ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

PhD THESIS

Ayça ÇAKMAK PEHLİVANLI

CONSENSUAL CLASSIFICATION OF DRUG/NONDRUG COMPOUNDS FOR DRUG DESIGN

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

ADANA, 2008

ÇUKUROVA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

CONSENSUAL CLASSIFICATION OF DRUG/NONDRUG COMPOUNDS FOR DRUG DESIGN

Ayça ÇAKMAK PEHLİVANLI

DOKTORA TEZİ

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

Bu tez 09/06/2008 Tarihinde Aşağıdaki Jüri Üyeleri Tarafından Oybirliği/~~Oyçokluğu~~ İle Kabul Edilmiştir.

Yrd. Doç. Dr. Turgay İBRİKÇİ DANIŞMAN	Prof. Dr. Seyhan TÜKEL ÜYE	Prof. Dr. Fikret GÜRGEN ÜYE
Prof. Dr. Hamza EROL ÜYE		Yrd. Doç.Dr. Ulus ÇEVİK ÜYE

Bu tez Enstitümüz Elektrik-Elektronik Mühendisliği Anabilim Dalında hazırlanmıştır.

Kod No

Prof. Dr. Aziz ERTUNÇ
Enstitü Müdürü

Bu Çalışma Çukurova Üniversitesi Bilimsel Araştırma Projeleri Fonu ve kısmen TÜBİTAK-EEEAG Tarafından Desteklenmiştir.

Proje No: MMF2006D11 & TUBITAK – TBAG104T505

ABSTRACT

PhD THESIS

CONSENSUAL CLASSIFICATION OF DRUG/NONDRUG COMPOUNDS FOR DRUG DESIGN

Ayça ÇAKMAK PEHLİVANLI

ÇUKUROVA UNIVERSITY
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
INSTITUTE OF NATURAL AND APPLIED SCIENCES

Advisor	Asst. Prof. Dr. Turgay İBRİKÇİ
Co-Advisor	Prof. Dr. Okan K. ERSOY
	Year 2008, 86 pages
Jury	Asst. Prof. Dr. Turgay İBRİKÇİ
	Prof. Dr. Seyhan TÜKEL
	Prof. Dr. Hamza EROL
	Prof. Dr. Fikret GÜRGEN
	Asst. Prof. Dr. Ulus ÇEVİK

Because of the costly, lengthy and laborious drug development process, the aim of the pharmaceutical industry has shifted from traditional trial-and-error process of drug discovery to a structure-based drug design. Although many potential drug candidates may look promising in in vitro studies, they can fail during in vivo stages. Therefore, it is very important to know if the compound is druglike or nondruglike for selection and development of new potential drug candidates. In this thesis, a special consensual approach is proposed for this purpose. The constructed consensual model has a preprocessing unit which consists of transformation of input patterns by random matrices and median filtering to generate independent errors for a single type of classifier, and a postptocessing unit for consensus. Three different combining methods, genetic algorithm, pseudoinverse and equal weight, for postprocessing and four different neural network methods, general regression neural network, adaptive general regression neural network, supervised self organizing map and self organizing global ranking for individual classification were used.

Murcia-Soler and Cherkasov data sets which consist of drug and nondrug compounds were used with descriptors calculated with the Molecular Operating Environment tool.

This thesis presented that the best results to discriminate between the drug-like and the nondrug-like compounds were obtained with the proposed consensus approach with the genetic algorithm which is used for the first time for combining. On the other hand, the self organizing global ranking algorithm which has been applied for the first time on chemical data set also produced high classification results. The consensus approach presented is also used for the first time with chemical data.

In this thesis, on the basis of results obtained from the experiments, new approach is provided for choosing potential drug candidates by using molecular data automatically. Hence, it can be stated that this narrowing considerably reduces the time and effort for the discovery of novel pharmaceuticals.

Keywords: Consensual classification, drug molecules, nondrug molecules, neural networks, rational drug design, chemometrics.

ÖZ

DOKTORA TEZİ

İLAÇ/İLAÇ OLMAYAN BİLEŞENLERİN İLAÇ YAPIMI İÇİN ORTAK KARARLA SINIFLANMASI

Ayça ÇAKMAK PEHLİVANLI

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK ANA BİLİM DALI

Danışman	Yard. Doç. Dr. Turgay İBRİKÇİ
İkinci Danışman	Prof. Dr. Okan K. ERSOY
	Yıl 2008, 86 sayfa
Jüri	Yard. Doç. Dr. Turgay İBRİKÇİ
	Prof. Dr. Seyhan TÜKEL
	Prof. Dr. Hamza EROL
	Prof. Dr. Fikret GÜRGEN
	Yard. Doç. Dr. Ulus ÇEVİK

İlaç yapımının çok uzun, pahalı ve emek gerektiren bir süreç olmasından dolayı farmasötik çalışmalarda kullanılan deneme-yanılma yöntemlerinin yerini "akıllı ilaç geliştirme" yöntemleri almıştır. Laboratuar ortamında çok iyi sonuçlar veren bir çok aday molekül, doğal ortamda tam tersi sonuçlar verebilmektedir. Bu nedenle kimsayal bir bileşiğin ilaç olup olamayacağının tahmin edilebilmesi, potansiyel ilaç aday moleküllerinin seçimi ve geliştirilmesi açısından önemli bir adımdır. Bu amaçla yeni bir ortak karar (konsensüs) yaklaşımı önerilmiştir. Bu model bağımsız hatalara sahip sonuçlar için rasgele matris dönüşümü ve filtreleme ile tek tek elde edilmiş sonuçları birleştirme aşamalarından oluşmuştur. Üç farklı birleştirme metodu, Pseudoinvers, eşit ağırlık ve genetik algoritma, ile tek tek sınıflama sonuçları için dört farklı sinir ağları metodu, genel regresyon sinir ağları, uyarlanmış genel regresyon sinir ağları, kendini organize eden haritalar ve kendini organize eden küresel sıralama kullanılmıştır.

Molecular Operating Environment ile hesaplanan açıklayıcı değişkenlerden oluşan Murcia-Soler ve Cherkasov veri setleri bilinen ilaçlar ve ilaç olmayan bileşenleri içerir. Bu veri setleri çeşitli ön işleme aşamalarından geçirilerek dört farklı metod ile uygulanmıştır. Elde edilen sonuçlar konsensüs için üç farklı birleştirme metodu ile birleştirilmiştir. Birleştirme için ilk defa kullanılan genetik algoritma diger iki metoda göre en yüksek sınıflama başarısını göstemiştir. Ayrıca ilaç verisine ilk defa uygulanan kendini organize eden küresel sıralama metodu da oldukça yüksek sınıflama başarısı sağlamıştır. Kullanılan konsensüs yaklaşımı da kimyasal veriler ile ilk defa kullanılmıştır.

