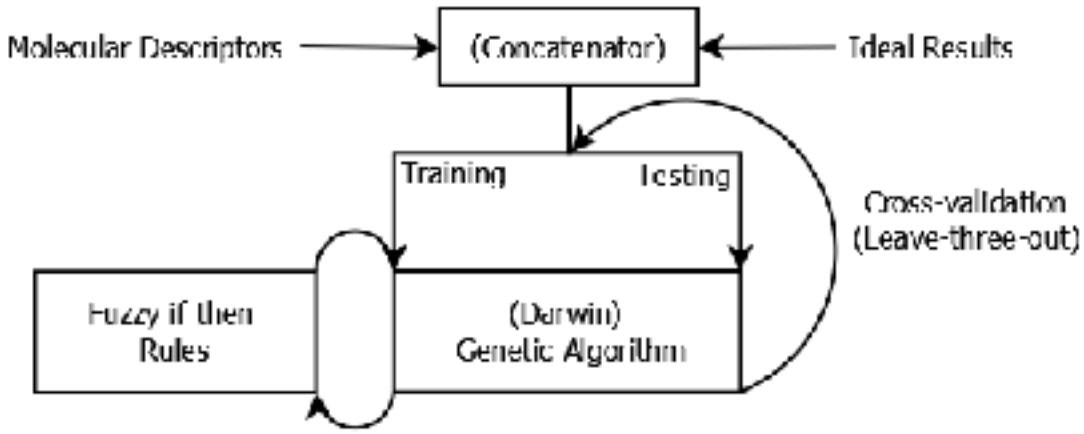


The Self Organising Maps were run using MATLAB and a variation of the MATLAB SOM Toolbox provided by Vesanto et al[107].

Having considered Self-Organising Maps, it is worth considering that Genetic Algorithms can also be useful for rapidly searching high-dimensional spaces to find optimum solutions to this sort of problem.

4.6 Genetic Algorithms

Figure 4.12 – A data flow diagram for the Genetic Algorithms



A series of Genetic Algorithms were created according to the precepts in Goldberg[108] and the tenets laid out in Chapter 2, first incorporating a method of selection for a small number of significant fuzzy rules to create a classification system with good classification ability, according to the system outlined in Russo[109] and extended in Chakraborty & Pal[110]. The rule optimisation problem is

represented as a combinatorial one. Its objectives are to maximise the number of correctly-banded data points and to minimise the number of fuzzy *if-then* rules.

A Genetic Algorithm was used to process the data. A series of fuzzy *if-then* rules was coded as a string and treated as an individual (phenotype) in the Genetic Algorithms. The GAs used Stochastic Universal Sampling to choose breeding pairs (which were combined to form children), and Decay Fitness to choose those samples that would mutate, by randomly altering how important each of the molecular descriptors was considered. Truncation was used to select *elite* individuals to pass direct to the next generation of the algorithm.

The variables were selected as follows:

1. Proportion of daughter generation created by asexual reproduction with mutation = 0.45;

2. Proportion of daughter generation created by sexual reproduction= 0.45;

3. Proportion of daughter generation copied directly from the previous generation (elitism) = 0.1;

4. Size of population = 100;

5. Number of cycles before elitism was used = 1000 (note that for the first 1000 cycles the proportion of daughter generation that was created by sexual and asexual reproduction was 0.5 for each).

The result was a series of Java programs which undertook the necessary computations. These were written more simply that the ANN, with fewer menus and less error-handling; variables were entered directly into the source before compilation. This made more sense than setting up menu systems, especially for what is only a comparator technique.

4.7 Leave-three-out Cross-validation

In order to compare techniques, descriptors and network architectures in a quantitative and scientific way, it is necessary to use some sort of cross-validation technique. This technique excels at comparisons between AI techniques, not least because it roots out over-fitting or under-fitting in ANNs. Cross-validation[111] is good at preventing testing hypotheses suggested by the data (Type III errors[112]), especially where further data is difficult to obtain. Leave-some-out (LSO) cross-validation works by setting aside a subset of the training data, completing the training using the remainder of the data, and then testing the system using the first subset. This process is then repeated with another subset of data set-aside, until all data have been set aside at least once. The average error is then calculated across all of the assays.

LOO (Leave-one-out) Cross-validation is computationally expensive, as each technique has to be run many times. However, setting aside too many examples can lead to ineffective cross-validation, which fails to identify errors in the technique. Thus a due medium between scientific laziness and computational profusion must be obtained. In this case it was judged that Leave-three-out Cross-validation would not take too long to compute, whilst still being scientifically justifiable.

4.8 Java programs associated with this thesis

A variety of home-made programs were necessary either to perform functions associated with this work, interact with mainstream programs, or acquire data.

1. *ANN A* is the Artificial Neural Network;
2. *Charon A* is the Support Vector Machine interface;
3. *Charon B* is the Fuzzy Support Vector Machine interface;
4. *Darwin* is the Genetic Algorithm;
5. *GRA* is the Voted Perceptron System;
6. *WebDataFinder* downloads and stores Mass Spectral data from NIST;
7. *WebDataFinder2* downloads and stores Mass Spectral data from NIST;
8. *WebDataFinder 3* downloads and stores Mass Spectral data from NIST;
9. *DataFinder* extracts carcinogen data from the NTP datasheets;
10. *Standardiser* bands Mass Spectral data into ranges;

11. *Cartwrightiser* produces tables of Mass Spectral data from individual files;
12. *FrequencyX (several programs)* identify the modal fragments, and tabulate their intensities;
13. *GamessFilter* ensures that the CML data has been treated by GAMESS;
14. *GamessFilter*2 ensures that the CML data has been treated by GAMESS;
15. *CMLFinder* standardises the Random Factory CML database ;
16. *Peter* performs Principal Component Analysis;
17. *ConcatenatorLog* concatenates the molecular descriptors with their activity values (which in the case carcinogen data are banded logarithmically.

References

[97] Priest A.C., Williamson A.J. and Cartwright H.M., The Application of Artificial Neural Networks in the Identification of Quantitative Structure-Activity Relationships for Chemotherapeutic Drug Carcinogenicity, *Lecture Notes in Computer Science* 6065, *Advances in Intelligent Data Analysis IX* 137-146 (2010).

[98] Wolfram Research, Mathematica: http://www.wolfram.com/mathematica/, accessed 04/11/2010.

[99] Mathematica SVM Toolkit http://www.mathematica-journal.com/issue/v10i1/SupportVectorMachines.html, accessed 01/11/2010.

[100] Inoue T., Abe S., Fuzzy support vector machines for pattern classification, *Proceedings of International Joint Conference on Neural Networks*, Washington DC, 1449–1454 (2001).

[102] Kreßel U., Pairwise classification and support vector machines, *Advances in kernel methods: support vector learning*, MIT Press, Cambridge, MA (1999).

[103] Abe S., Pattern Classification: Neuro-Fuzzy Methods and Their Comparison, Springer, London (2001).

[104] Vanderbei R.J., LOGO: An Interior Point Code for Quadratic Programming, Princeton University SOR-95-15 (1998).

[105] Tan H.S. and George S.E., Investigating learning parameters in a standard 2-D SOM model to select good maps and avoid poor ones, *Proceedings of Advances in Artificial Intelligence*, Springer, London, 425-437 (2004).

[106] Kohonen T, Self-organising maps, *Springer Series in Information Sciences* 30, Springer, Berlin (1995).

[107] Vesanto J., Himberg J., Alhoniemi E., and Parhankangas J., Self-organizing map in Matlab: the SOM Toolbox, *Proceedings of the Matlab DSP Conference 1999*, Espoo, Finland, Comsol Oy, 35-40 (1999).

[108] Goldberg, D.E., Genetic algorithms in search, optimization, and machine learning, Addison-Wesley, New York (1989).

[109] Russo, M., FuGeNeSys-a fuzzy genetic neural system for fuzzy modeling, *IEEE Transactions on Fuzzy Systems* 6 (3), 373 - 388 (1998)

[110] Chakraborty, D. Pal, N.R., A neuro-fuzzy scheme for simultaneous feature selection and fuzzy rule-based classification, *Neural Networks* 15 (1), 110 - 123 (2004).

[111] Kohavi, R., A study of cross-validation and bootstrap for accuracy estimation and model selection. C.S. Mellish (Ed.), *Proceedings of the 14th International Joint Conference on Artificial Intelligence* 1137-1140, Morgan Kaufmann (1995).

[112] Schwartz S., Carpenter K.M., The right answer for the wrong question: Consequences of type III error for public health research. *American Journal of Public Health* 89, 1175–1180 (1999).

Chapter 5

Raw, Graphical and Visual Results

Chapter 5 presents various new results, both in tables and visually; these experimental results are extremely interesting. It explains that no one AI technique dominated the field. In fact, given a new application, some previously-successful techniques fail spectacularly. Nor, it seems, does one system of molecular descriptors show greater fidelity than another - it very much depends on what activity of the chemical is being measured, and even (surprisingly) on which AI technique is being applied. Further explanation of why this is, in fact, logical can be found in the Chapter 6.

The results for the Fisher Iris data, whilst useful indicators of the capacity of the techniques, will tell us very little about their applications to more complex, biological data. They are provided for completeness.

Table 5.1– summary of cross-validation values, q^2, for Fisher Iris data

q^2	*Fisher Iris*
ANN	0.99
SVM	0.95
FSVM	0.95
GA	0.91
VP	0.93

The analysis in this Chapter tends to concentrate on the first three techniques (ANN, SVM, FSVM); it does so for two important reasons. First, they are the paradigms under principal examination, rather than those which are acting as comparators. Secondly, because, except in a few notable cases, they perform better than the comparator techniques. While it is interesting to ascertain why a

technique may have failed, it is more important for the furtherance of this area of research to explain the reasons why one succeeded.

5.1 Summary of Cross-Validation Values by Dataset

The cross-validation results for each of the AI techniques and molecular descriptor system, presented by dataset, were as follows.

Table 5.2 – A summary of cross-validation values, q^2, for carcinogenicity (TD_{50}) in rats

	PCAMS	*CoMFA*	*CoMSIA*	*HQSAR*	*EVA*
ANN	0.84	0.84	0.83	0.74	0.54
SVM	0.87	0.84	0.85	0.85	0.79
FSVM	0.88	0.84	0.85	0.85	0.82
GA	0.46	0.41	0.47	0.40	0.38
VP	0.85	0.81	0.79	0.78	0.69

Figure 5.2 – A graphical representation of Table 5.2; [No Figure 5.1]

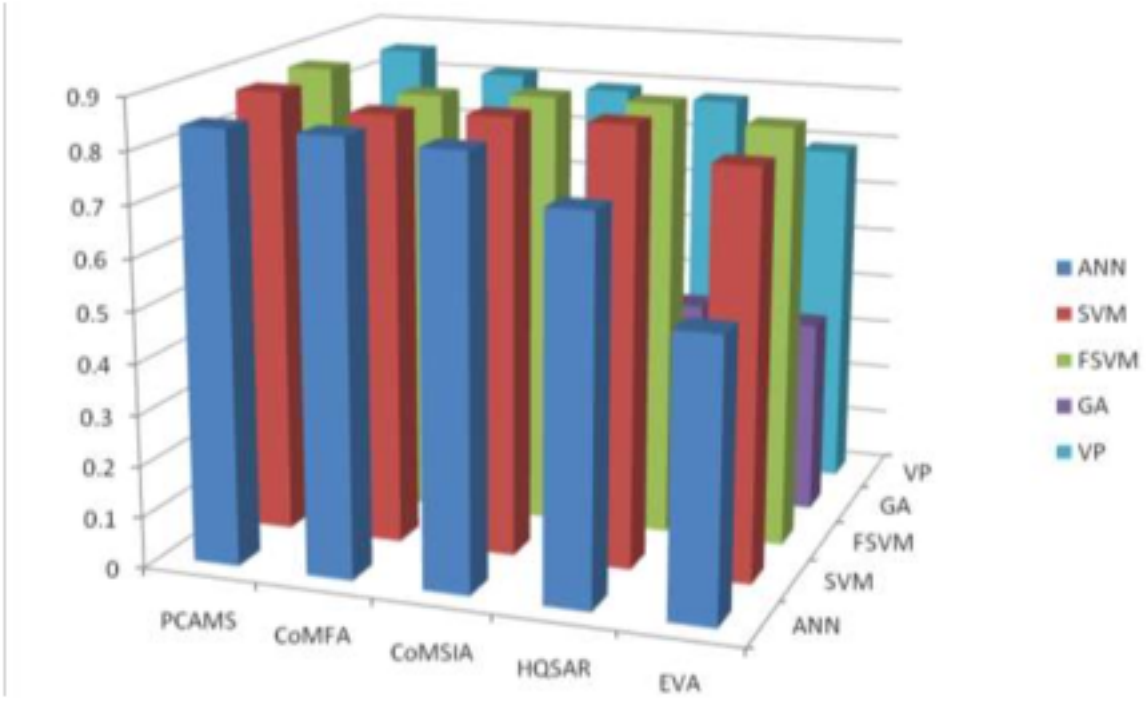


Table 5.3 – A summary of cross-validation values, q^2, for Human erythrocyte Cl^- transport inhibitors

	PCAMS	*CoMFA*	*CoMSIA*	*HQSAR*	*EVA*
ANN	0.64	0.55	0.56	0.54	0.67
SVM	0.74	0.61	0.66	0.54	0.68
FSVM	0.77	0.61	0.66	0.53	0.69
GA	0.63	0.59	0.51	0.58	0.60
VP	0.71	0.68	0.67	0.69	0.65

Figure 5.3 – A graphical representation of Table 5.3

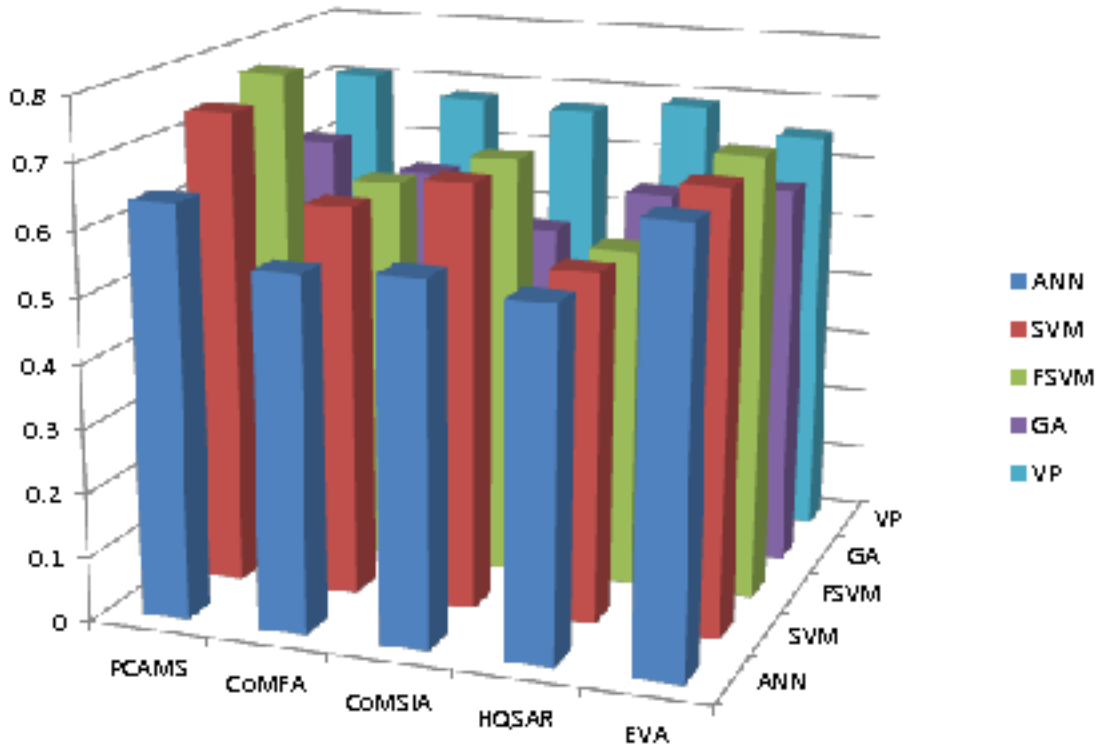
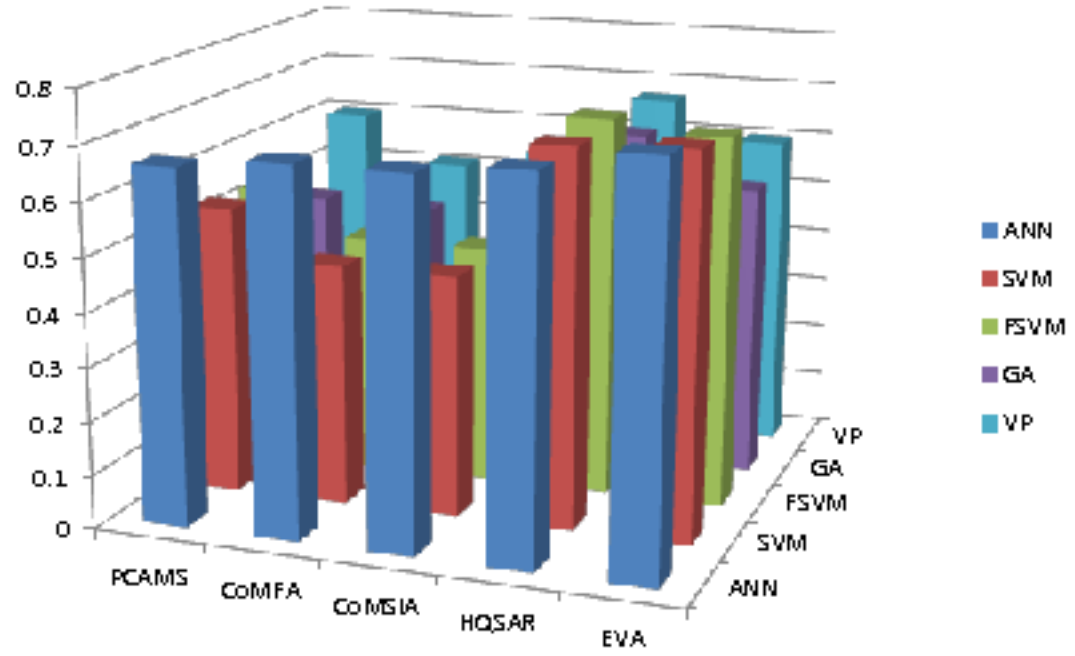


Table 5.4 – A summary of cross-validation values, q^2, for TOXNET mutagen data

	PCAMS	*CoMFA*	*CoMSIA*	*HQSAR*	*EVA*
ANN	0.66	0.68	0.68	0.70	0.74
SVM	0.54	0.45	0.45	0.70	0.71
FSVM	0.53	0.45	0.45	0.71	0.69
GA	0.47	0.46	0.47	0.64	0.55
VP	0.60	0.51	0.55	0.67	0.60

Figure 5.4 – A graphical representation of Table 5.4



5.2 Discussion: the AI Techniques

Artificial Neural Networks performed well across the board [Tables/Figures 5.2-5.4], and in the case of the TOXNET data performed best of all; indeed, it was very rare to see them perform poorly. The uniformly-good and sometimes-excellent capabilities of ANNs are one of the causes of the longevity of their popularity in the field of pattern recognition: they are to some extent reliable, if caution is taken to avoid over-fitting. However, note that this reliability is severely dependant on the molecular descriptor system used (by way of example, a very poor ANN using EVA in Table 5.2).