Bu çalışmadaki uygulamalardan elde edilen sonuçlar doğrultusunda, ilaç yapım sürecinin başında kullanılacak ya da incelenecek aday bileşenlerin sayısında ciddi bir düşüş ve ilaç olma olasılığı yüksek adayların seçiminde hız elde edileceği, dolayısı ile zaman ve maliyet açısından bir iyileşme olacağı düşünülmektedir.

Keywords: Konsensüs sınıflama, ilaç ve ilaç olmayan moleküller, yapay sinir ağları, akıllı ilaç tasarımı, kemometri.

ACKNOWLEDGEMENTS

I would like to express my respects and deepest gratitude to both my supervisor Asst. Prof. Dr. Turgay İbrikçi and Prof. Dr. Okan K. Ersoy. They have provided me with their thoughtful guidance, remarkable insights and continuous encouragement.

I wish to express my appreciation to Prof. Dr. Süleyman Güngör, the head of the Department of Electrical and Electronics Engineering, Prof. Dr. Hamit Serbest, and faculty and staff in the Department of Electrical and Electronics Engineering for their supports.

I would like to thank Prof. Dr. Murat Taylı, the dean of Faculty of Engineering and Architecture in İstanbul Kültür University, Prof. Dr. Ümit Karakaş, the head of Computer Engineering Department in İstanbul Kültür University, and all faculty and staff in this department for their patience, supports and creating a very friendly environment for the last two years of my Phd.

I also thank my committee members, Prof. Dr. Seyhan Tükel, Prof. Dr. Hamza Erol, Prof. Dr. Fikret Gürgen and Asst. Prof. Dr. Ulus Çevik for their supports and valuable discussions.

I wish to express my appreciation to Prof. Dr. Anatoli Dimoglo from Gebze Institute of Technology, Assoc. Prof. Dr. Erden Banoğlu from Faculty of Pharmacy in Gazi University, Prof. Dr. Esin Aki Şener from Faculty of Pharmacy in Ankara University, Dr. Wolfram Altenhofen from Chemical Computing Group, Germany, for their guidance to prepare and understand the drug data sets. I also wish to special thanks my sister Asst. Prof. Dr. Gonca Çakmak Demircigil from Faculty of Pharmacy in Gazi University and her husband pharmacist Tolga Demircigil for their patience, supports, and encouragements.

I greatly appreciate my dear parents for their invaluable supports, being my first and best teachers and for giving me so many opportunities. They are always there to take care of me and motivate me.

My special thanks to my sister Hatice for her great supports and encouragements, and one of my best friends Asst. Prof. Dr. Eylem Deniz Akıncı from Statistics in Mimarsinan University, for her valuable discussions and patience in my questions about statistics.

I wish to express my best feelings to my teammate and best friend Irem Ersoz Kaya for her endless encouragement, patience, knowledges, helpful collaboration and morale support.

Last, I wish to very special thanks my husband Burak for everything. He gave me the strongest support through this study. He always believed in me and motivated me. He made me feel brave. He has accompanied me in completing this thesis. I wish to express my appreciation for his tolerance, patience and love.

CONTENTS **PAGE**

LIST OF ABBREVIATIONS

INN	:	International Non-Proprietary Name
USAN	:	United States Adopted Name
RO5	:	Rule of Five
RO3	:	Rule of Three
QSAR	:	Quantitative Structure Activity Relationships
QSPR	:	Quantitative Structure-Property Relationship Analysis
ADMET	:	Absorption, Distribution, Metabolism, Extraction, Toxicity
CMC	:	Comprehensive Medicinal Chemistry
ACD	:	Available Chemicals Directory
WDI	:	World Drug Index
MOE	:	Molecular Operating Environment
SMILES	:	Simplified Molecular Input Line Entry System
MLP	:	Multilayer Perceptrons
ANN	:	Artificial Neural Networks
SVM	:	Support Vector Machine
GRNN	:	General Regression Neural Networks
adGRNN	:	Adaptive General Regression Neural Networks
SOM	:	Self Organizing Map
SOGR	:	Self Organizing Global Ranking
GA	:	Genetic Algortihms
PCs	:	Principle Components

LIST OF TABLES **PAGE**

LIST OF FIGURES **PAGE**

1. INTRODUCTION

The traditional drug design and development process is a complex, time consuming and expensive process. The introduction of a new drug requires approximately 10 to 15 years of research and development and on the order of 0.5 billion dollars to reach market from laboratory [Arciniegas et al., 2000]. Given this situation, the rational, structure-based drug discovery process aims at replacing this traditional trial-and-error process with lengthy, costly and laborious development process.

During the 1990s, the pharmaceutical companies realized that too many compounds were terminated in clinical development due to unsatisfactory pharmacokinetics [Keller et al., 2007]. That is, although many potential drug candidates may look promising in *in vitro* (outside of a living organism) studies, they may fail during *in vivo* (inside a living organism) stages [Jónsdóttir et al., 2005]. Therefore, it is very important to know if the compound is drug-like or nondrug-like for selection and development of new potential drug candidates [Murcia-Soler et al., 2003]. For reaching this information about the compounds is a preclinical stage of drug discovery process and has received increased attention in earlier stages of discovery of drug candidates.

1.1. Chemoinformatics and Bioinformatics

Drug design can benefit from both chemoinformatics and bioinformatics methods. In 1990s, bioinformatics was established and used by most major pharmaceutical companies. Bioinformatics can easily access vast amounts of data with the structured nature of genetic information [Jónsdóttir et al., 2005]. According to Engel, there is no clear separation between chemoinformatics and bioinformatics. Traditionally, chemoinformatics has mainly dealt with small molecules, whereas bioinformatics concerned genes, proteins, and other larger chemical compounds [Engel, 2006]. Since a drug can be simply defined as a small molecule, drug design process can take advantage of chemoinformatics. Table 1.1 briefly illustrates

differences between chemoinformatics and bioinformatics [Wishart, 2005].

Table 1.1. Chemoinformatics vs. Bioinformatics

Chemoinformatics	Bioinformatics
The application of information technology to the study, analysis, distribution and archiving of chemical data	The application of information technology to the study, analysis, distribution and archiving of molecular biological data
Established in the 1960's	Established in the 1990's
Designed for the needs of organic chemists	Designed for the needs of molecular biologists
User-pay, limited public access	Web-based, open access model
Funded by large companies (MDL, Bielstein, Sigma, CAS)	Funded by large government agencies (NCBI, EBI, NIH, GC)

Chemoinformatics has rapidly gained widespread use. It concerns gathering and managing chemical information, and the use of those data to predict the behavior of unknown compounds in silico.

One of the first definitions of chemoinformatics was made by Frank Brown in 1998 [Brown, 1998]:

"The use of information technology and management has become a critical part of the drug discovery process. Chemoinformatics is the mixing of those information resources to transform data into information and information into knowledge for the intended purpose of making better decisions faster in the area of drug lead identification and organization."

According to this definition, chemoinformatics focused on drug design. A much more general definition was made in 1999 by Greg Paris [Paris, 1999]:

"Chemoinformatics is a generic term that encompasses the design, creation, organization, management, retrieval, analysis, dissemination, visualization, and use of chemical information."

A much broader definition of chemoinformatics was published in one of the first textbooks on chemoinformatics by Gasteiger et al. [Gasteiger et al., 2003]:

"Chemoinformatics is the application of informatics methods to solve chemical problems."

1.2. Druglike Compounds

Druglike compounds are molecules which contain functional groups and/or have physical properties similar to the majority of known drugs [Walters et al., 2002]. Since a drug-like compound is a potential starting point to develop a new drug, researchers have made more efforts to extract sets of 'rules', i.e. 'filters' or 'criteria', to identify if a compound is a 'drug-like' compound or not.

The pioneering study on 'rules' for drug-like compounds was carried out by Lipinski in the 1980s and 1990s. As a result of the analysis, Lipinski proposed a very simple set of physicochemical parameter ranges named 'Rule of Five' (RO5) [Lipinski, 1997]. The set of rules was associated with 90% of orally active drugs that have achieved phase II clinical stages [Lipinski, 2004].

Compounds that fail to survive the phase I human toleration studies or the preclinical stage do not receive an International Non-Proprietary Name (INN name) and a United States Adopted Name (USAN name). INN names and USAN names are assigned when a compound enters phase II clinical states. A compound entering into phase II is a real drug in the sense that it has all the properties of a real drug [Lipinski, 2003].

The name of RO5 comes from the parameter cutoff values all containing 5's. There are actually only four simple physicochemical parameter ranges to be drug-like according to the RO5:

- Molecular weight (MWT) $\leq$ 500,
- Lipophilicity ClogP $\leq$ 5 (or Moriguchi logP (MlogP) $\leq$ 4.15)
- H-bond donors $\leq$ 5 (sum of OH and NH)
- H-bond acceptors $\leq$ 10 (sum of O and N atoms)

If a compound fails the RO5, there is a high probability that oral activity problems will occur. On the other hand, it is not guaranteed that a compound which successfully passed the RO5 is drug-like. That is, the RO5 compliant compounds are

not automatically good.

In 2002, the GlaxoSmithKline (GSK) group described an important exception to the RO5 [Veber et al., 2002]. They proposed that compounds with MWT > 500 in violation of the RO5, but with reduced molecular flexibility and constrained polar surface area may also show good oral bioavailability. According to Lipinski, the meaning of 'drug-like' is dependent on mode of administration [Lipinski, 2004]. Veber et al., added the following polar surface area and rotatable bonds range to the RO5 as extensions [Veber et al., 2002].

- Polar surface area ≤ 140 A^2 (or sum of H-donors and acceptors ≤ 12)
- Rotatable bonds ≤ 10

Another exception to the RO5 is compounds that are of natural product origin or have structural features originally derived from natural products as fungi, bacteria and plants. Although the RO5 doesn't work well for these natural origin products, they have provided a number of very successful oral drugs [Lipinski, 2003; Clardy et al., 2004].

Understanding the common features present in a drug molecule and drug-like molecule is helpful to understand the concept of drug-likeness. According to a general analysis of the shapes of molecules with the help of a 2D simple graph approach, any molecule can be examined in four units: rings, linker, side chain and framework shown in Figure 1.1 [Bemis et al., 1996; Bemis et al., 1999].

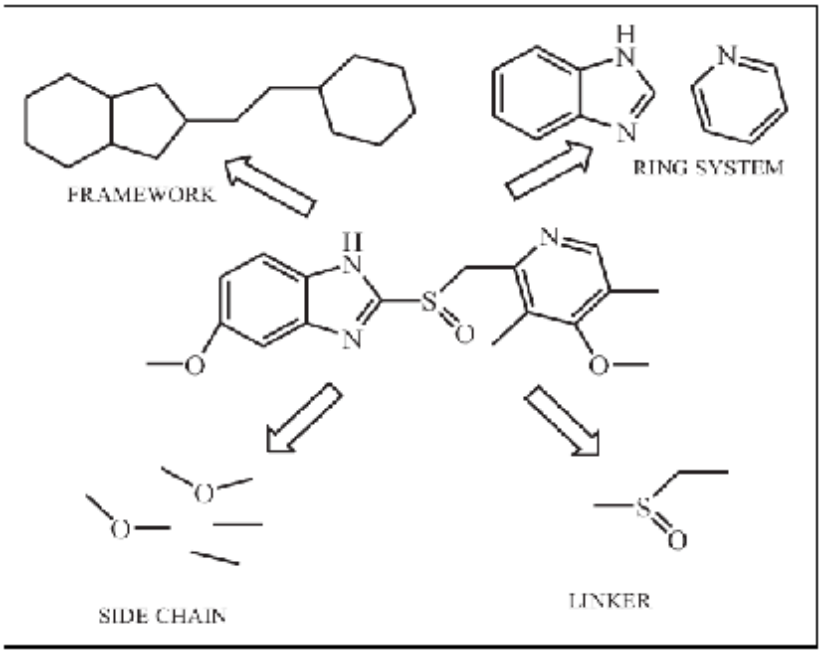


Figure 1.1. Simple structure of molecules (Kadam et al., 2007)

Ring system is the cyclic part within the graph representation of a molecule and sharing edge; linker atoms form the direct path connecting the two rings; side chain atoms are any non-ring, non-linker atoms, and framework is the union of ring systems and linker in a wire frame.