Whilst the majority performed well, the EVA descriptors generally scored poorly in this application, and so the failure of this combination (ANN/EVA) must derive from an underlying poor fidelity of the descriptor in this application.

While they may be thought reliable, ANNs were routinely matched or outperformed, if only marginally, by Support Vector Machines [Tables 5.2 and 5.3], when they were applied to the Carcinogen and Cl^- Transport Inhibition datasets. This mild improvement through the use of the SVM is well supported in the literature (see Chapter 2), and therefore comes as no surprise. Support Vector Machines did, however, fail to perform as well when applied to the fewer bands of the TOXNET mutagen data [Table 5.4]. This is perhaps a little surprising, especially as the TOXNET data were banded into the fewest bands (4), and it would be logical to assume that an ANN, which can produce continuous predictions, would perform better than the SVM (which must undertake pairwise binary prediction), with larger numbers of bands. However, it appears that the opposite is true. *Is the number of bands into which the output data is segregated, therefore, important?*

A similar picture exists for Fuzzy Support Vector Machines, which outperformed ordinary SVMs, in the first two cases, and performed a little worse in the last. This is a logical result: Fuzzy SVMs are better classifiers, as no

regions exist in which classification is not possible, but *why should they perform worse than generic SVMs when applied to the TOXNET data, with its fewer bands?*

Voted Perceptrons generally performed extremely uniformly, and generally very well, when applied to the different datasets, achieving predictive capabilities somewhere between the ANNs and the SVMs in each of the examples. Although reliable, they seem unable to compete with the precision obtained by ANNs and SVMs, and so, unless they could be precursors of more refined techniques which would give rise to better predictions, it seems that they largely can be discounted when considering potential applications in this area.

In contrast, the Genetic Algorithms performed very poorly indeed [ie q^2=0.47, for GA cf. q^2=0.87 for FSVM, when applied to the Carcinogenicity TD_{50} data in Table 5.2]. There are possibly many reasons for this. First, it may be an underlying problem with the technique itself. The Building Block Hypothesis, which underpins the technique, is often criticised by speculation that "the various claims about GAs that are traditionally made under the name of the *building block hypothesis* have, to date, no basis in theory and, in some cases, are simply incoherent" (Wright et al.[113]). Secondly, it may be just that a technique which seeks to discover fit answers within a large feature space is unsuited to the nuances in the complexity of this type of data.

5.3 Summary of Cross-Validation Values by Technique

Tabulation and visualisation of cross-validation data, sorted by AI technique, will enable us to judge the molecular descriptor systems' fidelity at representing molecules. The results were as follows.

Table 5.5 – A summary of cross-validation values, q^2, for ANN

	PCAMS	*CoMFA*	*CoMSIA*	*HQSAR*	*EVA*
Carcinogenicity (TD_{50}) in rats	0.84	0.84	0.83	0.74	0.54
Human erythrocyte Cl^- Transport inhibitors	0.64	0.55	0.56	0.54	0.67
TOXNET mutagens	0.66	0.68	0.68	0.70	0.74

Figure 5.5 – A graphical representation of Table 5.5

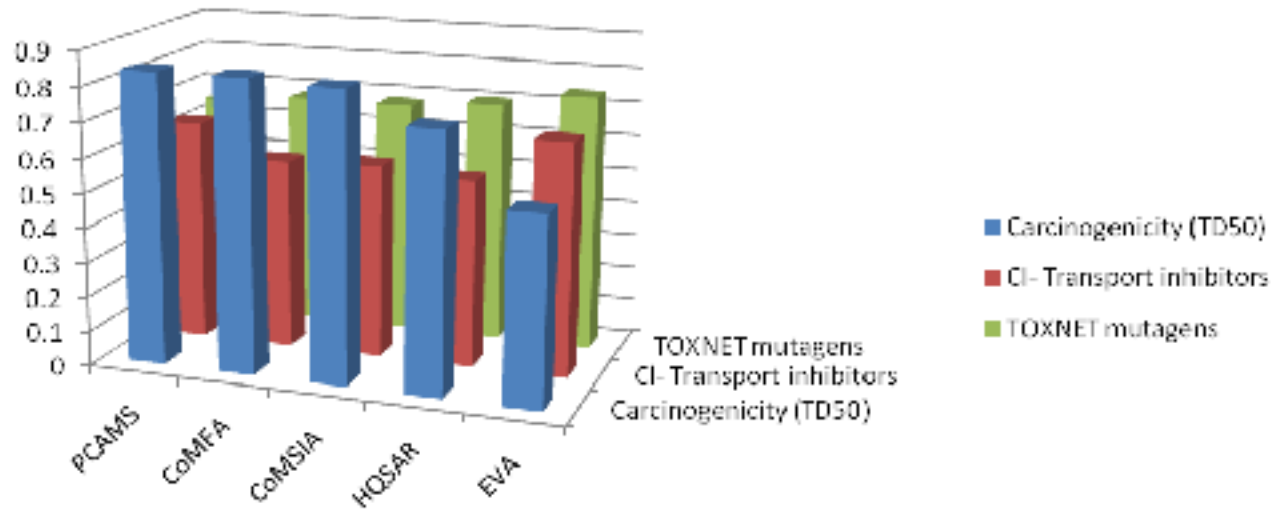


Table 5.6 – A summary of cross-validation values, q^2, for SVM

	PCAMS	*CoMFA*	*CoMSIA*	*HQSAR*	*EVA*
Carcinogenicity (TD_{50}) in rats	0.87	0.84	0.85	0.85	0.79
Human erythrocyte Cl^- Transport inhibitors	0.74	0.60	0.66	0.54	0.68
TOXNET mutagens	0.54	0.45	0.45	0.70	0.71

Figure 5.6 – A graphical representation of Table 5.6

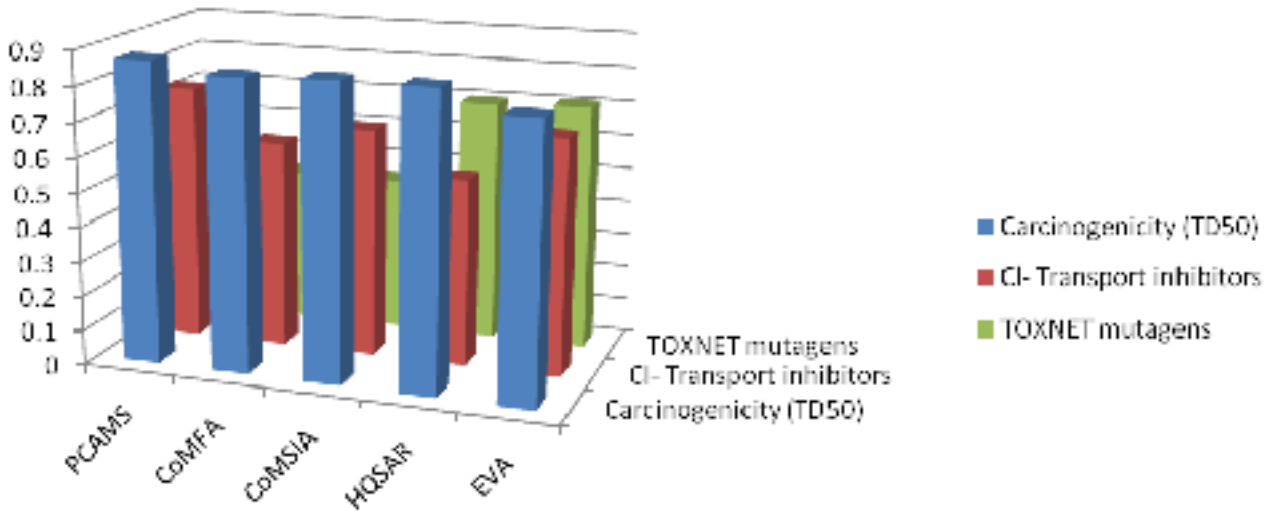


Table 5.7 – A summary of cross-validation values, q^2, for PSVM

	PCAMS	*CoMFA*	*CoMSIA*	*HQSAR*	*EVA*
Carcinogenicity (TD_{50}) in rats	0.88	0.84	0.85	0.85	0.82
Human erythrocyte Cl- Transport inhibitors	0.77	0.61	0.66	0.53	0.69
TOXNET mutagens	0.53	0.45	0.45	0.71	0.69

Figure 5.7 – A graphical representation of Table 5.7

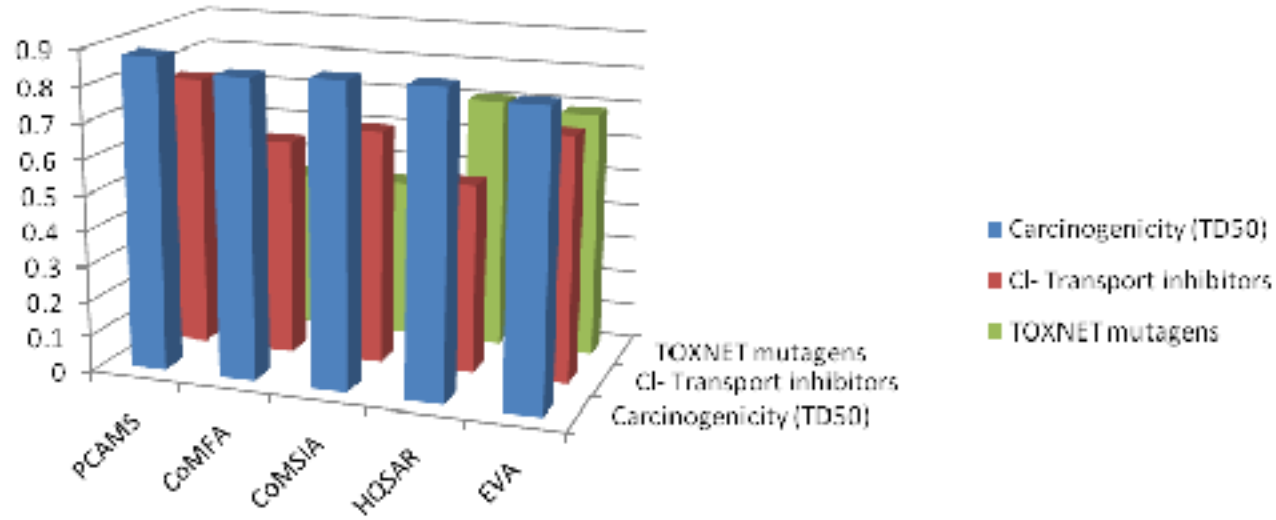


Table 5.8 – A summary of cross-validation values, q^2, for GA

	PCAMS	*CoMFA*	*CoMSIA*	*HQSAR*	*EVA*
Carcinogenicity (TD_{50}) in rats	0.46	0.41	0.47	0.40	0.38
Human erythrocyte Cl^- Transport inhibitors	0.63	0.59	0.51	0.58	0.60
TOXNET mutagens	0.47	0.46	0.47	0.64	0.55

Figure 5.8 – A graphical representation of Table 5.8

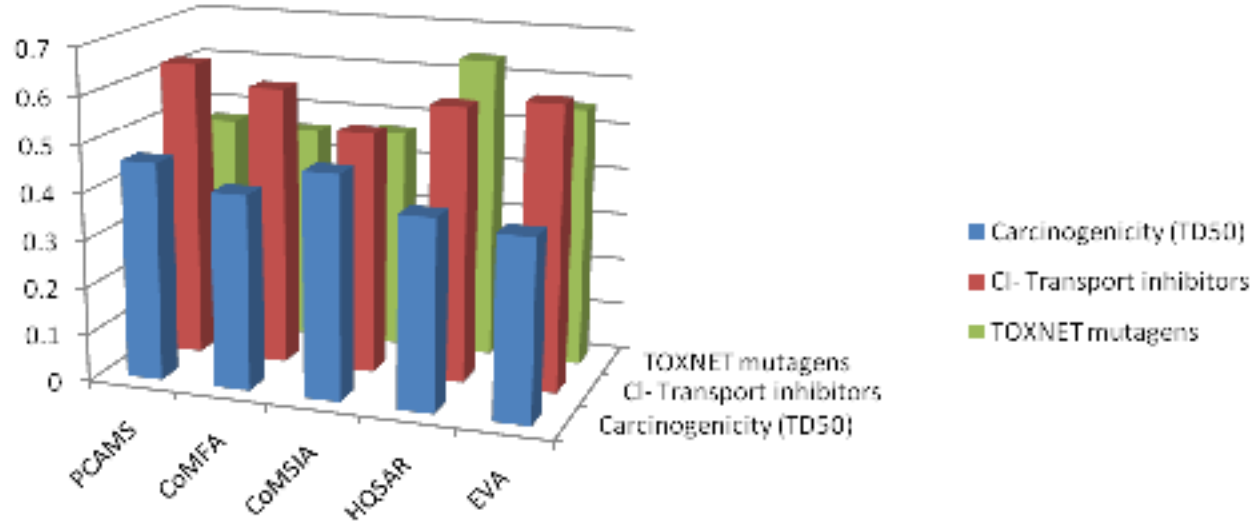
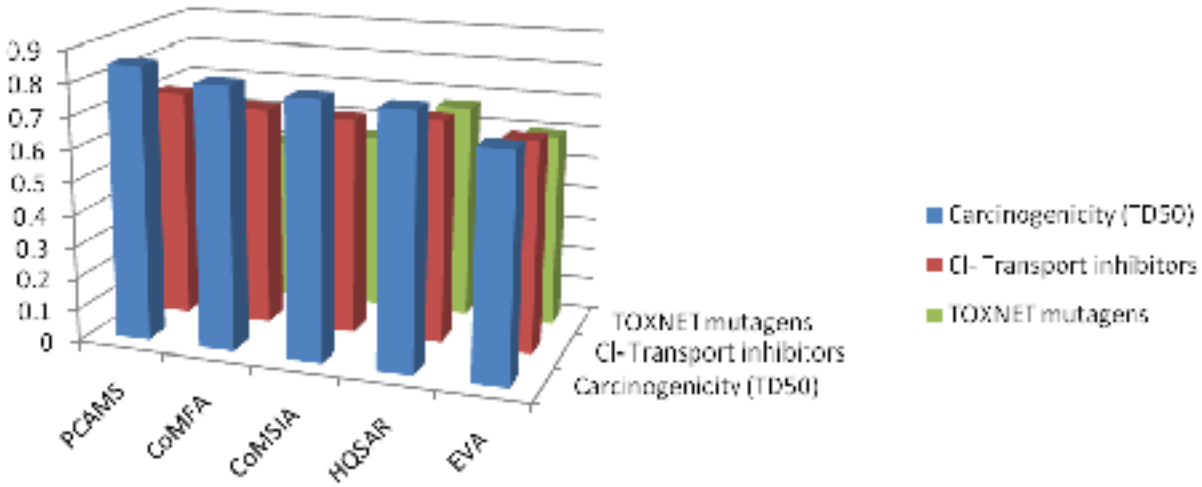


Table 5.9 – A summary of cross-validation values, q^2, for VP

	PCAMS	*CoMFA*	*CoMSIA*	*HQSAR*	*EVA*
Carcinogenicity (TD_{50}) in rats	0.85	0.81	0.79	0.78	0.69
Human erythrocyte Cl^- Transport inhibitors	0.71	0.68	0.67	0.69	0.65
TOXNET mutagens	0.60	0.51	0.55	0.67	0.60

Figure 5.9 – A graphical representation of Table 5.9



5.4 Discussion: the Molecular Descriptors systems

A great deal of disparity seems to exist between the various molecular descriptors' capacity to aid these sorts of predictions. Again, this is to be expected. One system of descriptors is more likely to be faithful at predicting a particular activity if its method of description concentrates on some physical property which underlies the activity under examination.

For example, in the case of the case of the Carcinogen TD_{50} data [Tables/Figures 5.2, 5.5-5.9], PCAMS, CoMFA and CoMSIA seem to have encouraged excellent predictions, HQSAR reasonable ones, and EVA poor. *What aspect of the first three techiques' configuration is it that makes them excel at this application?* It may well be, for example, that the steric and electrostatic effects, hydrophobicity, and

hydrogen bonding represented by CoMFA, rather than the vibrational frequencies of EVA, are better indicators of chemical disruptors of genome or cellular metabolic processes. It is apparent that hydrogen bonding ability must be an important issue, as CoMSIA-based predictions often performed a little better than those of CoMFA (hydrogen bonding donor and acceptor values being the principal addition in the former).

PCAMS, on the other hand, through Mass Spectral data, holds a great deal of basic information about a molecule, such as bond strengths, conformations, and molecular fragments, and these, in some cases, seem to lead to better predictivity than through the use of more complicated molecular descriptor systems.

Turning to Cl^- transport inhibitors [Tables/Figures 5.3, 5.5-5.9], we note that EVA significantly outperformed other comparators, with the only exception being PCAMS. This implies than when considering this brand of activity of a molecule, vibrational frequencies and the positioning of atoms within the molecule are extremely important. Counter-intuitively hydrophobicity, hydrogen bonding, and electrostatics seem to have a lesser role to play. Most interestingly, the basic, more complete information of the Mass Spectral data has provided all of the necessary information for successful prediction.

Finally, when we consider the TOXNET results [Tables/Figures 5.4, 5.5-5.9], EVA outperformed all other descriptors, including PCAMS, which, in the last analysis, performed poorest of all. The last's poor results suggest that in certain examples it is important which conformer is being investigated. Correspondingly, vibrational frequencies (for example in EVA) are a better descriptor in this application. This may be because these are more differentiable in the substitution of nucleotide bases and the insertion or deletion of one or more nucleotides in DNA sequences.

It appears, therefore, that the use of PCAMS, as with any descriptor system, must be with caution: if it is the case that the underlying data found in a Mass Spectrum is not sufficiently nuanced to encode that element of the molecule which is important in the activity, then it will necessarily be the case that PCAMS will give rise to poor results.

In order to observe these discrepancies more clearly, it is necessary to examine them by taking each molecular descriptor in turn (Figures 5.10)

5.5 Summary of Cross-Validation Values by Molecular Descriptor System

Table 5.10 – A summary of cross-validation values, q^2, for PCAMS

	ANN	*SVM*	*FSVM*	*GA*	*VP*
Carcinogenicity (TD_{50}) in rats	0.84	0.87	0.88	0.46	0.85
Human erythrocyte Cl^- Transport inhibitors	0.64	0.74	0.77	0.63	0.71
TOXNET mutagens	0.66	0.54	0.53	0.47	0.60

Figure 5.10 – A graphical representation of Table 5.10

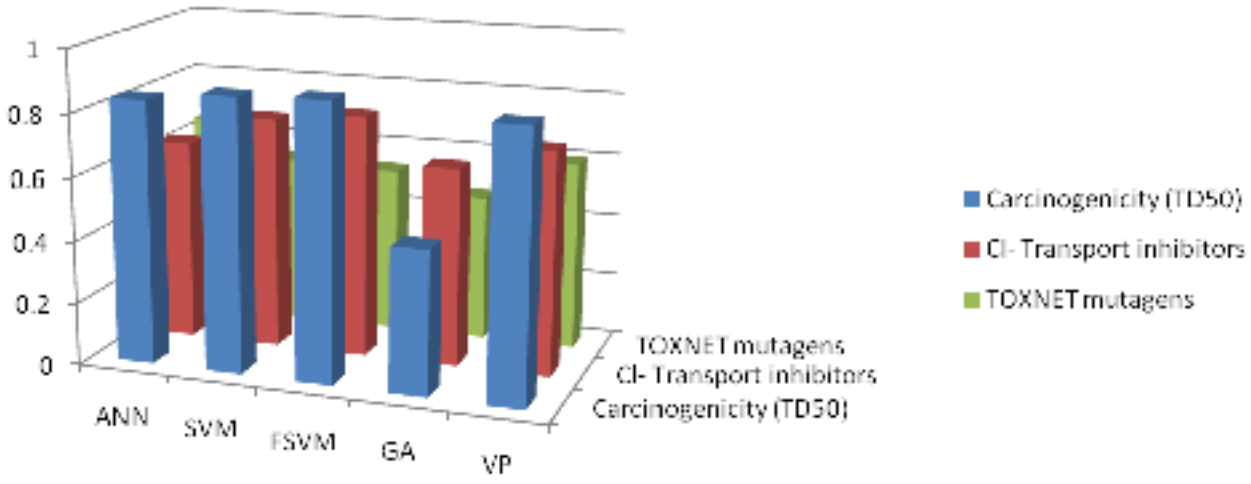


Table 5.11 – A summary of cross-validation values, q^2, for CoMFA

	ANN	*SVM*	*FSVM*	*GA*	*VP*
Carcinogenicity (TD_{50}) in rats	0.84	0.84	0.84	0.41	0.81
Human erythrocyte Cl^- Transport inhibitors	0.55	0.61	0.61	0.59	0.68
TOXNET mutagens	0.68	0.45	0.45	0.46	0.51

Figure 5.11 – A graphical representation of Table 5.11

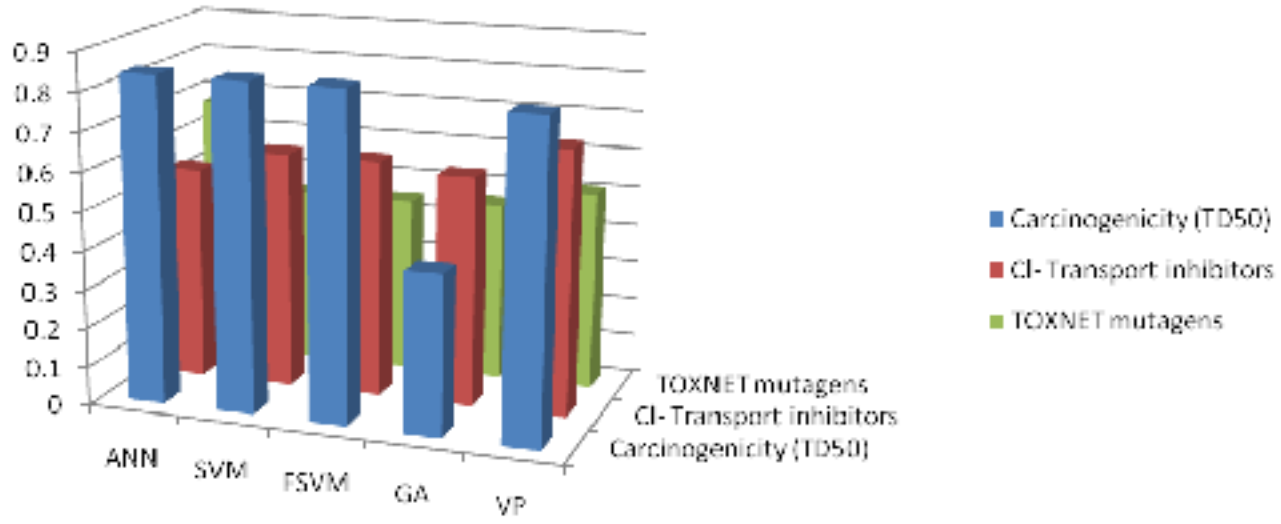


Table 5.12 – A summary of cross-validation values, q^2, for CoMSIA

	ANN	*SVM*	*FSVM*	*GA*	*VP*
Carcinogenicity (TD_{50}) in rats	0.83	0.85	0.85	0.47	0.79
Human erythrocyte Cl- Transport inhibitors	0.56	0.66	0.66	0.51	0.67
TOXNET mutagens	0.68	0.45	0.45	0.47	0.55

Figure 5.12 – A graphical representation of Table 5.12

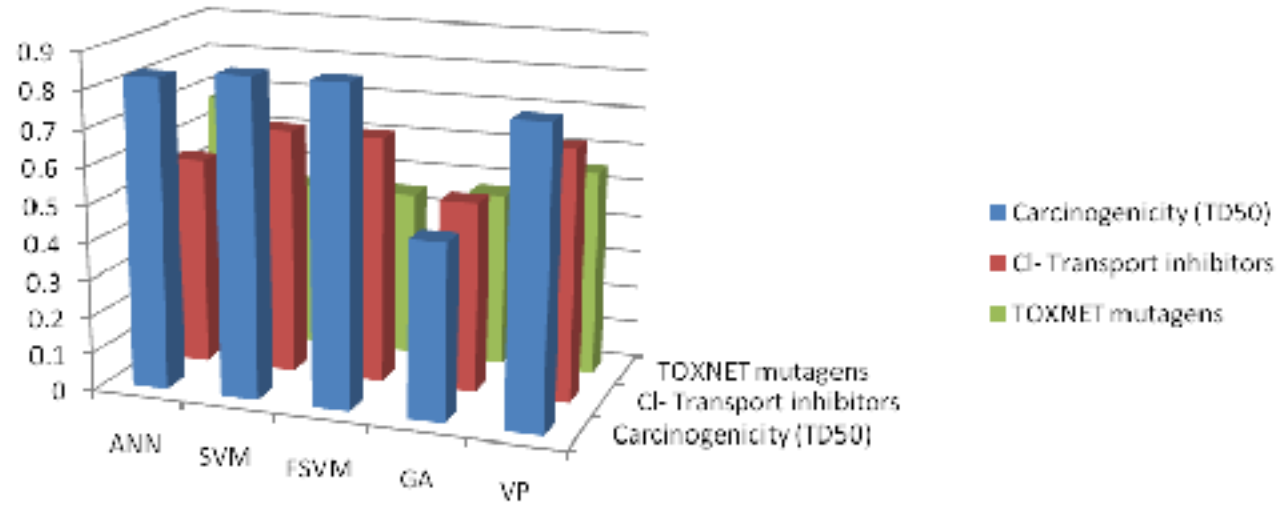


Table 5.13 – A summary of cross-validation values, q^2, for HQSAR

	ANN	*SVM*	*FSVM*	*GA*	*VP*
Carcinogenicity (TD_{50}) in rats	0.74	0.85	0.85	0.40	0.78
Human erythrocyte Cl^- Transport inhibitors	0.54	0.54	0.53	0.58	0.69
TOXNET mutagens	0.70	0.70	0.71	0.64	0.67

Figure 5.13 – A graphical representation of Table 5.13

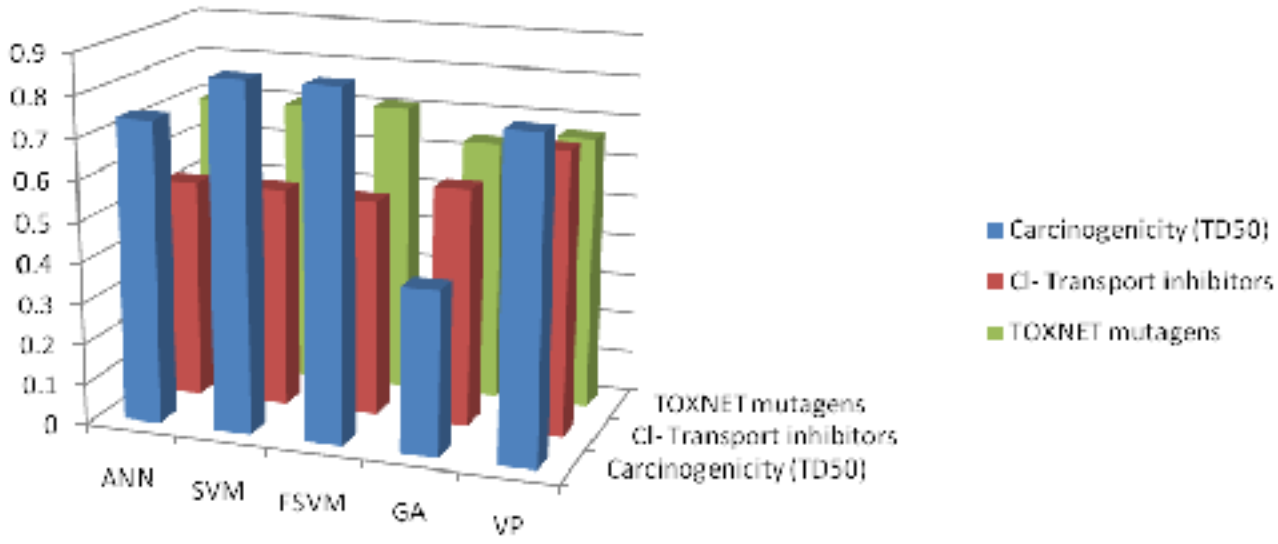
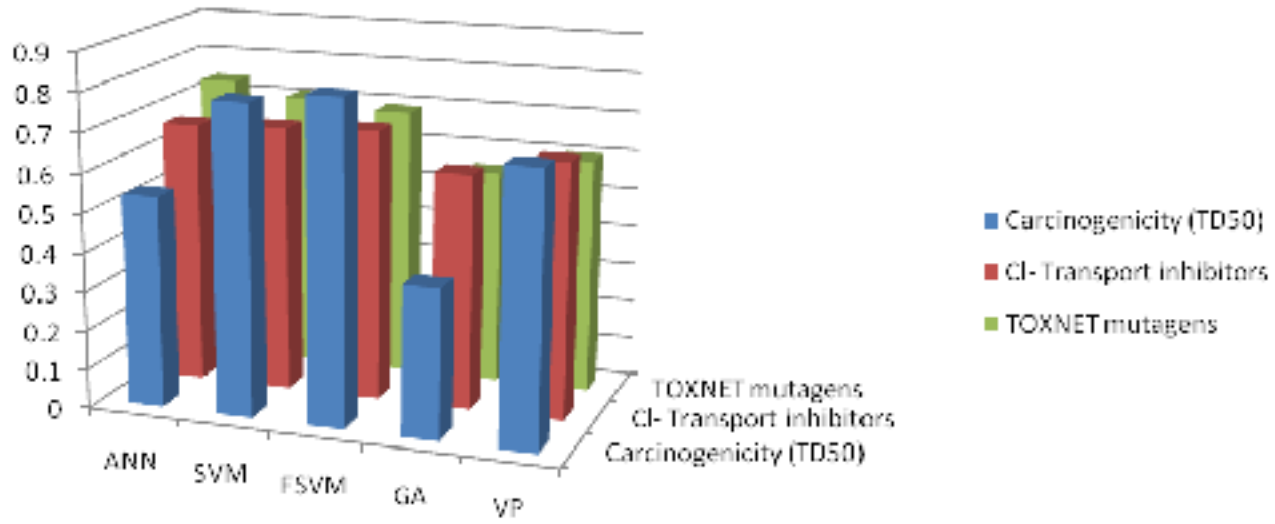


Table 5.14 – A summary of cross-validation values, q^2, for EVA

	ANN	*SVM*	*FSVM*	*GA*	*VP*
Carcinogenicity (TD_{50}) in rats	0.54	0.79	0.82	0.38	0.69
Human erythrocyte Cl^- Transport inhibitors	0.67	0.68	0.69	0.60	0.65
TOXNET mutagens	0.74	0.71	0.69	0.55	0.60

Figure 5.14 – A graphical representation of Table 5.14



5.6 Detailed results of predicted Carcinogen TD_{50} bands

This section provides visual representations of the best predictions of the ANN, the SVM, the PSVM, the GA and the VP (Tables/Figures 5.10-5.14), regardless of molecular descriptor system used. To plot each example would be unnecessarily long-winded.

The graphs below must be plotted for a good reason – it is necessary to ensure that examples to do not exist of cases where, whilst a technique might be considered an excellent predictive tool in terms of pure cross-validation values, a datum is misbanded to an extent where serious implications would exist for the application of the technique in a field as exacting as drug design.

Fig 5.15 – Predicted bands, grouped by actual band, using an ANN and PCAMS. q^2=0.84

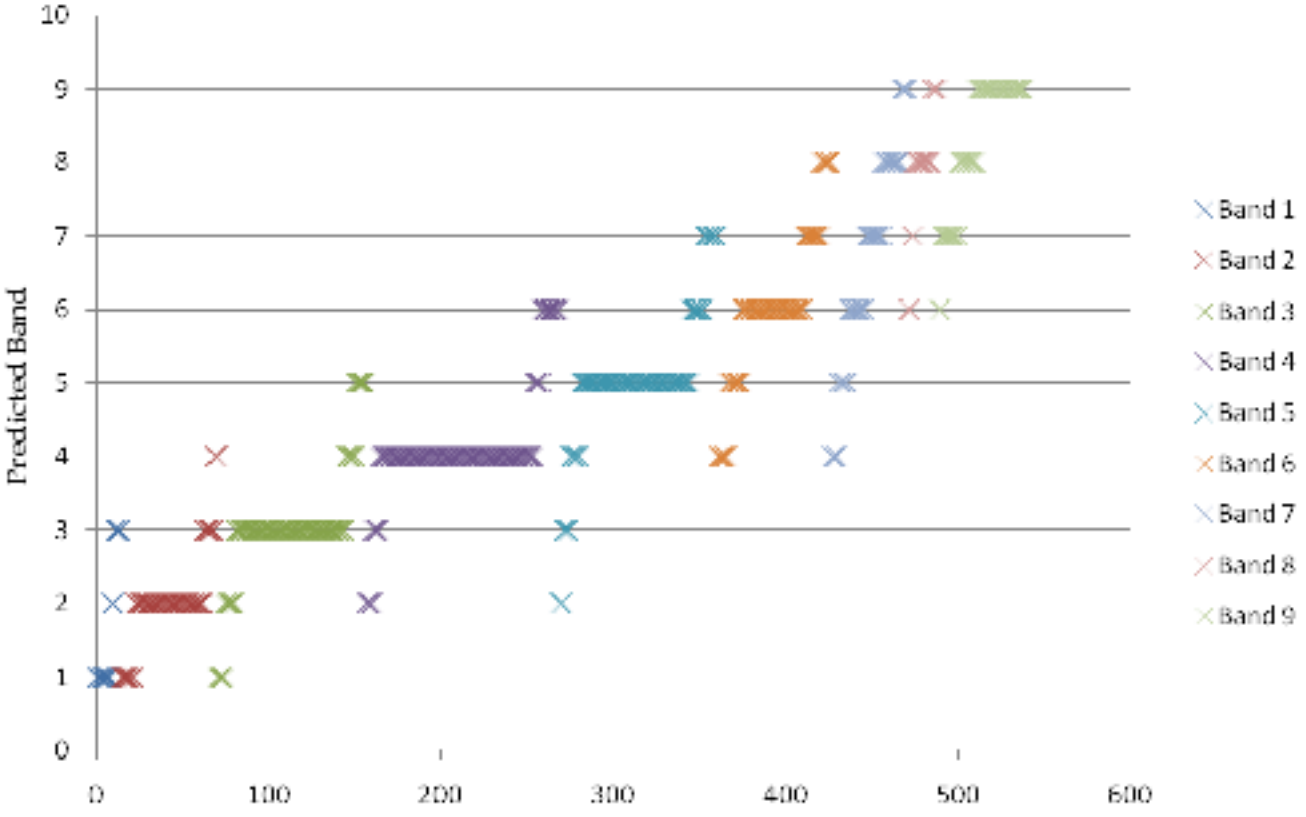


Fig 5.16 – Predicted bands, grouped by actual band, using a SVM and PCAMS. q^2=0.87

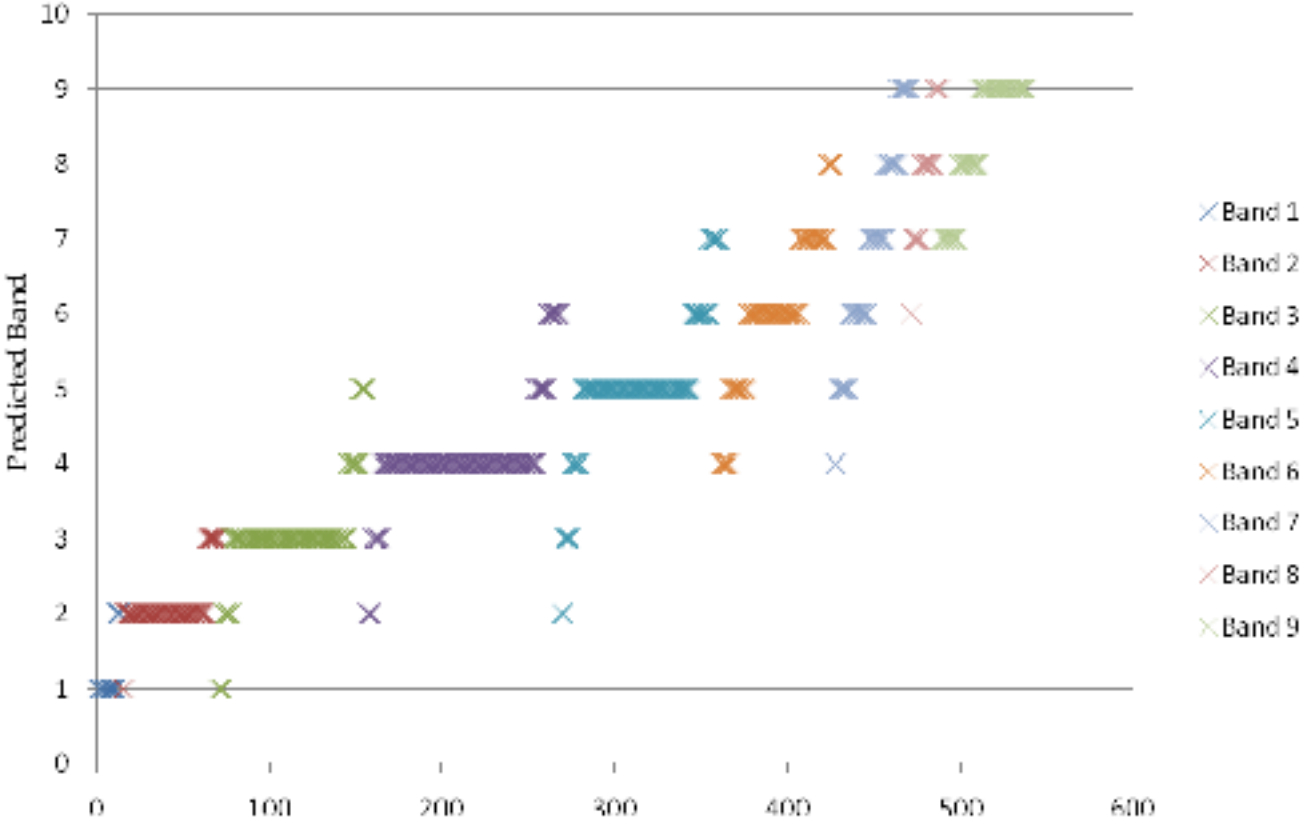


Fig 5.17 – Predicted bands, grouped by actual band, using a FSVM and PCAMS. q^2=0.88