1.3. Lead-Like Compounds

Lead-like concept is based on significantly smaller datasets than drug-like concept. There are two general definitions for the lead-like [Lipinski, 2004]. In one lead-like definition, compounds have reduced property range dimensions compared to the drug. In another definition, lead-like discovery refers to the screening small MWT libraries. These libraries have the compounds according to the following rule of three (RO3):

- Molecular weight (MWT) ≤ 300,
- logP ≤ 3
- H-bond donors ≤ 3
- H-bond acceptors ≤ 3

- Rotatable bonds ≤ 3

Leads have less complex parameters than drugs have. However, there is a strong structural similarity between starting lead and drug. This implies that a quality lead is more likely to result in a real drug than a flawed lead [Lipinski, 2004].

1.4. Comparison of Drug-like and Nondrug-like Compounds

Both simple properties like number of rotatable bond parameters and complex calculations like neural network-based models explain the relationship of structural features to ADME (absorption, distribution, metabolism, extraction) properties [Matter et al., 2001]. Drug-like compounds are expected to meet ADME and toxicology profiles. Using previous analyses of known drugs, including simple property counting schemes (the RO5 and its extensions), machine learning methods, regression models, neural network-based models and clustering methods have all been used to distinguish between drugs and nondrugs [Muegge et al., 2001].

In this thesis, neural network-based models were used to distinguish drugs from nondrugs. The basic assumption is that drugs have distinct values for certain properties. Analyzing in multi-dimensional space, drugs and nondrugs are distinguished as a result of these properties and methods. A chemistry space is defined by calculating a number of descriptors for each molecule and using the descriptor values as points in a multi-dimensional space [Kadam, 2007].

The approach of this study to predict drug-likeness involves examining two sets of compounds, one consisting of known drugs and the other containing compounds known (or presumed) not to be potential drugs. Using computer methods, the properties or characteristics of the known drugs set that differentiate them from the nondrugs set can then be illustrated [Clark et al., 2000].

1.5. Absorption, Distribution, Metabolism, Extraction, Toxicity (ADME-Tox)

ADME-Tox drug properties, namely, absorption, disposition, metabolism, elimination and toxicity, are important and critical for clinical success. Selection of

drug candidates that meet with the best ADMET properties should increase the likelihood of clinical success.

The important drug-like properties can be elucidated by examining an orally available drug. The following events would occur after taking a drug into the body [Li, 2005].

- Dissolution: separation of the active components of a drug from its matrix.
- Absorption: movement of the drug through the intestinal epithelium into the systemic circulation.
- Disposition: entrance of the drug into the system circulation and distribution to various organs and tissues.
- Metabolism: biotransformation of the drug by the drug metabolizing enzymes.
- Biological interaction: interaction of the drug and its metabolites with intended and unintended targets.
- Drug-drug interaction: interaction of the drug with other (coadministered) drugs.
- Elimination: removal of the drug from the body.

Therefore, the desirable drug should have the following properties which are preferred results of the above events: high solubility and absorption, ready distribution to intended target tissues, appropriate metabolic stability and minimal formation of toxic metabolites, extensive interactions with intended targets and minimal interactions with unintended targets, minimal interaction with coadministered drugs, minimal toxicity and an appropriate rate of elimination [Li, 2005]. A drug candidate with these desirable drug-like properties will be probably successful in clinical trials.

1.6. Quantitative Structure Activity Relationships (QSAR)

QSAR analysis is a technique used by pharmaceutical companies in the drug discovery process. It represents an essential part of this process. The underlying assumption behind QSAR analysis is that the variation of biological activity within a

group of compounds can be correlated with the variation of their respective structural and chemical features. That is, there exists a rule or function that predicts a molecule's activity from the values of its physicochemical descriptors. The aim of QSAR analysis is to discover such general rules and equations and predict the bioactivity of molecules based on a set of descriptive features [Liu, 2005]. QSAR problems are challenging for neural networks methods because a typical QSAR data consists of large number of descriptive features (300-1000), often for relatively small number of molecules (50-300) [Embrechts et al., 2002]. This means, in most QSAR applications, the ratio of the number of molecular descriptors to the number of measured activity values is very high. To solve this problem, some feature selection, i.e. feature reduction, approaches are suggested.

A QSAR generalizes how the structure of a compound relates to its biological activity. The usual representation of the data is a matrix in which compounds are defined by individual rows and molecular properties, i.e. descriptors, are defined by associated columns. A QSAR attempts to find consistent relationships between the variation in the values of descriptors and the biological activity for a series of compounds so that this model can be used to evaluate new chemical entities [Richon et al., 1997].

QSAR models based on computational methods like artificial neural networks provide medicinal chemists with mechanism for predicting the biological activity of compounds using their chemical structure or properties. The time to discover a new drug can be significantly reduced by the result of this process [Cedeno et al., 2005].

1.7. Artificial Neural Networks in Drug Design

Artificial neural networks (ANN) enable the investigation of complex, nonlinear relationships, and therefore are ideally suited for use in drug design and QSAR. In the field of drug design, neural networks can be used in the following applications [Terfloth et al., 2001]:

- analysis of multi-dimensional data;
- classification and prediction of biological activity and ADME-Tox

(absorption, distribution, metabolism, excretion, and toxicity) properties;

- lead discovery;
- comparison of compound libraries;
- analysis of similarities and diversity of combinatorial libraries;
- analysis of data from HTS (high throughput screening).

2. RELATED WORKS

Recent studies use computational methods which are mostly statistical or neural networks-based for prediction of drug likelihood for early elimination and identification of candidate compounds during the development process. There is growing interest in the application of neural network-based models to a wide range of chemical problems to predict whether a chemical compound is drug-like or nondrug-like.

The most important studies were started at the end of 1990s. In 1998 Ajay et al. applied Bayesian architecture for separating drug molecules in Comprehensive Medicinal Chemistry (CMC) from nondrug molecules in Available Chemicals Directory (ACD), and 80% of the molecules were classified as drug-like [Ajay et al., 1998]. In the same year, Sadowski and Kubinyi introduced a scoring scheme for classification of drug and nondrug molecules by using Ghose atom types with 77% and 83% prediction accuracy, respectively [Sadowski et al., 1998].

Wagener and von Geerestein developed a model using decision trees by combining Ghose-Crippen descriptors with compounds from ACD and the World Drug Index (WDI) databases as training set. They reached 17.4% error rate on an independent validation dataset. To increase the accuracy, the number of false negatives was reduced by penalizing the misclassification of drugs, so that 92 out of 100 potential drugs were correctly recognized [Wagener et al., 2000]. A feedforward neural network was used by Frimuer et al. with 88% accuracy for both drug and nondrug prediction in 2000. The neural network was trained to distinguish between a set of drug-like and nondrug-like chemical compounds taken from the MACCS-II Drug Data Report (MDDR) and ACD [Frimuer et al., 2000].