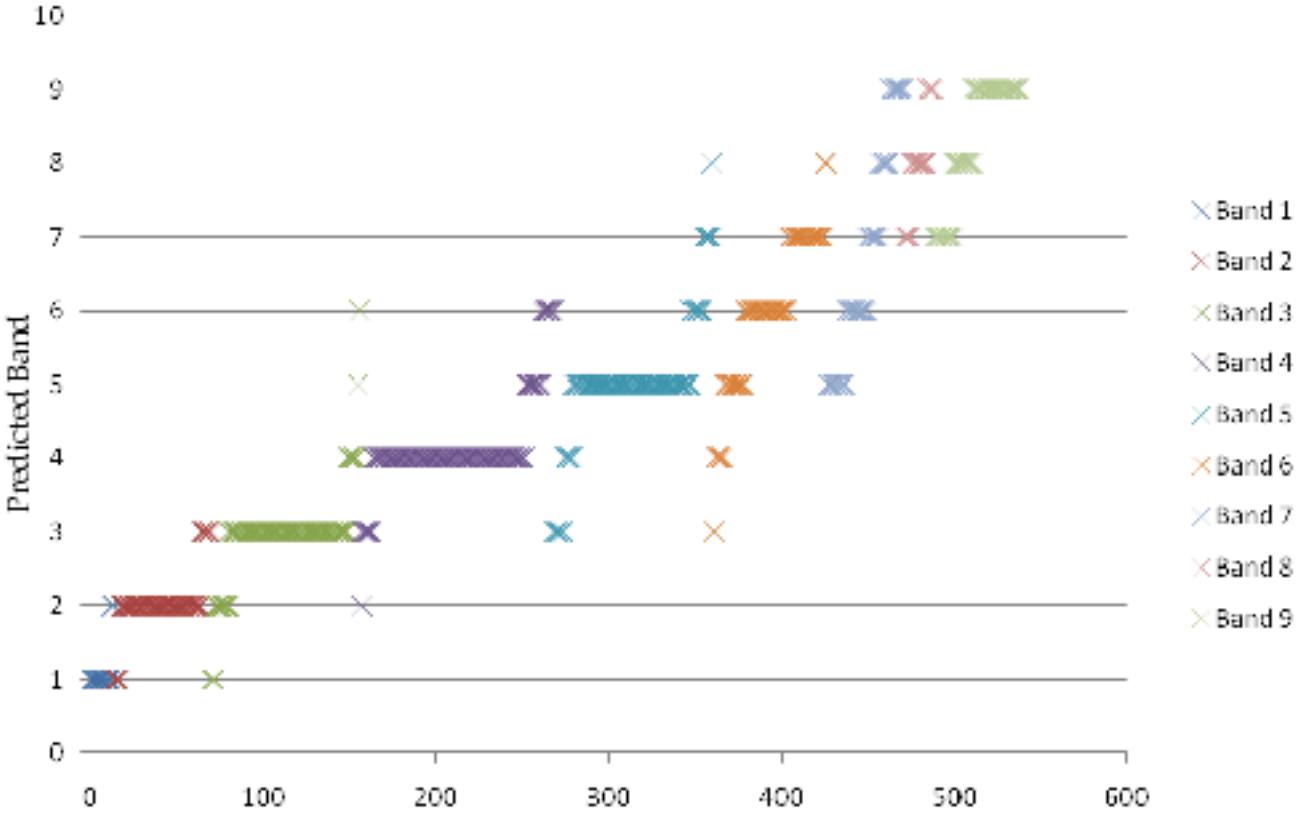


Fig 5.18 – Predicted bands, grouped by actual band, using a GA and CoMSIA. q^2=0.47

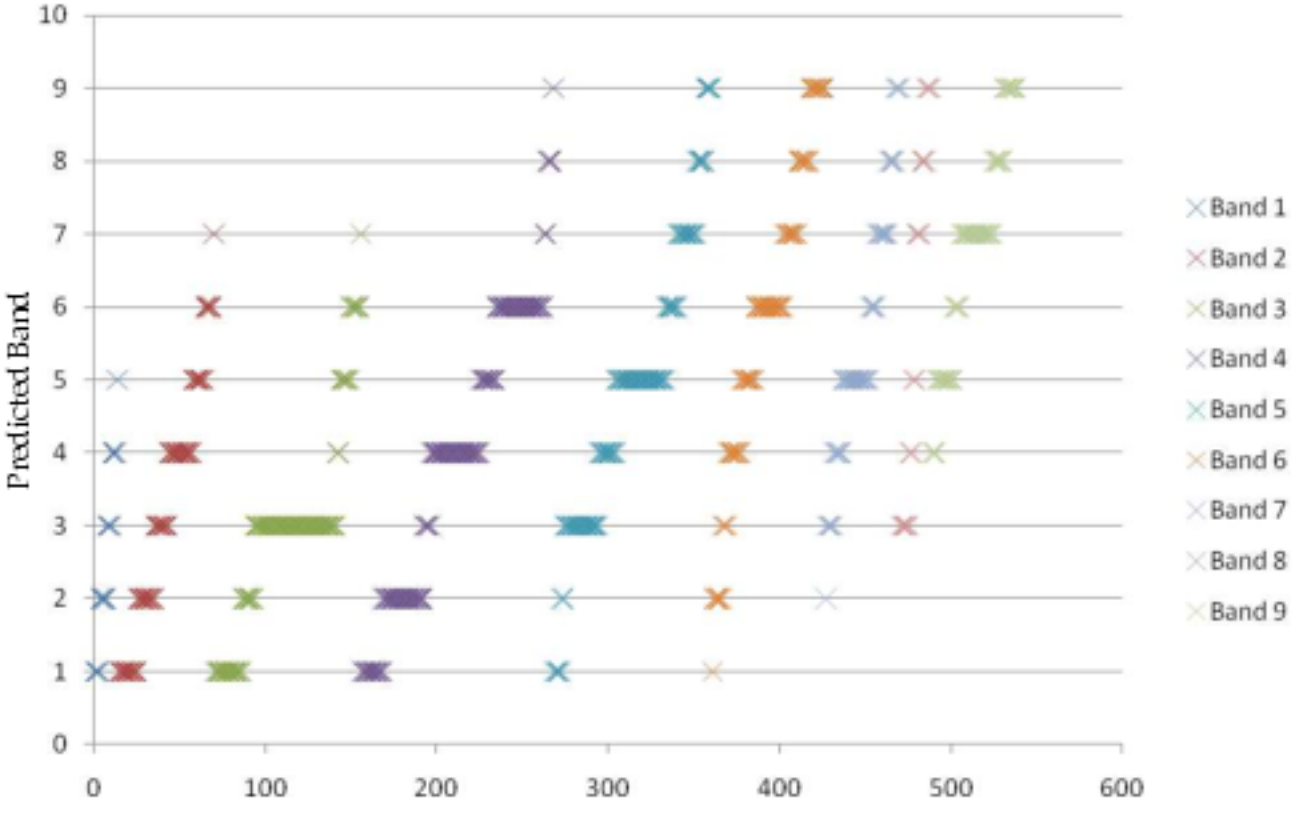
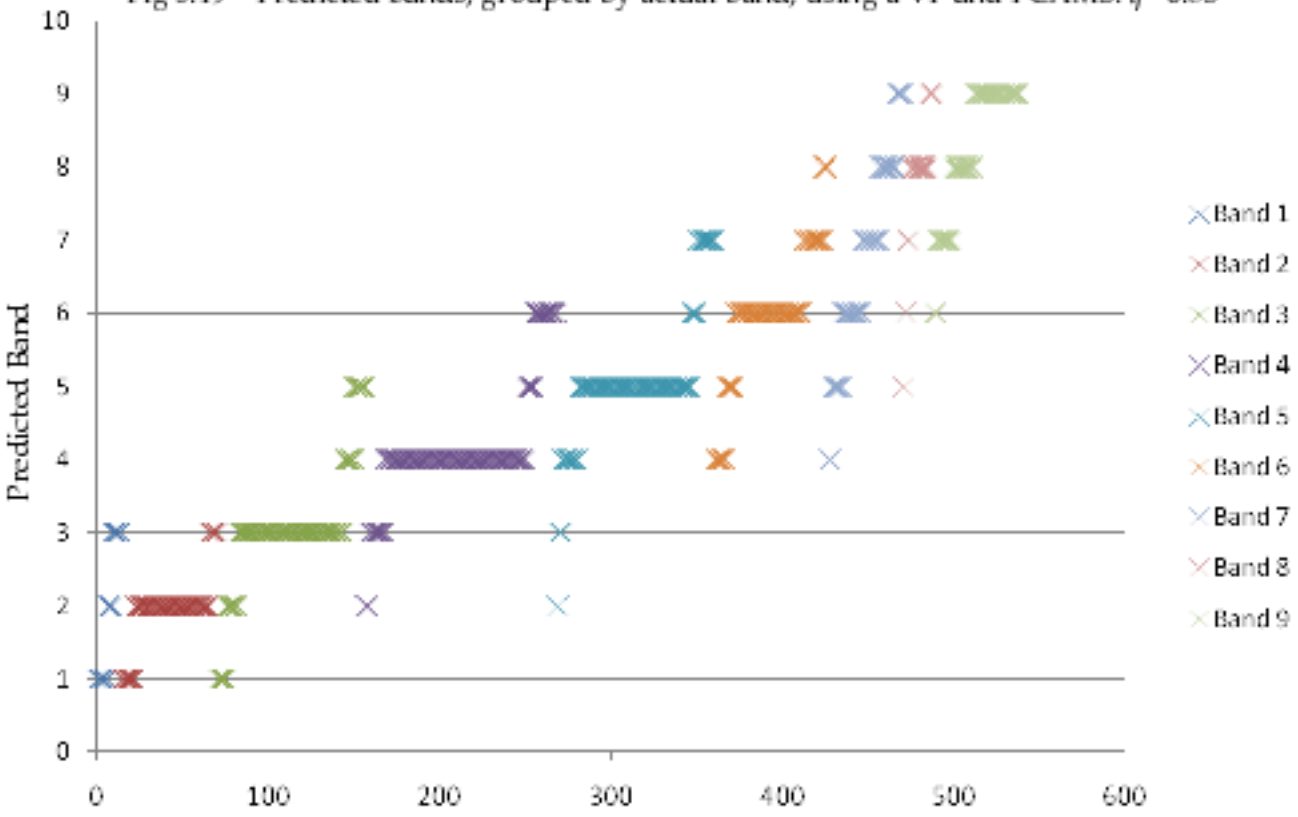


Fig 5.19 – Predicted bands, grouped by actual band, using a VP and PCAMS. q^2=0.85



5.5 Detailed results of predicted Cl^- Transport Inhibitor Bands

Fig 5.20 – Predicted bands, grouped by actual band, using an ANN and EVA. q^2=0.67

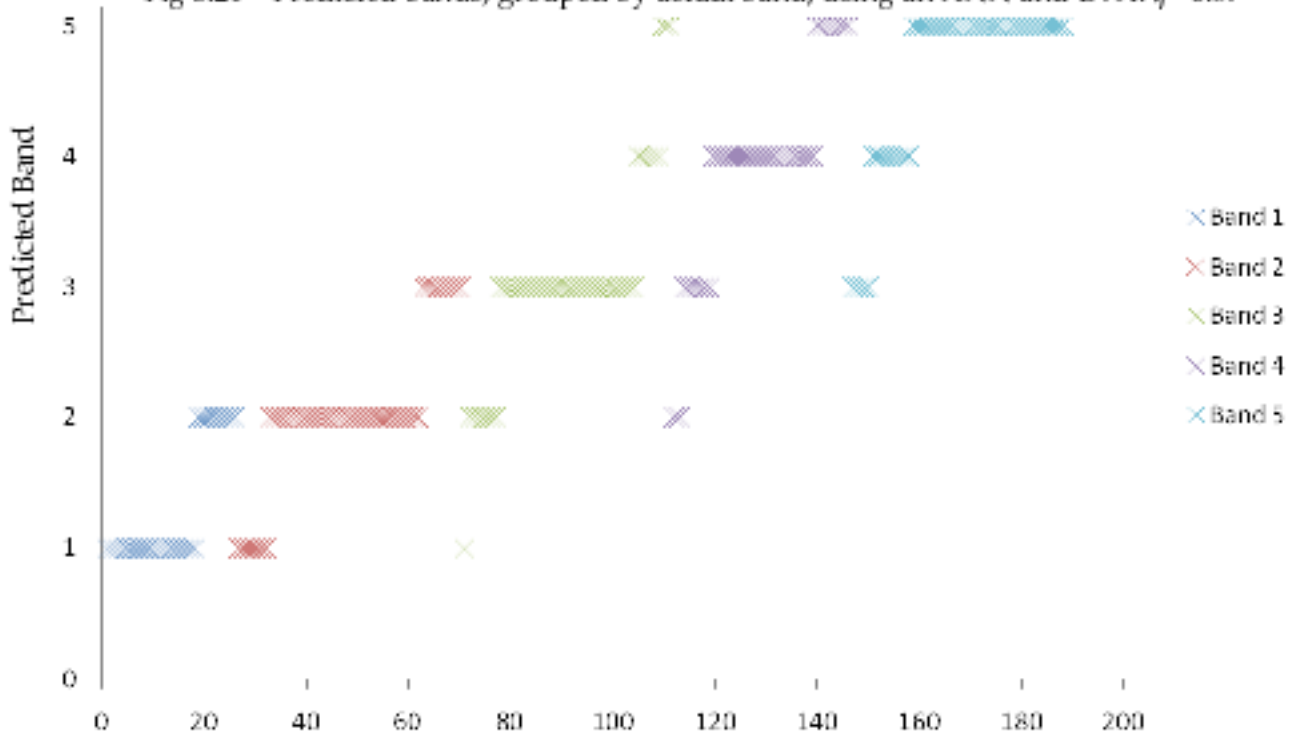


Fig 5.21– Predicted bands, grouped by actual band, using an SVM and PCAMS. q^2=0.74

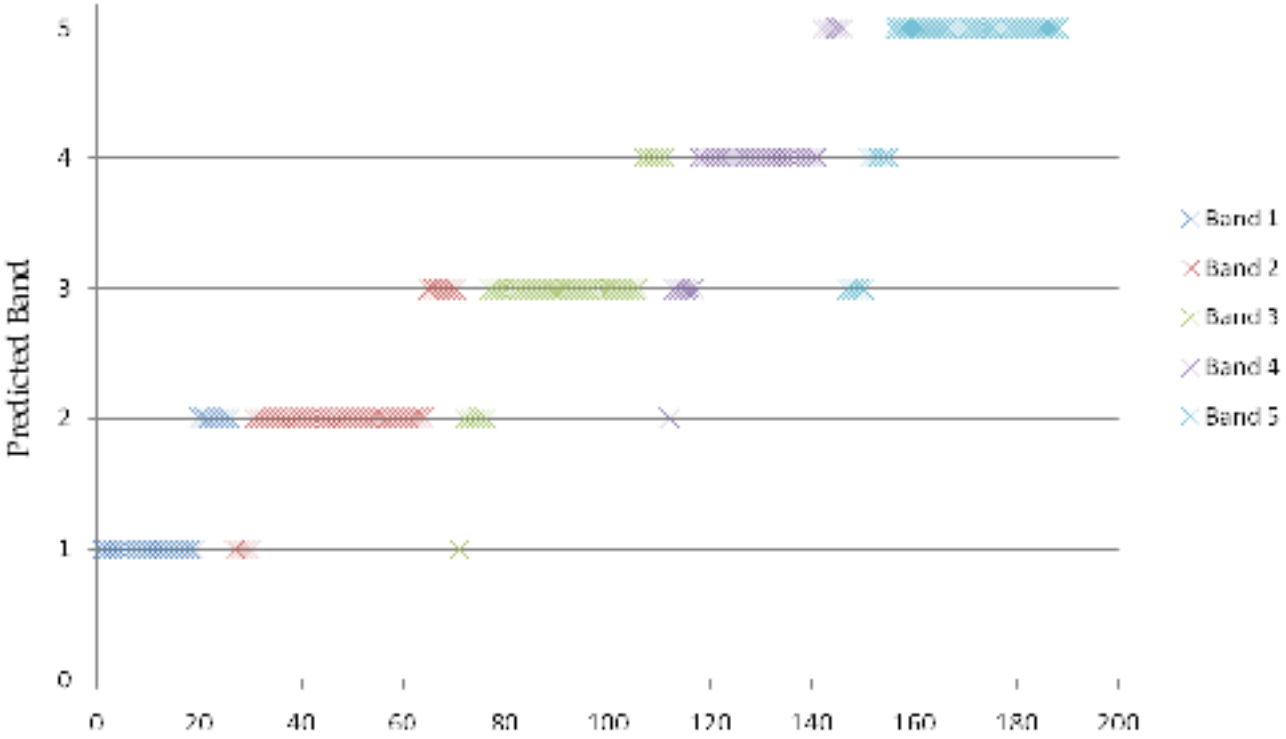


Fig 5.22 – Predicted bands, grouped by actual band, using an FSVM and PCAMS. q^2=0.77

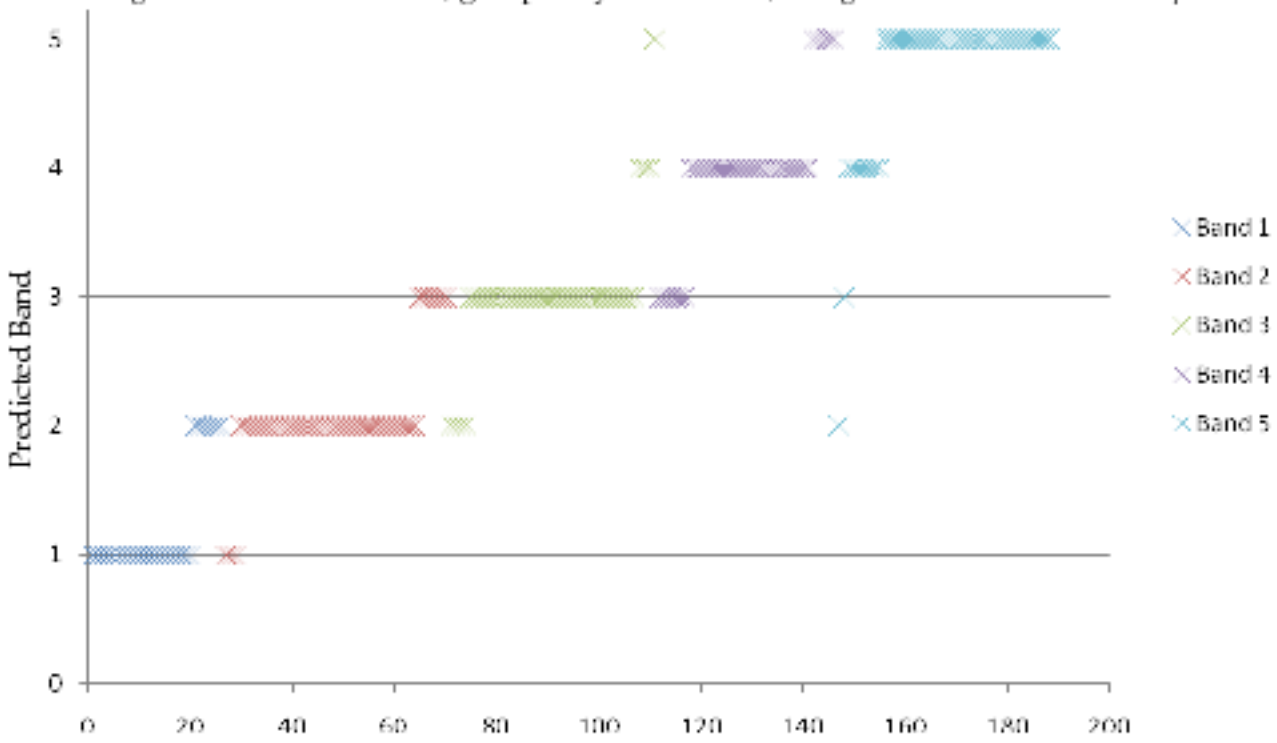


Fig 5.23 – Predicted bands, grouped by actual band, using a GA and PCAMS. q^2=0.63

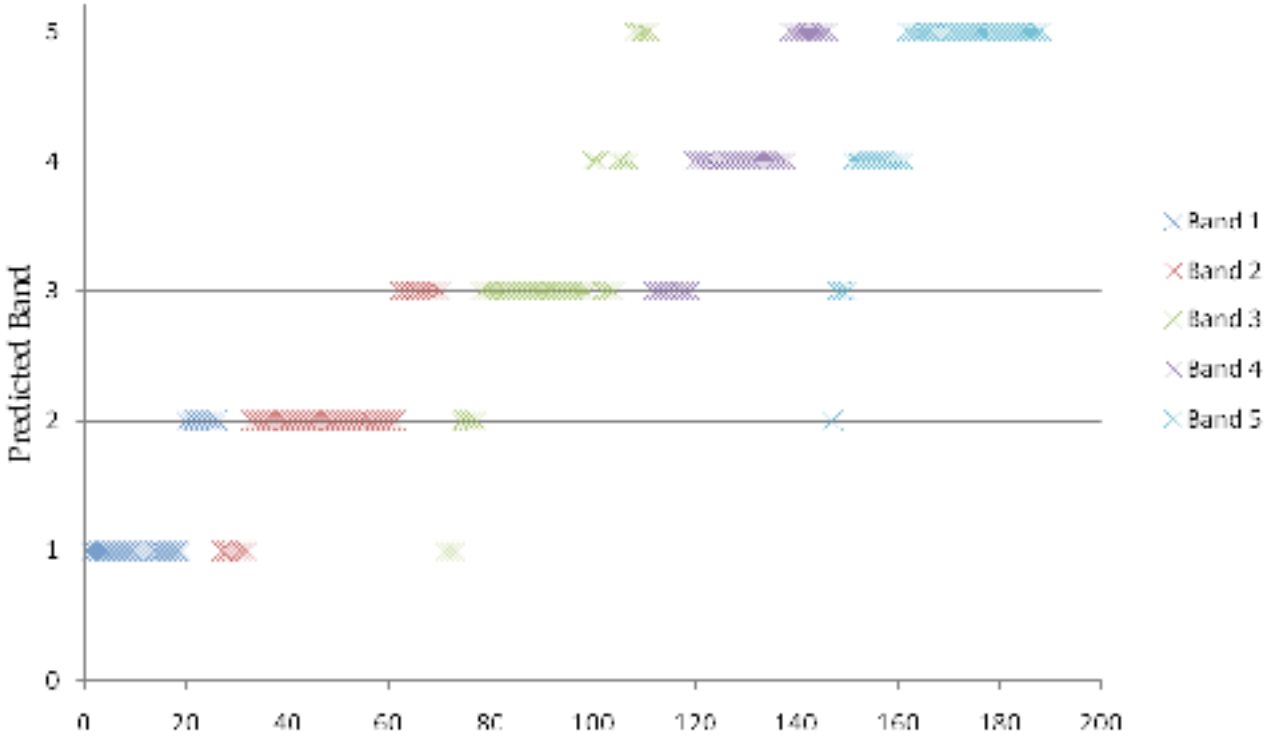
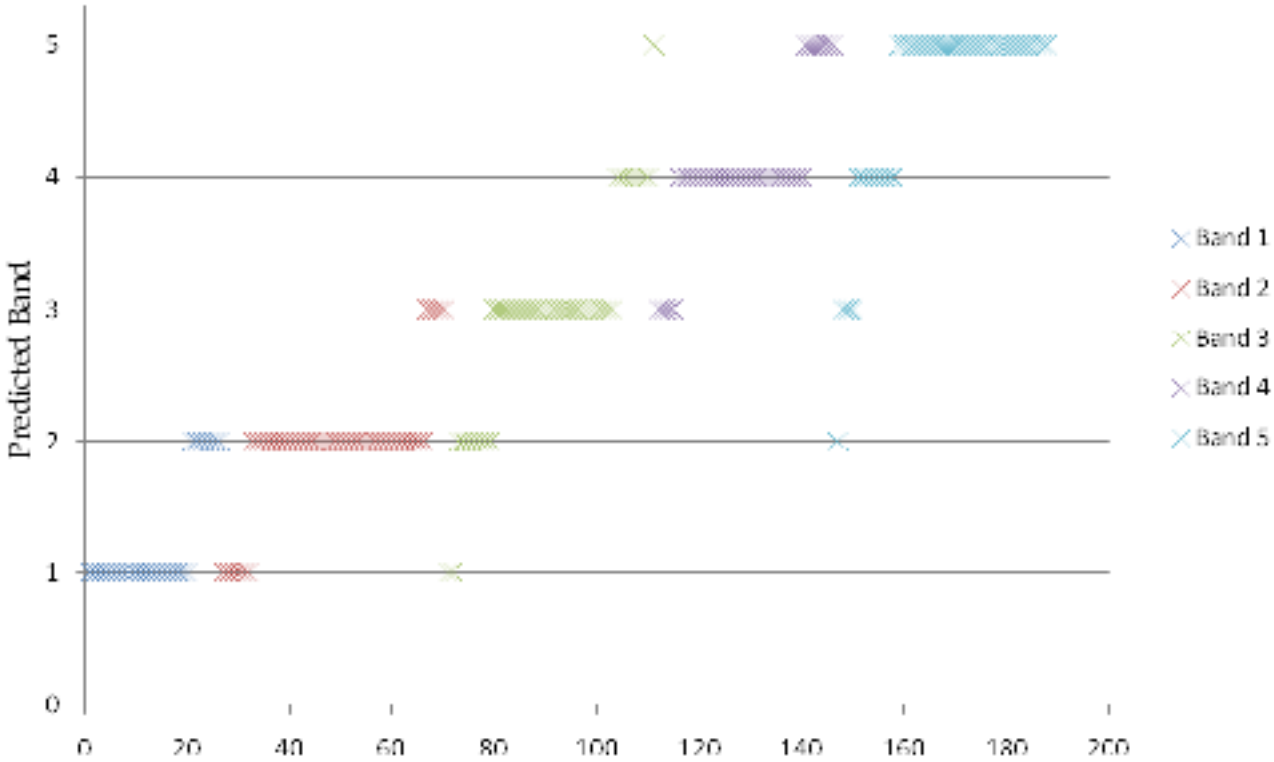


Fig 5.24 – Predicted bands, grouped by actual band, using a VP and PCAMS. q^2=0.71



5.6 Detailed results of predicted TOXNET Mutagen Bands

Fig 5.25 – Predicted bands, grouped by actual band, using an ANN and EVA. q^2=0.74

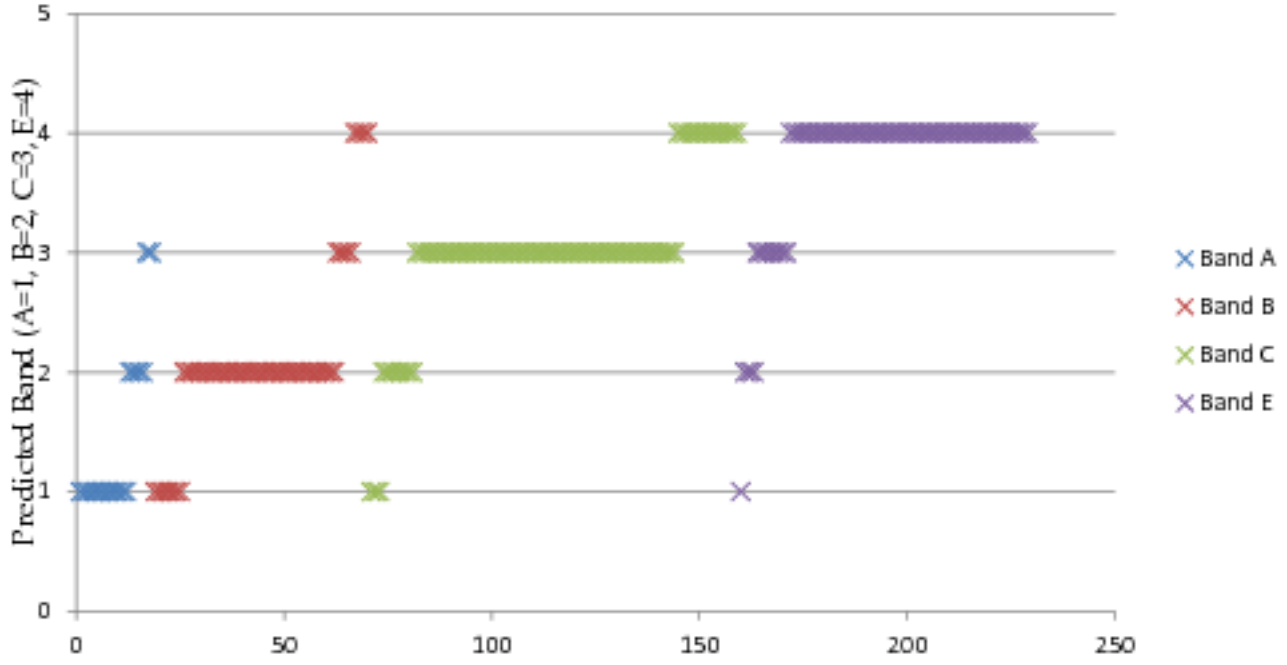


Fig 5.26 – Predicted bands, grouped by actual band, using an SVM and EVA. q^2=0.71

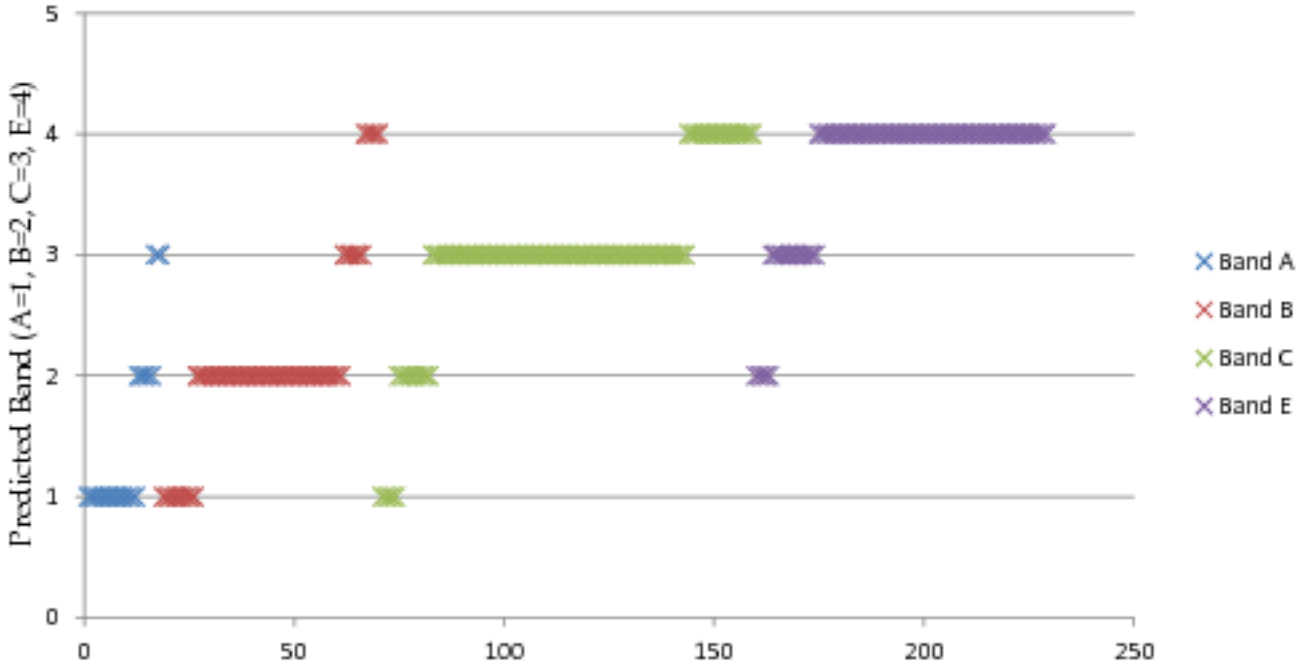


Fig 5.27 – Predicted bands, grouped by actual band, using an FSVM and HQSAR. q^2=0.71

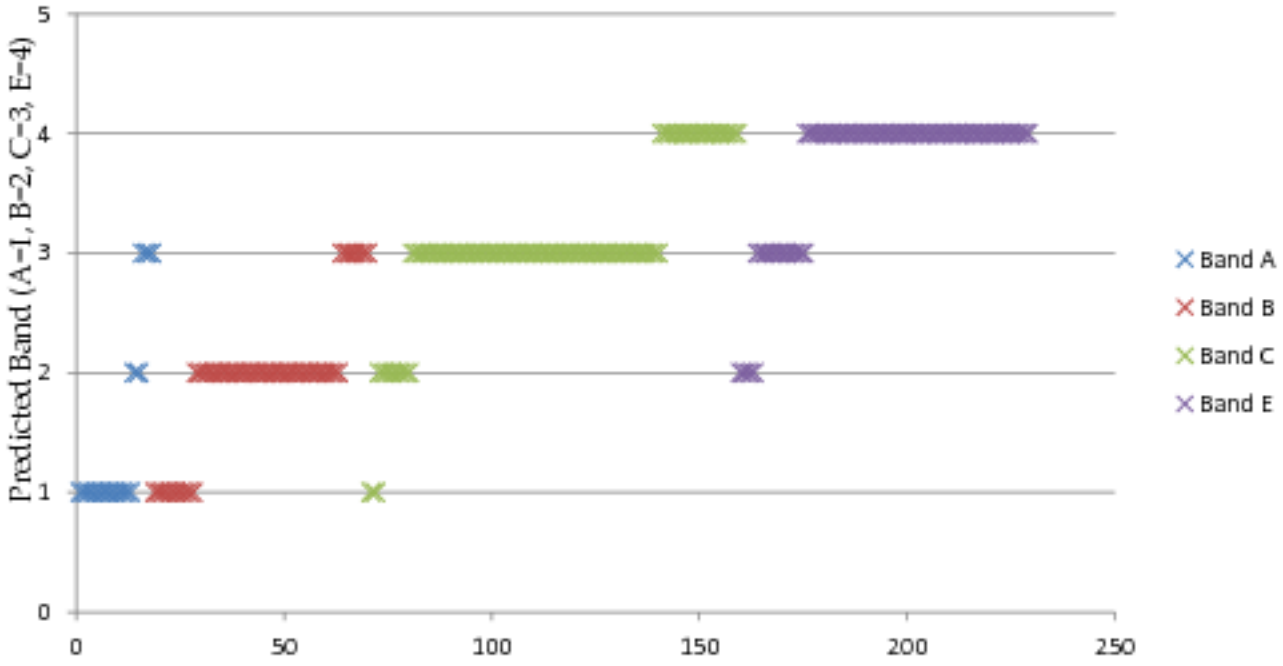


Fig 5.28 – Predicted bands, grouped by actual band, using a GA and HQSAR. q^2=0.64