In 2001 Muegge et al. proposed a simple selection scheme for drug-likeness based on the presence of certain functional groups, pharmacophore points, in the molecule. The prediction performance of this study was 83.7% for drug compounds and 75% for nondrug compounds [Muegge et al., 2001]. In the same year, Anzali et al. developed a QSAR type method for predicting biological activities with the computer system PASS (prediction of activity spectra for substances). Both data for

drugs from WDI and nondrugs from ACD were used in the training of the model and discriminating between drugs and nondrugs [Anzali et al., 2001].

Brüstle and co-workers developed techniques to separate drugs from nondrugs using a set of molecular descriptors derived from semi-empirical molar orbital (AM1) calculations. The drug set was derived from WDI, and the nondrugs set was from Maybridge database. They used self organizing map projection to distinguish between drugs and nondrugs [Brüstle et al., 2002].

In 2003, Murcia-Soler et al. used a set of topological structural descriptors to discriminate general pharmacological activity. The accuracy results obtained by multilayer perceptrons (MLPs) were 76.36% in the active group and 70.15% in the inactive group. These percentages increased to 80% and 73.44%, respectively, when the undetermined molecules were eliminated [Murcia-Soler et al., 2003]. In the study of Byvatov et al., a support vector machine (SVM) and artificial neural network (ANN) were applied to the datasets prepared by Sadowski and Kubinyi for separating drug molecules from nondrug molecules with 82% correct prediction by SVM, and 80% correct prediction by ANN [Byvatov et al., 2003]. Another study in the same year was done by Manallack et al. They used a consensus neural network-based technique for discriminating soluble and poorly soluble compounds and reached 95% correct classification [Manallack et al., 2003]. Takaoka et al. also developed a neural network method by assigning compound scores for drug-likeness and ease of synthesis based on chemists' intuition. Their method predicted the drug-likeness of drug molecules with 80% accuracy [Takaoka et al., 2003].

Bayram and co-workers applied genetic algorithms (GA) and self organizing map (SOM) for modeling complex QSPR (quantitative structure-property relationship analysis) and QSAR (quantitative structure-activity relationship analysis) problems. They concluded that the combination of GA with supervised SOM showed great potential as a QSAR and QSPR relationship modeling tool [Bayram et al., 2004].

Cherkasov used 30 inductive QSAR descriptors to develop the series of QSAR models enabling in silico distinguishing between antimicrobial compounds, conventional drugs, and drug-like substances. This study resulted in up to 97%

accurate separation of the three types of molecular activities with MLP [Cherkasov, 2006]. In the same year, Karakoç et al. also developed a number of QSAR models using methods of neural networks, k-nearest neighbors, linear discriminant analysis, and multiple linear regressions. They compared methods to recognize five types of chemical compounds that include conventional drugs, inactive drug-likes, antimicrobial substituents, and bacterial and human metabolites. Using a variety of inductive and traditional 2D QSAR descriptors, 20 binary classifiers were created. The study achieved very good separation of chemical compounds in the dataset [Karakoç et al., 2006].

A probabilistic kernel based classifier was used in more recent study by Li et al. with 92.73% correct classification. They used a dataset of 341601 compounds from the WDI and the ACD [Li et al., 2007]. In the same year, the distribution of atom types and their pair-wise combinations in known drugs and nondrugs was examined to quantify the drug-like character of chemical compounds by Hutter. Confirmed drugs were predicted with an accuracy of at least 71% while many falsely predicted nondrugs were found to closely resemble actual drugs [Hutter, 2007].

3. DATA AND METHODS

3.1. Data Sets

In this thesis, there are two different data sets used. The first data set was based on the compilations by Murcia-Soler et al. [Murcia-Soler, 2003]. It consists of 641 compounds. The second one was based on the compilations by Cherkasov [Cherkasov, 2006]. It includes 2684 compounds. In this thesis, only the names of the compounds were taken from both data sets, and the descriptors were calculated by using the Molecular Operating Environment [MOE, 2006].

3.1.1. Murcia-Soler Data Set

Murcia-Soler et al. originally compiled this dataset to discriminate general pharmacological activity with topological methods and artificial neural networks [Murcia-Soler et al., 2003]. The original data set consists of 430 compounds with proven pharmacological activity confirmed by the 12th Edition of Merck Index [Budavari, 1996] and 250 compounds with no pharmacological activity. In this study, only the names of compounds were taken from the original data set. Since this data set was constructed in 2003, each active compound was checked by the 14th Edition of Merck Index [Merck Index, 14th Ed.] to confirm their current therapeutic categories. The final set of compounds used in this thesis is made up of 416 compounds with proven pharmacological activity including different therapeutic categories and 225 compounds without any activity. The set of active compounds belongs to different groups of analgesic, antibacterial, antidepressant, antidiabetic, antifungal, antihypertensive, antihistaminic, antiinflammatory, diuretic, antihyperlipoproteinemic, and sedative categories confirmed by the Merck Index. Table 3.1. illustrates the distribution of the drug and nondrug compounds in the data set.

Table 3.1. Distribution of Drugs and Nondrugs Compounds in the Data Set

Therapeutic Categories	Numb
Drugs	**416**
1. analgesic	66
2. antibacterial	94
3. antidepressant	33
4. antidiabetic	9
5. antifungal	24
6. antihistaminic	30
7. antihyperlipoproteinemic	18
8. antihypertensive	58
9. antiinflammatory	24
10. diuretic	24
11. sedative	36
Nondrugs	**225**
Total	641

3.1.2 Cherkasov Data Set

The Cherkasov data set was constructed from antimicrobial compounds, general drugs and druglike compounds [Cherkasov, 2006]. Cherkasov compiled the set of antimicrobial compounds from several public resources including the ChemIDPlus database [ChemIDPlus, 2006], and an antibiotic database hosted by the Journal of Antibiotics [J. Antibiot., 2006]. The conventional drugs, i.e. general drugs, used in this database were identified from the Merck Index Database [Merck Index, 13.4th Ed]. The druglike compounds used in the data set were randomly collected from the Assinex Gold collection [Assinex Gold, 2004]. These druglike compounds were filtered according to the following criteria:

- Number of H-bond acceptors between 1 and 10
- Number of H-bond donors between 1 and 5
- Molecular weight between 200 and 500 Dalton
- Number of rotating bonds below 12
- Hydrophopicity in the range 1-7
- The total polar surface area below 140 A^2

The final Cherkasov data set used in this study consists of 523 antimicrobial compounds, 959 general drugs and 1202 druglike compounds.