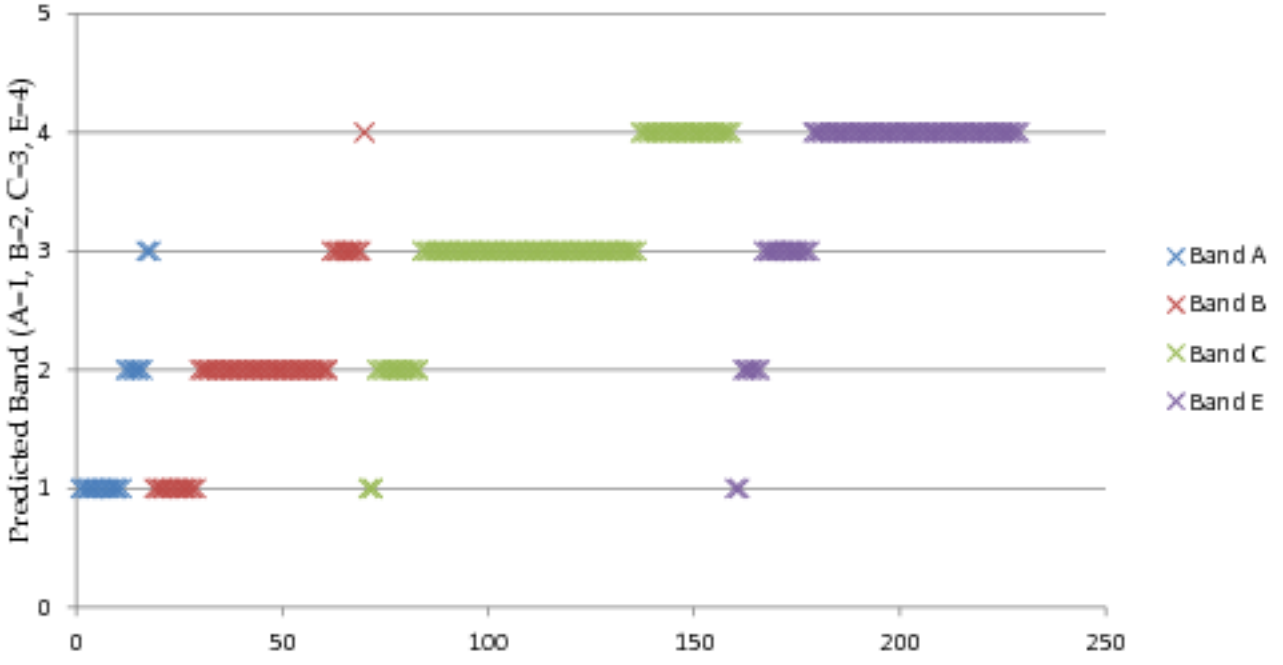
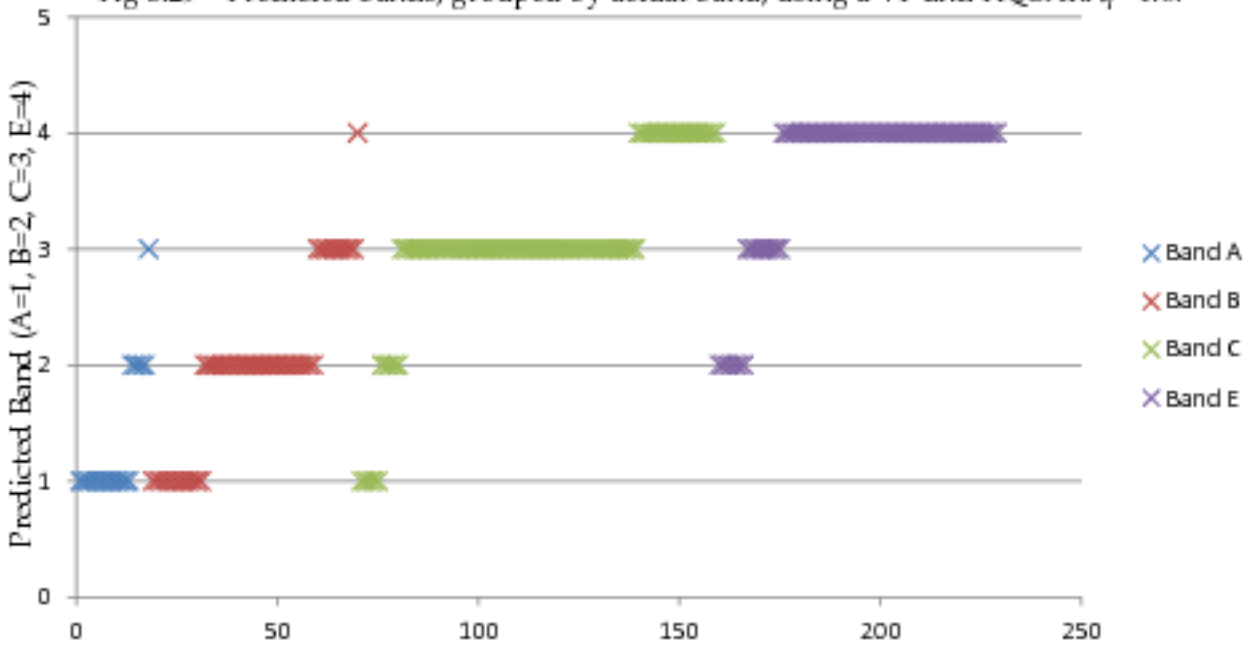


Fig 5.29 – Predicted bands, grouped by actual band, using a VP and HQSAR. q^2=0.67



5.7 Discussion: misbanding and its implications

Comparisons between Figures 5.15 (ANN), 5.16 (SVM), and 5.17 (FSVM), show that there were fewer severe misclassifications (misbanding by 3 or more) in the SVM than in the ANN and even fewer than that in the FSVM. This is also the case, to a lesser extent if we compare Figures 5.20, 5.21, and 5.22 for the Cl^- Transport Inhibition data (and assume that severe misbanding is misbanding ≥2). Even in the case where the ANN performed better than the SVM and the FSVM overall (the TOXNET data), the ANN severely misclassified more data than the SVM. For example in Figure 5.25, a Band 4 mutagen is classified as a Band 1. It is unnecessary to point out that this could have severe consequences in the field of drug design.

If we look at Figure 5.23, it is apparent that while Genetic Algorithms get some sort of handle on the problem, they are too erratic to provide reliable predictions. In the case where there are *de facto* three bands (TD_{50} carcinogenicity), a cross-validation value of 0.47 is little better than random prediction. It may be that we are forced to conclude, despite repeated trials, that these techniques and architectures are generally not applicable to this sort of problem.

Otherwise individual predictions seem to be in line with the overall predictive ability of the technique and molecular descriptor combination as a whole.

5.7 Results obtained through the Self-Organising Map

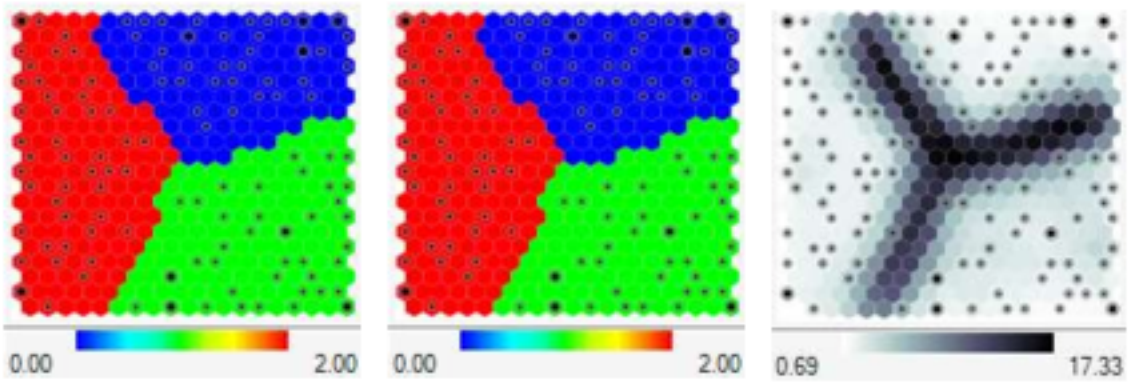
I will now turn to the results obtained through the use of a Self-Organising Map. For each example three visualisations of the clustering are shown. The first is a visualisation of the hexagonal clustering, where the number of clusters (represented by different colours) has been selected automatically by the SOM. The second visualisation is where the SOM is forced to define the same number of clusters as bands assigned in this thesis to the dataset under investigation. This visualisation will allow us to identify whether or not this clustering extracts the various features in which we are interested, or whether other irrelevant characterisations have emerged. Thirdly, a black-and-white

visualisation of the unified distance matrix will give an insight into the unified distances of each map unit.

5.7.1 Self-Organising Map Clustering of the Fisher Iris Data

As an example of how these should appear in success, Figure 5.30 displays the relevant figures for the Iris data, which, as is immediately visible, cluster the nodes representing the three species (Versicolour, Verginica, and Setosa) very easily, even though algebraic agents would struggle to achieve this feat.

Fig 5.30 – SOM Clusters using physical descriptors, when applied to the Iris data. Visualisations (from left to right) include automatic calculation of the number of clusters, an enforced number of clusters to match the number of bands, and a U-matrix.



5.7.2 Self-Organising Map Clustering of the TD_{50} carcinogen Data

First, we shall consider the figures for the various molecular descriptor systems when applied to the TD_{50} carcinogen data.

Fig 5.31 – SOM Clusters using PCAMS, when applied to the TD_{50} carcinogen data. Visualisations (from left to right) include automatic calculation of the number of clusters, an enforced number of clusters to match the number of bands, and a U-matrix.

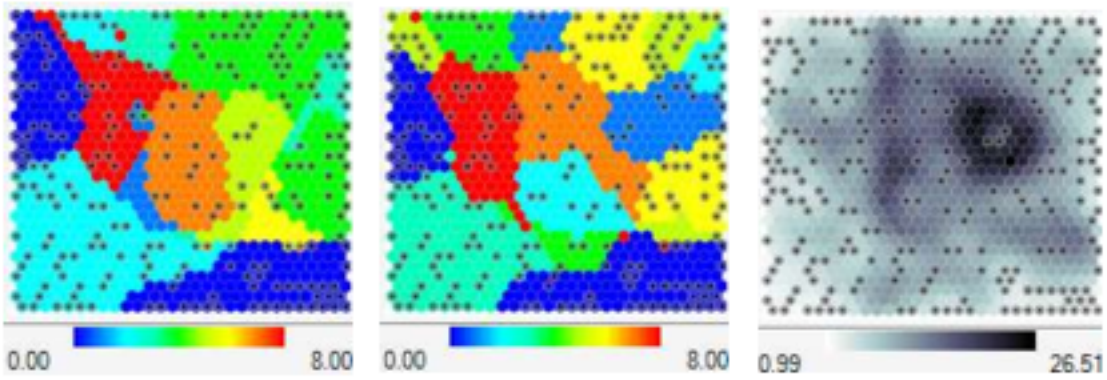


Fig 5.32 – SOM Clusters using CoMFA, when applied to the TD_{50} carcinogen data. Visualisations (from left to right) include automatic calculation of the number of clusters, an enforced number of clusters to match the number of bands, and a U-matrix.

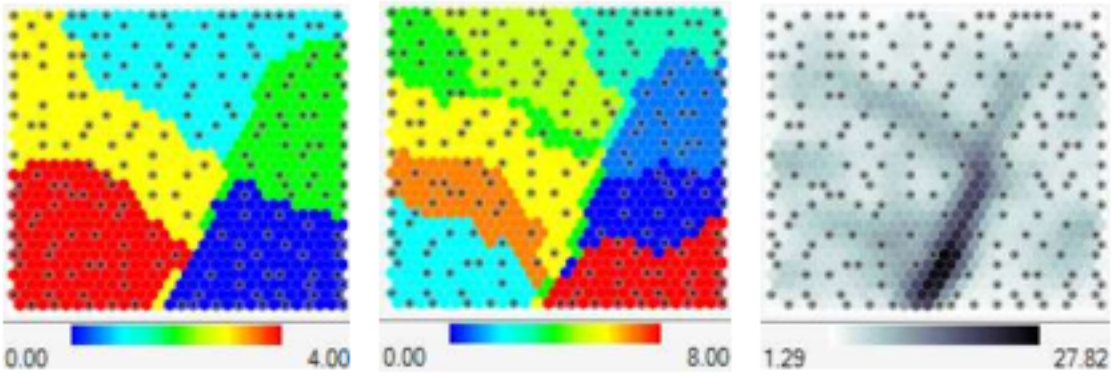


Fig 5.33 – SOM Clusters using CoMSIA, when applied to the TD_{50} carcinogen data. Visualisations (from left to right) include automatic calculation of the number of clusters, an enforced number of clusters to match the number of bands, and a U-matrix.

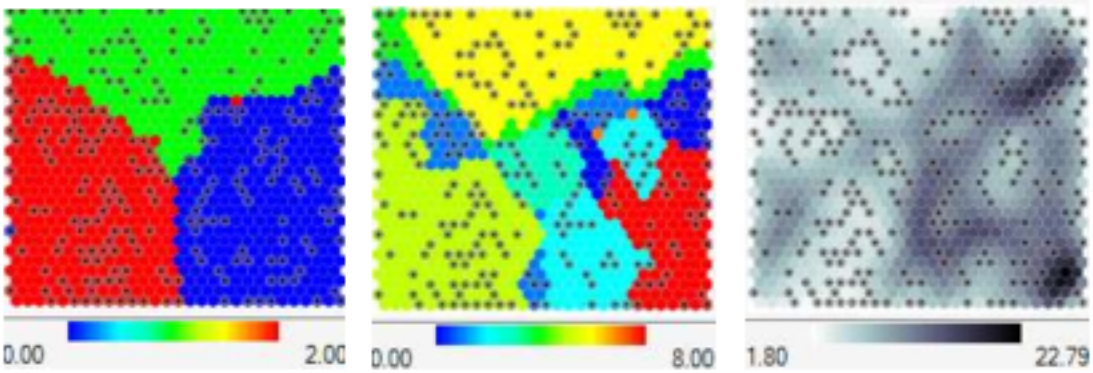


Fig 5.34 – SOM Clusters using HQSAR, when applied to the TD_{50} carcinogen data. Visualisations (from left to right) include automatic calculation of the number of clusters, an enforced number of clusters to match the number of bands, and a U-matrix.

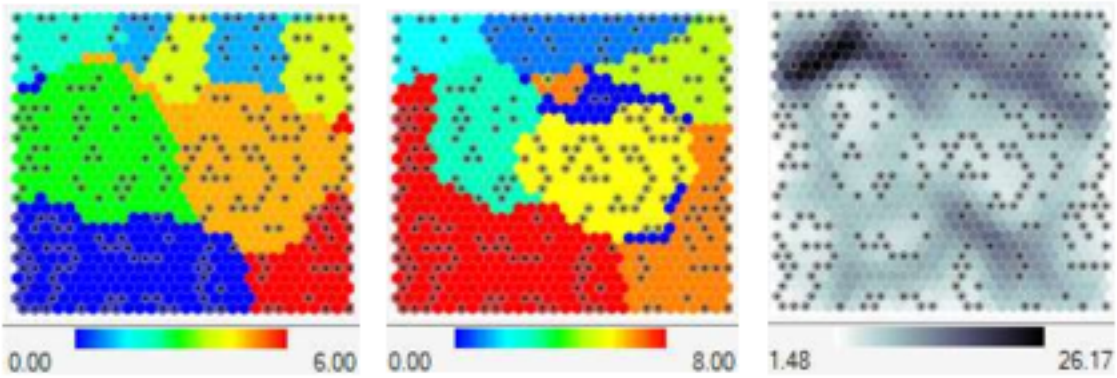
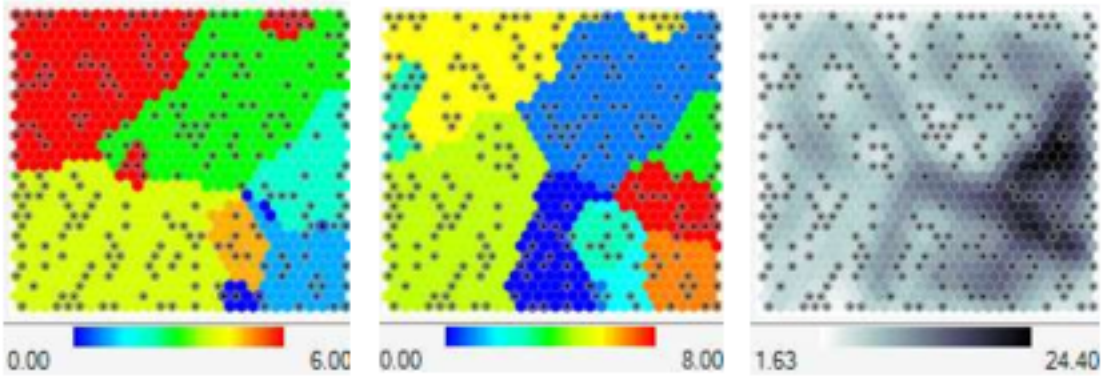


Fig 5.35 – SOM Clusters using EVA, when applied to the TD_{50} carcinogen data. Visualisations (from left to right) include automatic calculation of the number of clusters, an enforced number of clusters to match the number of bands, and a U-matrix.



These results (Figures 5.31-5.35) are interesting, if disappointing. Whilst certain features of the molecules appear to be extracted by clustering, unfortunately whichever features these are, it is apparent from an evaluation of how the different carcinogens map onto these clusters, that none of them is classified by TD_{50} carcinogenicity.

In many ways, this result could be seen as pleasing. They indicate that carcinogenicity seems not to be linked to more simple features of the molecule, through which we could predict it; the relationship is more complex and nuanced, and therefore cannot be solved by unsupervised learning, but requires supervised learning to extract from the data the explicit relationship which carcinogens have to their underlying structure. This implies that the examples of successful supervised learning elsewhere in this thesis are more of an achievement.

5.7.3 Self-Organising Map Clustering of Cl^- Transport Inhibitor Data

The Human Erythrocyte Cl^- Transport Inhibitor dataset has fewer examples, and therefore needs a smaller map.

Fig 5.36 – SOM Clusters using PCAMS, when applied to the Cl^- Transport Inhibitor data. Visualisations (from left to right) include automatic calculation of the number of clusters, an enforced number of clusters to match the number of bands, and a U-matrix.

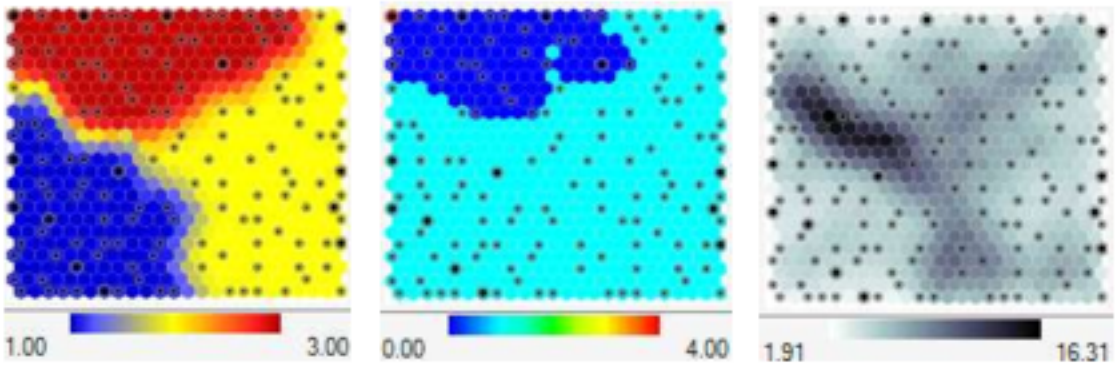


Fig 5.37 – SOM Clusters using CoMFA, when applied to the Cl^- Transport Inhibitor data. Visualisations (from left to right) include automatic calculation of the number of clusters, an enforced number of clusters to match the number of bands, and a U-matrix.