3.1.3 Descriptors

The commercially available Molecular Operating Environment (MOE) modeling package was used to calculate the 2D QSAR descriptors [MOE, 2006]. The MOE modeling package is an interactive, windows-based chemical computing and molecular modeling tool. In this study, the QuaSAR-Descriptor calculation of the MOE was used to calculate the descriptors. There are three descriptor classes, two-dimensional (2D), internal three-dimensional (i3D) and external three-dimensional (x3D) descriptors. Each class includes various descriptors. This study considered only the 2D descriptor calculations. Table 3.2 demonstrates the descriptors partitioned into categories used in this thesis. A complete list of 2D MOE descriptors including short descriptions is given in Appendix.

To use MOE modeling package, SMILES (Simplified Molecular Input Line Entry System) records of the compounds in both datasets were extracted from the PubChem [NCBI, 2007].

SMILES is a line notation for entering and representing compounds. Some simple SMILES examples are as follows:

Ethanol	CCO
Acetic acid	CC(=O)O
Cyclohexane	C1CCCCC1
Pyridine	c1cnccc1
Trans-2-butene	C/C=C/C
L-alanine	N[C@@H](C)C(=O)O
Sodium chloride	[Na+].[Cl-]
Displacement reaction	C=CCBr>>C=CCI

The SMILES representation is a language with a simple vocabulary (atom and bond symbols) and only a few grammar rules. SMILES records are unique for the compounds, understandable and quite compact compared to most other methods of representing compounds [Daylight, 2007].

Table 3.2. 2D QSAR Descriptors in the Data Sets

Categories	Descriptors
Physical Properties	Weight, FCharge, logS, apol, bpol, mr, TPSA, density, vdw_area, vdw_vol, logP(o/w), SlogP, SMR
Subdivided Surface Areas	SlogP_VSA0, SlogP_VSA1, SlogP_VSA2, SlogP_VSA3, SlogP_VSA4, SlogP_VSA5, SlogP_VSA6, SlogP_VSA7, SlogP_VSA8, SlogP_VSA9, SMR_VSA0, SMR_VSA1, SMR_VSA2, SMR_VSA3, SMR_VSA4, SMR_VSA5, SMR_VSA6, SMR_VSA7
Atom and Bond Counts	a_aro, a_count, a_IC, a_ICM, a_nH, b_1rotN, b_1rotR, b_ar, b_count, b_double, b_rotN, b_rotR, b_single, b_triple, chiral, chiral_u, reactive, rings, a_heavy, a_nBr, a_nC, a_nCl, a_nF, a_nI, a_nN, a_nO, a_nP, a_nS, b_heavy, VAdjEq, VAdjMa, lip_acc, lip_don, lip_druglike, lip_violation, opr_brigid, opr_leadlike, opr_nring, opr_nrot, opr_violation
Kier&Hall Connectivity and Kappa Shape Indices	chi0v, chi0v_C, chi1v, chi1v_C, chi0, chi0_C, chi1, chi1_C, zagreb, Kier1, Kier2, Kier3, KierA1, KierA2, KierA3, KierFlex
Adjacency and Distance Matrix	balabanJ, diameter, petitjean, petitjeanSC, radius, VDistEq, VDistMa, weinerPath, weinerPol, BCUT_PEOE_0, BCUT_PEOE_1, BCUT_PEOE_2, BCUT_PEOE_3, BCUT_SLOGP_0, BCUT_SLOGP_1, BCUT_SLOGP_2, BCUT_SLOGP_3, BCUT_SMR_0, BCUT_SMR_1, BCUT_SMR_2, BCUT_SMR_3, GCUT_PEOE_0, GCUT_PEOE_1, GCUT_PEOE_2, GCUT_PEOE_3, GCUT_SLOGP_0, GCUT_SLOGP_1, GCUT_SLOGP_2, GCUT_SLOGP_3, GCUT_SMR_0, GCUT_SMR_1, GCUT_SMR_2, GCUT_SMR_3
Pharmacophore Feature	a_acc, a_acid, a_base, a_don, a_hyd, vsa_acc, vsa_acid, vsa_base, vsa_don, vsa_hyd, vsa_other, vsa_pol
Partial Charges	PEOE_PC+, PEOE_PC-, PEOE_RPC+, PEOE_RPC-, PEOE_VSA+0, PEOE_VSA+1, PEOE_VSA+2, PEOE_VSA+3, PEOE_VSA+4, PEOE_VSA+5, PEOE_VSA+6, PEOE_VSA-0, PEOE_VSA-1, PEOE_VSA-2, PEOE_VSA-3, PEOE_VSA-4, PEOE_VSA-5, PEOE_VSA-6, PEOE_VSA_FHYD, PEOE_VSA_FNEG, PEOE_VSA_FPNEG, PEOE_VSA_FPOL, PEOE_VSA_FPOS, PEOE_VSA_FPPOS, PEOE_VSA_HYD, PEOE_VSA_NEG, PEOE_VSA_PNEG, PEOE_VSA_POL, PEOE_VSA_POS, PEOE_VSA_PPOS, Q_VSA_HYD, Q_VSA_POS

3.1.4. Preprocessing of the Data Sets

The general structure of the data sets used is the matrices in which each row corresponds to a chemical compound, and the columns are the descriptors of the compounds. The input descriptors are multiple features (attributes) of a given compound, and the output label indicates whether the compound has a drug activity or not. Thus, the data set is a X X D matrix, where X is the number of compounds and D is the number of descriptors.

A high-dimensional descriptor vector may represent quite different properties of a compound. The descriptor metric might differ substantially from column to column. This distortion can result in some problems. To avoid potential problems, the data is preprocessed before being used as input for the methods. The descriptor values were normalized according to the z-score shown in Equation 1.

$$z_{.j} = \left(x_{.j} - m_j \right) / s_j \tag{1}$$

where $x_{.j}$ is the j^{th} descriptor for the compounds, m_j is the mean given in Equation 2 and s_j is the standard deviation given in Equation 3 of the j^{th} descriptor.

$$m_j = \frac{\sum_{i=1}^{n} x_{ij}}{n} \tag{2}$$

$$s_j = \sqrt{\frac{\sum_{i=1}^{n} \left(x_{ij} - m_j \right)}{n-1}} \tag{3}$$

where x_{ij} is the i^{th} compound for the descriptor j, n is the number of compounds in

the j^{th} column. This scaling process reveals data with zero mean and unit variance.