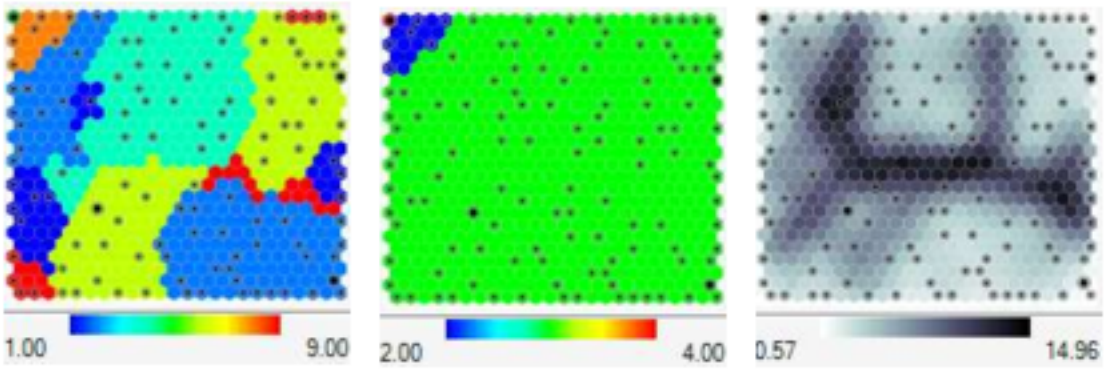


Fig 5.38 – SOM Clusters using CoMSIA, when applied to the Cl^- Transport Inhibitor data. Visualisations (from left to right) include automatic calculation of the number of clusters, an enforced number of clusters to match the number of bands, and a U-matrix.

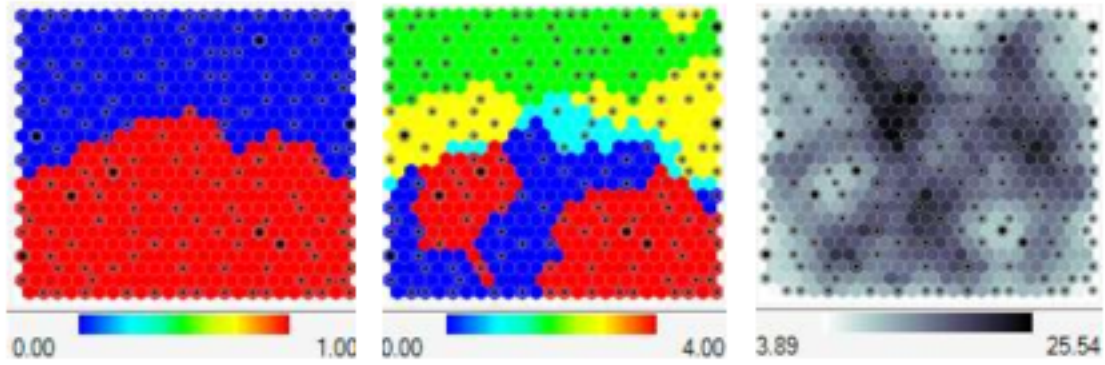


Fig 5.39 – SOM Clusters using HQSAR, when applied to the Cl^- Transport Inhibitor data. Visualisations (from left to right) include automatic calculation of the number of clusters, an enforced number of clusters to match the number of bands, and a U-matrix.

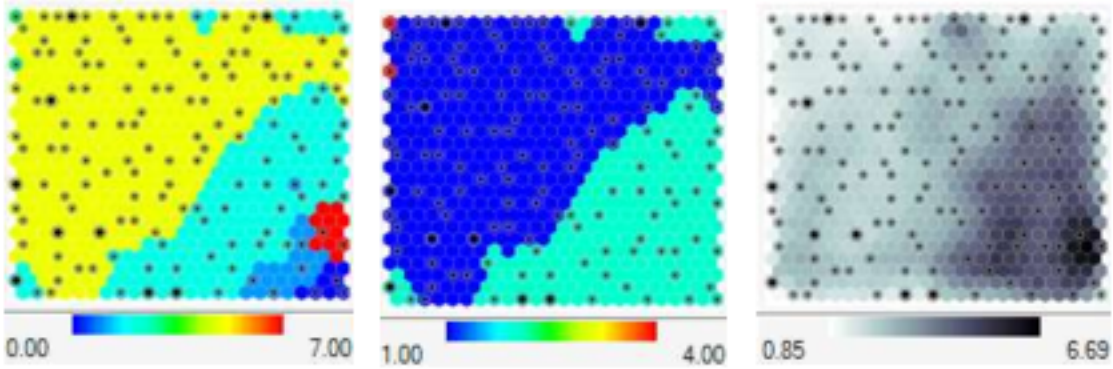
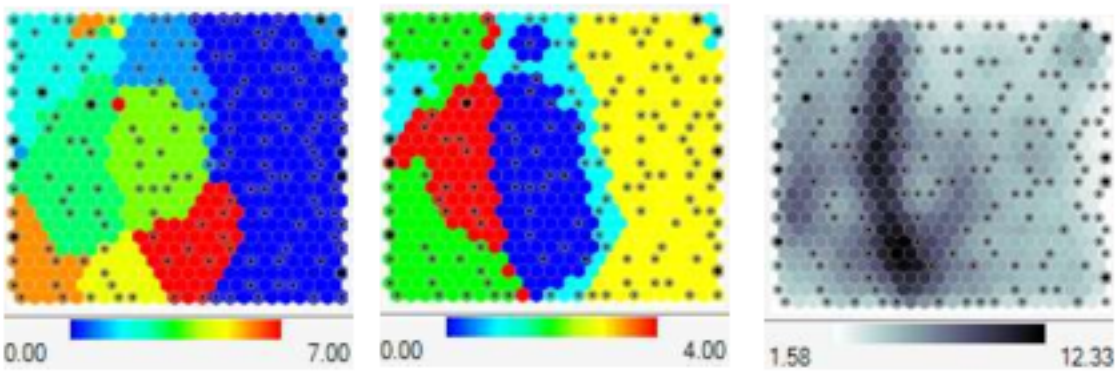


Fig 5.40 – SOM Clusters using EVA, when applied to the Cl^- Transport Inhibitor data. Visualisations (from left to right) include automatic calculation of the number of clusters, an enforced number of clusters to match the number of bands, and a U-matrix.



Two of the maps (Figures 5.36 and 5.40), which make use of PCAMS and EVA respectively, have performed rather well. Present in the red zone of the former, are most of the strong inhibitors and in the dark blue zone the weaker ones. The yellow zone represents a mixture of strong and weak. In those cluster SOM using EVA, the weaker and intermediate inhibitors can be found on the right-hand two thirds of the map and the stronger examples on the left. It is, perhaps unsurprising that we should be able to predict this type of activity rather well; it is clearly less complex that the activity of carcinogens which can have effect through several pathways. In the case of Cl^- Transport inhibition, there is only one action pathway, so it is more likely that that feature might be extracted during clustering.

Figures 5.37, 5.38, and 5.40 demonstrate that the incorporation of these molecular descriptors system into this technique has proved unsuccessful.

5.7.4 Self-Organising Map Clustering of TOXNET Mutagen Data

Fig 5.41 – SOM Clusters using PCAMS, when applied to the TOXNET Mutagen data. Visualisations (from left to right) include automatic calculation of the number of clusters, an enforced number of clusters to match the number of bands, and a U-matrix.

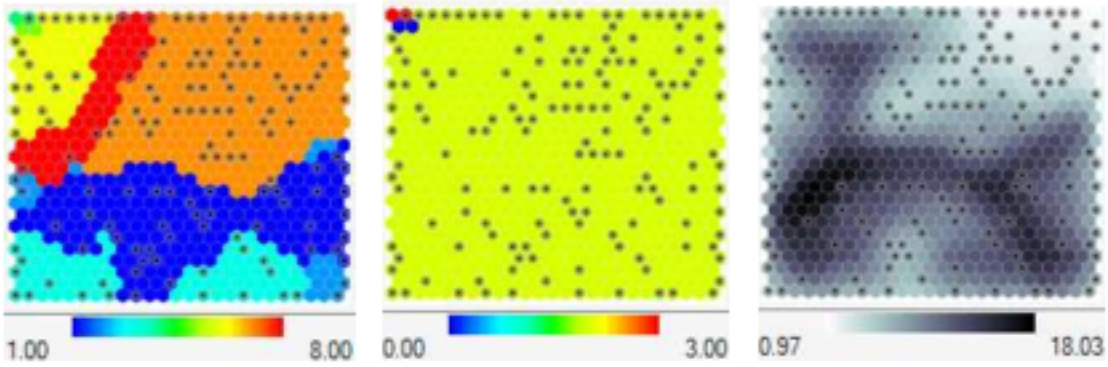


Fig 5.42 – SOM Clusters using CoMFA, when applied to the TOXNET Mutagen data. Visualisations (from left to right) include automatic calculation of the number of clusters, an enforced number of clusters to match the number of bands, and a U-matrix.

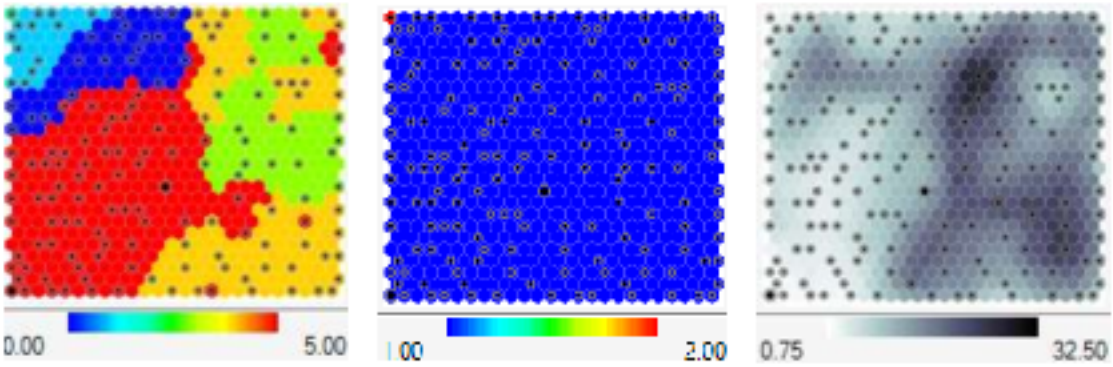


Fig 5.43 – SOM Clusters using CoMSIA, when applied to the TOXNET Mutagen data. Visualisations (from left to right) include automatic calculation of the number of clusters, an enforced number of clusters to match the number of bands, and a U-matrix.

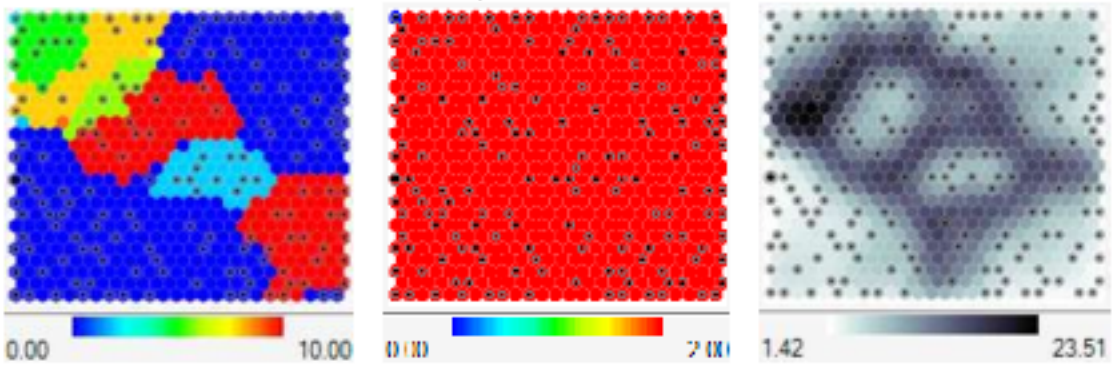


Fig 5.44 – SOM Clusters using HQSAR, when applied to the TOXNET Mutagen data. Visualisations (from left to right) include automatic calculation of the number of clusters, an enforced number of clusters to match the number of bands, and a U-matrix.

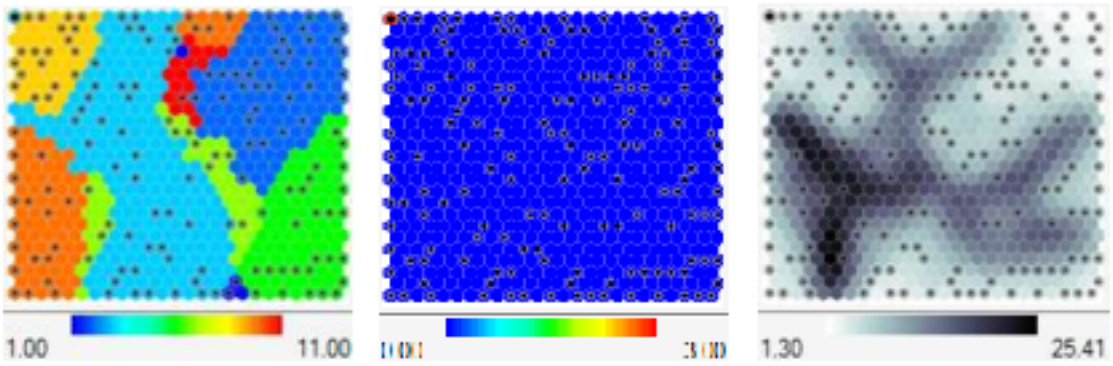
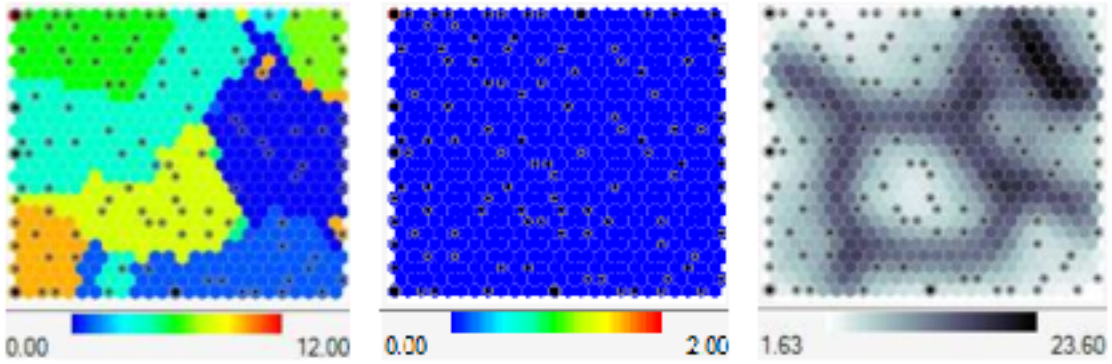


Fig 5.45 – SOM Clusters using EVA, when applied to the TOXNET Mutagen data. Visualisations (from left to right) include automatic calculation of the number of clusters, an enforced number of clusters to match the number of bands, and a U-matrix.



Somewhat unsurprisingly, these results (Figures 5.41-5.45) are of a similar nature to those of the carcinogens. None of the researched activity groups of the various mutagens is apparently discernable in any of these clusters. This lack of success is probably for an analogous reason to that given for the semblant results discussed in Section 5.7.2. It is also worth noting that, in each case, the

feature(s) which has been extracted seems to contain one extreme variable, such that clustering was extremely difficult, as all assigned cluster groups were calculated in reference to this one point. This does not affect ability to recognise the efficacy of the system, but it does make it harder.

References

[113] Wright, A.H. et al., Implicit Parallelism, *Proceedings of the Genetic and Evolutionary Computation Conference*, Chicago (2003)

Chapter 6

Discussion and Extensions

In the previous chapter, various interesting questions were posed by the results, which warrant further investigation. Various additional comparisons or experiments are undertaken in this Chapter to examine these problems.

6.1 Dividing activity data into fewer bands

An extension experiment was undertaken, where the carcinogen data were re-processed with three activity (carcinogenicity) bands: 1, (representing former bands 1-3), 2 (bands 4-6), and 3 (bands 7-9). New predictions were made using the PCAMS descriptors, and with an ANN, a SVM, and a FSVM. In the previous system (of three floating "quasi-bands"), the allocation of a datum to either of the bands adjacent to the correct band was regarded as a successful prediction. However, with only three (hard) bands, this criterion was obviously removed

Table 6.1 – The application of the techniques to carcinogenicity TD_{50} data with fewer activity bands

q^2	*9 bands*	*3 bands*
ANN	0.84	0.83
SVM	0.87	0.81
FSVM	0.88	0.79

Table 6.1 gives us an intriguing example of the importance of the number of bands selected for the output data. We expect the figures generally to be lower, because there is less flexibility with three bands. [Say, for example, that a carcinogen which should be placed in Band 4, were placed in Band 3. Under the

9-band system this would have been considered successful banding; under the 3-band system it would not. As carcinogens are much more likely to be mis-banded by 1 than 2 bands, this difference becomes mildly important. A potential counter-argument is that, because the bands are larger, there is also a chance that carcinogens previously misbanded by 2 bands might be correctly assigned now that the bands are wider. As there are fewer examples of these 2-misbandings, I feel they would have a lesser effect on this disparity than 1-misbandings.]

However, this data seems to suggest that, in fact, ANNs deal better with fewer bands, whereas SVMs can more accurately process data with a greater number of activity bands. This conclusion has serious ramifications for a user's initial system design: the number of bands into which output activity data are placed has to be compatible with the technique and architecture chosen for the analysis.

6.2.1 Is hydrogen bonding information important?

CoMSIA molecular descriptors are almost identical to CoMFA descriptors, except that they give information about hydrogen bond donors and acceptors. Therefore, interpreting whether hydrogen bonding should be influential on the choice of descriptor (CoMSIA as opposed to CoMFA), is imperative for selection between these techniques. It is well-known that aqueous solubility is predictable using ANNs[114]. An ANN was therefore trained using data from the

AQUASOL database of the University of Arizona and the SCR PHYSPROP Database, making use of first CoMFA and the CoMSIA molecular descriptors. 1156 training examples were used, as was leave-three-out cross validation, but with a restricted processing run. The initial cross-validation values are given in Table 6.2.

Table 6.2 – the prediction of aqueous solubilities through the use of the CoMFA and the CoMSIA molecular descriptor systems.

q^2	*CoMFA*	*CoMSIA*
ANN	0.67	0.73

Using a molecular descriptor which takes into account hydrogen bond donors and acceptors is clearly important in situations where hydrogen bonding is important, such as solubility. Even though CoMFA includes solubility information, the use of the extra information immediately makes the system predict solubility better (at least in its initial stages).

6.3 Practical Extension A: PCAMS applicability

It is always important when undertaking interdisciplinary research, especially involving computer science, not to become too rarefied in one's approach to the application. I therefore want to have a cursory look at how successful the best of these techniques was at predicting the carcinogenicity of half-a-dozen commonly-prescribed chemotherapeutic drugs. I will be comparing the

predicted carcinogenetic with the *International Agency for Research on Cancer*'s Carcinogenicity Classification.

This classification ranks chemicals into 5 groups with the following definitions:

Group 1: The agent is *carcinogenic to humans*.

Group 2A: The agent is *probably carcinogenic to humans*.

Group 2B: The agent is *possibly carcinogenic to humans*.

Group 3: The agent is *not classifiable as to its carcinogenicity to humans*.

Group 4: The agent is *probably not carcinogenic to humans*.

Having acquired the relevant mass spectra for six chemotherapeutics (both alkylating agents and antimetabolites), which represented a wide range of carcinogenicities, I used the FSVM and PCAMS (the best predictor) to ascertain some bands for them. Table 6.3 plots these against the IARC's Classification.