The evaluations were done by three-fold cross validation in all of the experiments in this thesis. The normalized data set was splitted into three nonoverlapping blocks of roughly equal size. In each trial two out of these three blocks were used as training data, and the last block was reserved for testing data. The results were than averaged over the three runs. The number of active and inactive compounds distribution of training and testing sets for Murcia-Soler and Cherkasov sets is shown in Table 3.3.

Table 3.3. The Number of Active and Inactive Compounds Distribution of the Murcia-Soler and Cherkasov Data Sets

		Active	**Inactive**	**Total**
Murcia-Soler	Training	278	150	428
	Testing	138	75	213
	Total	**416**	**225**	**641**
Cherkasov	Training	988	802	1790
	Testing	494	400	894
	Total	**1482**	**1202**	**2684**

3.2. Methods

In this study, several methods for classification of drug and nondrug compounds are proposed. These methods employ and extend techniques of artificial neural networks. Before introducing these methods, artificial neural networks are briefly explained.

3.2.1. Artificial Neural Networks

An artificial Neural Network (ANN) is a computer program inspired by biological nervous systems. It is designed to model the information processing similarly to the human brain. An ANN is composed of hundreds of single units, artificial neurons or processing elements, interconnected with weights to solve specific problems. A neural network consists of neurons which are connected by synaptic connections. Learning occurs in a neural network by adjustment of these connections. Each neuron model has weighted inputs, activation function and one output. Figure 3.1 shows the model for a neuron.

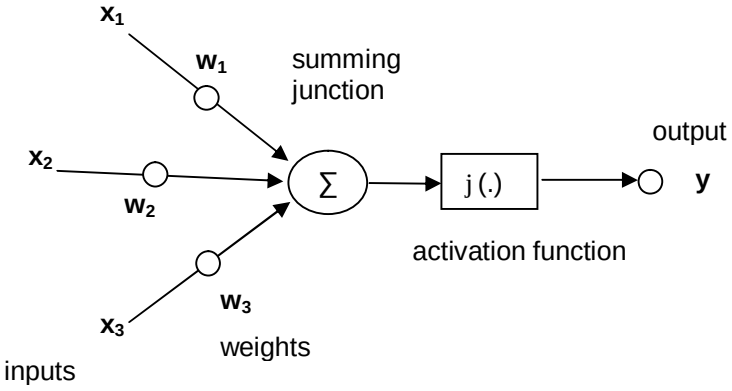


Figure 3.1. Model of an artificial neuron.

First, the synapses of the neuron are characterized by weights which are the strengths of the connections between a neuron and an input. In the next step, an adder sums up all the weighted input signals. This combination is a linear operation. Finally, the sum of the input signals is passed through the activation function which is typically nonlinear[Haykin, 1996]. In Figure 3.1, the arriving signals, called inputs,

multiplied by the connection weights (adjusted) are first summed (combined) given in Equation 4 and then passed through an activation function to produce the output neuron given in Equation 5.

$$summation\ junction = \sum x_i w_i \tag{4}$$

$$y = j\left(\sum x_i w_i\right) \tag{5}$$

These neurons are arranged in a layered structure to form a network. The network has a capability to perform massively parallel computations. Architecture of a general ANN is illustrated in Figure 3.2.

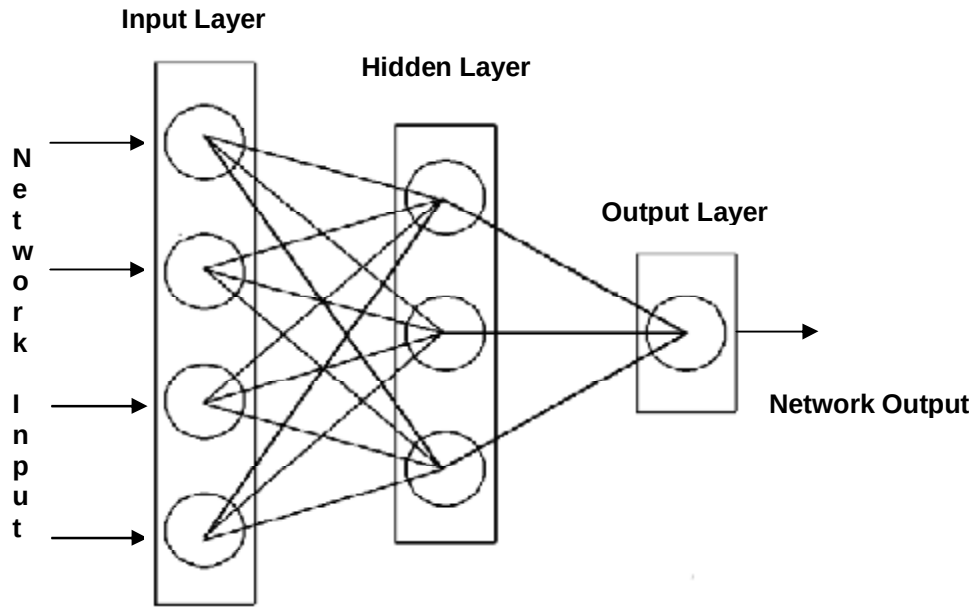


Figure 3.2. Architecture of artificial neural network.

There are three different types of networks, each corresponding to a particular abstract learning task. These are supervised learning, unsupervised learning and reinforcement learning. The supervised and unsupervised networks are widely used for classification tasks and for function approximation in many fields of chemistry and bioinformatics [Schneider et al., 1998]. In this thesis, the main objective is to

classify compounds as a drug or nondrug. Because of this reason, supervised neural networks and unsupervised architecture with supervised module have been used.

The objective in supervised learning is to predict the target values from the input variables. The supervised neural network (SNN) reads the input and output values in the training data set and makes changes to the values of the weighted connections to reduce the difference between the predicted output and target output values. This is repeated in many iterations until the network reaches specified level of accuracy or stopping criteria. The general flow diagram of SNN is given in Figure 3.3.

A supervised learning scheme is implemented using a database which consists of inputs and corresponding known targets pairs. SNN learns by external supervision in the way that each output unit is told what its target response to input signals ought to be.