Table 6.3 - IARC carcinogen classification against predicted carcinogen band, using FSVM/PCAMS, for six chemotherapeutic drugs.

Name	*Class*	*Class Subgroup*	*IARC's Class.*	*Predicted Band*
Chlorambucil	Alkylating Agent	Nitrogen Mustards	1	2
Cyclophosphamide	Alkylating Agent	Nitrogen Mustards	1	1
Mechlorethamine	Alkylating Agent		2A	3
Methotrexate	Antimetabolite	Folic Acid	3	8
Fluorouracil	Antimetabolite	Pyrimidine	3	4
Uracil	Antimetabolite		2B	7

Table 6.3, although the comparison is cursory, provides interesting, grounded evidence of the impact of this research and its practical usefulness. Interestingly Methotrexate and Fluorouracil, which have not yet been classified by the IARC, are predicted to be a weak and a medium-strong carcinogen respectively.

6.4 Practical Extension B: Different Classes of Drug

It is not possible to test larger groups of anti-cancer drugs using the novel PCAMS method, as reliable Mass Spectral data cannot be sourced. However, molecular descriptors for mainstream drugs can be calculated using CoMFA, which showed excellent predictivity for oral rat TD_{50} carcinogenicity. These numbers could be tested against our existing, trained FSVM, and the resulting predicted bands compared with the IARC's classification.

Naturally, it would be highly subjective to call this a quantitative analysis by attempting to reconcile the IARC's classification group with the oral rat TD_{50} carcinogenicity band. However some sort of broad, qualitative analysis is certainly possible. This comparison will be interesting as it will show whether many different classes of chemotherapeutic drug can all accurately be modelled.

Table 6.4 - Chemotherapeutic drugs' oral rat TD_{50} carcinogenicity band prediction by FSVM/CoMFA, against IARC classification.

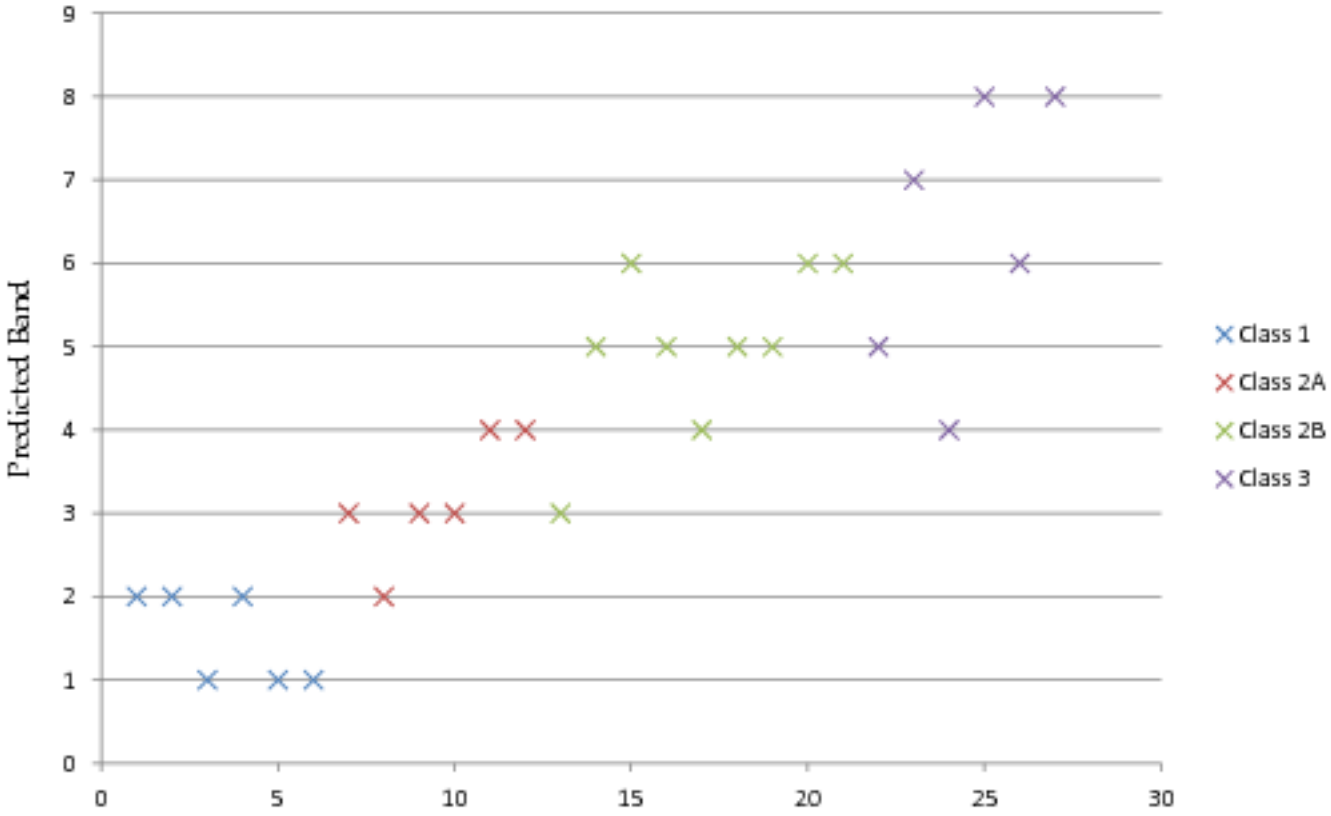
Class	*Subclass*	*Chemical Name*	*IUPAC Name*	*CAS No.*	*Pred.*	*IRAC*
Alkylating Agent	Nitrogen Mustards	Chlorambucil	4-[bis(2-chloroethyl)amino]benzenebutanoic acid	305-03-3	2	1
Alkylating Agent	Nitrogen Mustards	Cyclophosphamide	N,N-bis(2-chloroethyl)-2-oxo-1-oxa-3-aza-2λ[5]-phosphacyclohexan-2-amine	50-18-0	1	1
Alkylating Agent	Nitrogen Mustards	Ifosfamide	3-(2-chloroethyl)-2-[(2-chloroethyl)amino]tetrahydro-2H-1,3,2-oxazaphosphorine 2-oxide	3778-73-2	5	3
Alkylating Agent	Nitrogen Mustards	Melphalan	2-amino-3-[4-[bis(2-chloroethyl)amino]phenyl]-propanoic acid	148-82-3	1	1
Alkylating Agent	Nitrosoureas	Carmustine	1,3-bis(2-chloroethyl)-1-nitroso-urea	154-93-8	3	2A
Alkylating Agent	Nitrosoureas	Streptozocin	1-methyl-1-nitroso-3-[2,4,5-trihydroxy-6-(hydroxymethyl) oxan-3-yl]-urea	18883-66-4	3	2B
Alkylating Agent	Nitrosoureas	Lomustine	1-(2-chloroethyl)-3-cyclohexyl-1-nitroso-urea	13010-47-4	2	2A
Alkylating Agent	Platinum	Cisplatin	dichloroplatinum; azanide	15663-27-1	3	2A
Alkylating Agent		Busulfan	1,4-bis(methylsulfonyloxy)butane	55-98-1	2	1
Alkylating Agent		Dacarbazine	5-(3,3-Dimethyl-1-triazenyl)imidazole-4-carboxamide	4342-03-4	5	2B
Alkylating Agent		Uramustine	5-[bis(2-chloroethyl)amino]-1H-pyrimidine-2,4-dione	66-75-1	6	2B
Alkylating Agent		Actinomycin	2-Amino- 4,6-dimethyl- 3-oxo-3H-phenoxazine- 1,9-dicarboxylic acid bis- [(3,12-dibutpropyl-9, 13,16-trimethyl-4,7,11,14,17-pentaoxo-hexadecahydro- 10-oxa-3a,6,13,16-tetraaza-cyclopentacyclooxadecen- 8-yl)- amide]	50-76-0	7	3
Alkylating Agent		Amsacrine	N-(4-acridin-9-ylamino-3-methoxyphenyl) methanesulfonamide	51264-14-3	5	2B
Alkylating Agent		Mechlorethamine	2-chloro-N-(2-chloroethyl)-N-methyl-ethanamine	51-75-2	3	2A
Alkylating Agent		Thiotepa	N,N',N''-triethylenethiophosphoramide	52-24-4	1	1
Antmetabolite	Folic Acid	Methotrexate	(S)-2-[4-[(2,4-diaminopteridin-6-yl)methyl-methylamino]benzamido]pentanedioic acid	59-05-2	8	3
Antimetabolite	Purine	Mercaptopurine	3,7-dihydropurine-6-thione	50-44-2	8	3
Antimetabolite	Pyrimidine	Fluorouracil	5-fluoro-1H-pyrimidine-2,4-dione	51-21-8	6	3
Antimetabolite		Uracil	Pyrimidine-2,4(1H,3H)-dione	66-22-8	4	2B
Cytotoxic	Anthracycline	Daunorubicin	(8S,10S)-8-acetyl-10-[(2S,4S,5S,6S)-4-amino-5-hydroxy-6-methyl-oxan-2-yl]oxy-6,8,11-trihydroxy-1-methoxy-9,10-dihydro-7H-tetracene-5,12-dione	20830-81-3	5	2B
Cytotoxic	Anthracycline	Doxorubicin	(8S,10S)-10-(4-amino-5-hydroxy-6-methyl-tetrahydro-2H-pyran-2-yloxy)-6,8,11-trihydroxy-8-(2-hydroxyacetyl)-1-methoxy-7,8,9,10-tetrahydrotetracene-5,12-dione	23214-92-8	4	2A
Cytotoxic	Anthracycline	Mitoxantrone	1,4-dihydroxy-5,8-bis[2-(2-hydroxyethylamino)ethylamino]-anthracene-9,10-dione	65271-80-9	5	2B
Cytotoxic		Bleomycin	[3-[[2'-[(5S,8S,9S,10R,13S)-15-[6-amino-2- [[(1S)-3-amino-1-[[(2S)-2,3-diamino-3-oxopropyl]amino]-3-oxo	11056-06-7	6	2B

		cso propyl] -5-methyl[pyrimidin-4-yl]-1,3-[[[(2R,3S,4S,5S,6S)-3-[[(2R,3S,4S,5R,6R)-4-[carbamoyloxy)-3,5-dihydroxy-6-(hydroxymethyl)tetrahydro-2H-pyran-2-yl]oxy] -4,5-dihydroxy-6-(hydroxymethyl)tetrahydro-2H-pyran-2-yl]oxy] (1H-imidazol-5-yl)methyl]-9-hydroxy-5-[(1E)-1-hydroxyethyl]-8,10-dimethyl-4,7,12,15- tetraoxo-3,6,11,14-tetraazaoctadec-1-yl]-2,4'-bi-1,3- thiazol-4-yl]carbonyl]amino]propyl](dimethyl)sulfonium			
Cytotoxic	Hydroxycarbamide	hydroxyurea	127-07-1	8	3
Cytotoxic	Mitomycin	6-Amino-1,1a,2,8,8a,8b-hexahydro-8-(hydroxymethyl)- 8a-methoxy-5-methyl-azirino[2',3':3,4] pyrrolo[1,2-a]indole-4,7-dione carbamate (ester)	50-07-7	6	2B
Topoisomerase Inhibitor	Etoposide	4'-demethyl-epipodophyllotoxin 9-[4,6-O-(R)-ethylidene- beta-D-glucopyranoside], 4' -(dihydrogen phosphate)	33419-42-0	1	1
Topoisomerase Inhibitor	Teniposide		29767-20-2	4	2A

Plotting these data by IARC Class will be interesting, especially as Class 3 represents "not classifiable as to its carcinogenicity to humans," which presumably means that there is no reliable data. Figure 6.1 plots these in groups. Three distinct groups appear indicating that the FSVM/CoMFA system is giving reliable indicators of likely carcinogenicity.

Interestingly, those which are unclassified are mainly predicted to be in Bands 4 or higher. Is the reason they have not been classified because they only exist as very mild carcinogens? This would put them somewhere at the bottom of the list of chemicals to be tested by the various agencies involved in this type of research.

Figure 6.1 – Predicted Oral Rat Carcinogen TD_{50} band, predicted by FSVM/CoMFA, against IARC Carcinogen Class



Although once again fairly cursory, it is important to verify the success of the technique on a variety of chemotherapeutic agents; Figure 6.5 shows pleasing uniform accuracy. We can conclude that the carcinogenicity prediction technique is not only applicable to chemotherpeutic drugs generally, but also appears to work well for each of the different classes of drug, with different applications.

References

[134] Huuskonen, J., Estimation of Aqueous Solubility for a Diverse Set of Organic Compounds Based on Molecular Topology, *Journal of Chemical Information Computer Science* 40, 773 (2000)

Chapter 7

Conclusions

This thesis has arrived at a number of over-arching conclusions about both the viability and the practicalities of using both mass spectra-based molecular descriptors and other systems to predict the carcinogenicity, the mutagenicity and the Cl^- transport inhibition of a variety of molecules, with a pleasing success rate for anti-cancer drugs themselves.

First, it has confirmed that carcinogenicity and mutagenicity are indeed predictable using a variety of artificial intelligence techniques, both supervised and unsupervised, information which will be germane to the research of groups interested in the preliminary screening of candidate anti-cancer drugs. Which of these techniques works best depends on a number of factors, such as which molecular descriptor is used, into how many bands it is desirable to split the activity data, and the necessary accuracy of prediction, versus the amount of time allocated to the problem.

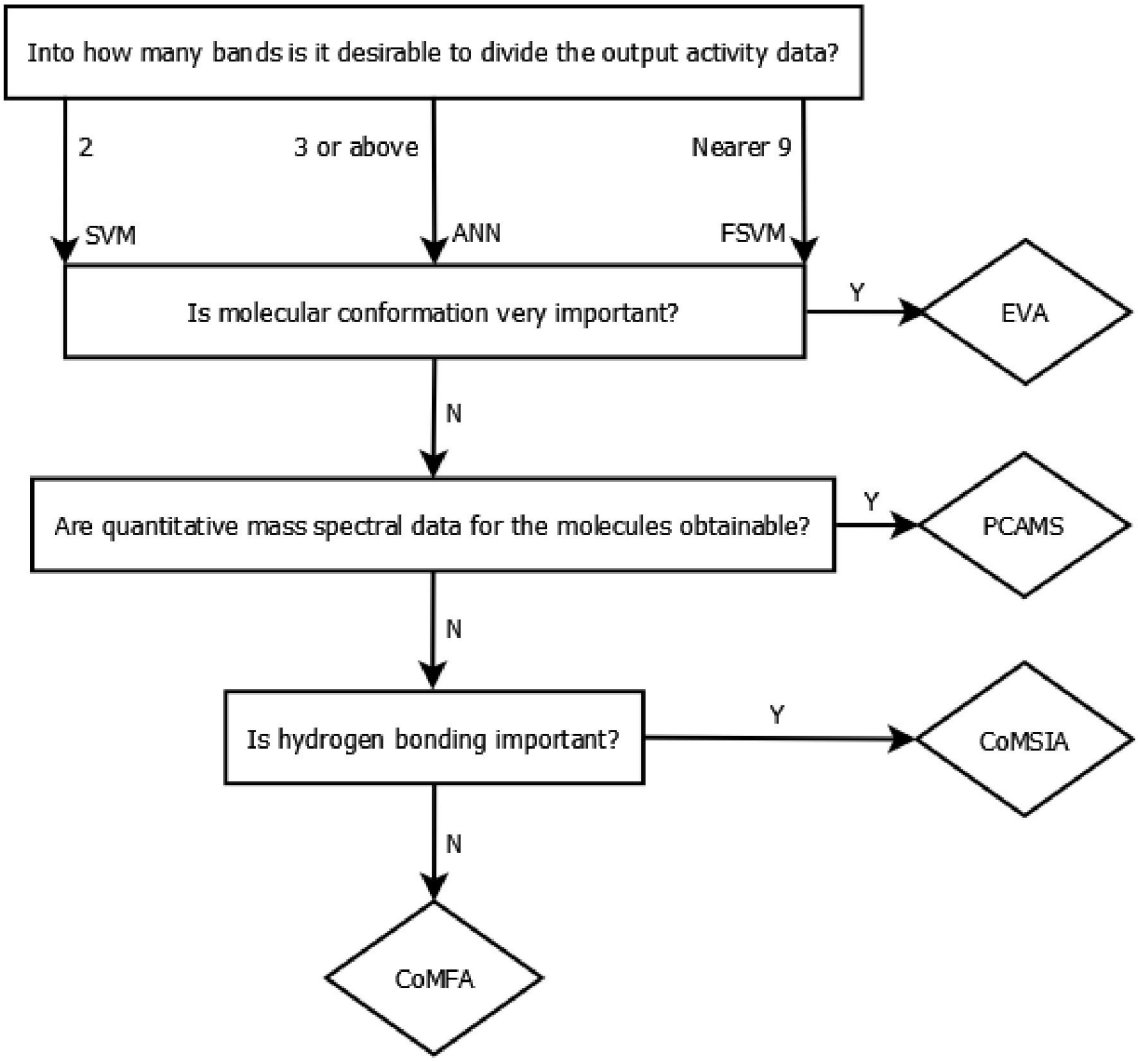
Secondly, it has demonstrated that mass spectra intensities possess great descriptive fidelity and so may be used as reliable molecular descriptors. Moreover, it has shown that reducing the burden of dimensionality is not only important, but imperative, and that selecting this smaller set of orthogonal descriptors is best achieved using Principal Component Analysis as opposed to the selection of a set of the most frequent fragments, or merely the use of every peak up to a number determined by the boundaries of supervised learning.

Third, it has introduced a novel system of backpropagation and shown that it is both more efficient at monitoring a series of connection weights, at least in this type of problem, but with the potential to be applied more generally.

It is possible to come to some preliminary conclusions about how exactly a user should decide upon which technique to use, how they might go about effecting it, and what the likelihood of success would be. Figure 7.1 goes some way to making that first step, although it should be borne in mind by users that this is a preliminary assessment and more work is needed to corroborate this conclusion. In cases where the user does not know the answer to questions in Figure 7.1 or where, for example, the number of groups is between 3 and 9, (s)he ought to take both courses and compare the techniques and descriptors by cross-validation; Figure 7.1 is a guide only.

It is also extremely likely that there are many other important factors which influence the likely predictivity of a system, but which are not highlighted through the use of the dataset included in this thesis. Furthermore, it has been hard to draw inferences about how user-defined parameters should be chosen. It is certainly the case that these should be selected through use of a grid-search or through some standard heuristic method, and not by guess-work; in the case of every technique these parameters make a large difference to either the computation time or the accuracy of the output.

Figure 7.1 – An initial proposed user pathway for the selection of an artificial intelligence technique and of a molecular descriptor system.



Appropriate extensions of this work would include the identification of other factors which would influence the efficacy of AI techniques when applied to the type of problem, the incorporation of new data sets to corroborate this research, and to understand precisely which factors should be included, and further research into whether PCAMS could be applicable to other problems, or whether it is applicable to this area of research alone.