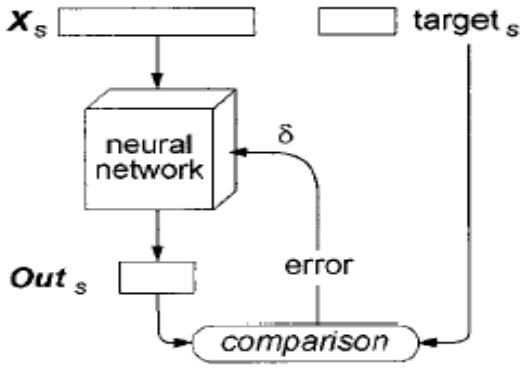


Figure 3.3. Outline of supervised training of a neural network (Gasteiger et al., 2003)

An unsupervised learning scheme does not have a supervisor. That is, there is no knowledge or definition about the output. An unsupervised neural network (UNN) automatically recognizes clusters of data in a given data set based on a rational representation of inputs and a sensitive similarity measure [Schneider et al., 1998]. A UNN is used to project the instances from a high-dimensional space into a low-dimensional space such as a two-dimensional plane shown in Figure 3.4.

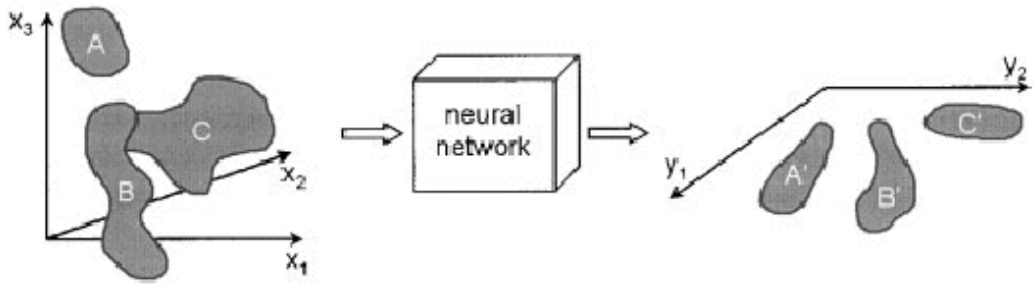


Figure 3.4. Projection of a dataset from high-dimensional space into lower-dimensional space, typically a 2D plane (Gasteiger et al., 2003)

In UNN structure output neurons represent possible data clusters or categories. In unsupervised training, the network is provided with inputs and no desired outputs. The decision about what feature it will use to cluster the input data is made by the system. This property is referred to as self organization or adaptation which may involve competition between neurons, co-operation or both [Agatonovic-Kustrin et al., 2000].

ANNs are used in many different fields of applications including pattern recognition, classification, control systems and identification. In this thesis, neural networks are used for classification of pharmaceutical data. Neural networks requires less formal statistical training, are able to detect complex non-linear relationships between independent and dependent variables, since they are distribution-free and no prior knowledge is needed about the distributions of the classes [Benediktsson et al., 1990].

In the following sections, all the supervised and unsupervised methods used in this study are explained.

3.2.2. General Regression Neural Networks (GRNN)

General Regression Neural Network (GRNN) is a one-pass supervised learning algorithm which can be used for estimation of continuous variables. It is a memory-based feedforward neural network with highly parallel structure [Specht, 1991].

The computation of the GRNN has no iteration. The root of the GRNN is based on regression analysis, which is one of the most commonly used statistical

techniques. To perform regression, a certain function is needed. Analyzing the data points and determination of the parameters of the function are the tasks performed. Specht proposed the probability density function of the observed data for this purpose [Specht, 1991].

The GRNN is based on the probability density function *f(X, y)* which can be estimated by a Parzen estimator [Parzen, 1962]. The regression of a dependent variable Y is to be performed on an independent variable X. The GRNN computes the most likely value of Y based on a finite set of possibly noisy observations of X and the associated values of Y.

Let x be a vector random variable and y be a scalar random variable. The joint continuous probability density function (pdf) of x and y is *f(x, y)*. The basic regression equation is given in Equation 6.

$$y(x) = E\left[y \middle| X\right] = \frac{\int_{-\infty}^{+\infty} yf\left(X, y\right)dy}{\int_{-\infty}^{+\infty} f\left(X, y\right)dy} \tag{6}$$

where *E[y|X]* is the conditional mean of the dependent variable y given X which is a particular value of the random independent variable x.

If the expression of the relationship between x and y is in a functional form with parameters, the regression will be parametric. The nonparametric estimation methods are generally more appropriate to use when there is no real knowledge of the functional form between the dependent and independent variables. That is, the probability density function is empirically determined from a sample of the data points of x and y using Parzen window estimation [Parzen, 1962].

According to Parzen, the estimation of the pdf *f(X, y)* for the univariate case is given in Equation 7.

$$f(X,Y) = \frac{1}{ns^2} \sum_{i=1}^{n} K\left(\frac{X - X_i}{s}\right) K\left(\frac{Y - Y_i}{s}\right) \tag{7}$$

where s is a kernel width (smoothing parameter), n is the number of samples in the training set, and $K(x)$ is a kernel function. In Equation 8, the Gaussian form of $K(x)$ is shown [Tomandl et al., 2001].

$$K(x) = \frac{1}{\sqrt{2p}} e^{-\frac{1}{2}x^2} \tag{8}$$

With the choice of the Gaussian kernel for $K(x)$ given in Equation 8, the $f(X, y)$ estimator holds in the multivariate case, Cacoullos extended Parzen's results in the following form [Cacoullos, 1966], [Parzen, 1962]:

$$f(X,Y) = C\frac{1}{n} \sum_{i=1}^{n} \exp\left(-\frac{\left(X - X^i\right)^T \left(X - X^i\right)}{2s^2}\right) \exp\left(-\frac{\left(y - Y^i\right)^2}{2s^2}\right) \tag{9}$$

where C is a constant given in Equation 10, X^i is the i^{th} training vector, Y^i is the corresponding value of y, and d is the dimension of the vector X.

$$C = \frac{1}{(2p)^{(d+1)/2} s^{(d+1)}} \tag{10}$$

The basic Nadaraya-Watson kernel estimator given in Equation 11 is obtained by substituting Equation 9 into Equation 6 and integrating over y [Specth, 1991].

$$Y(X) = \frac{\sum_{i=1}^{n} Y^i \exp\left(-\frac{\left(X - X^i\right)^T \left(X - X^i\right)}{2s^2}\right)}{\sum_{i=1}^{n} \exp\left(-\frac{\left(X - X^i\right)^T \left(X - X^i\right)}{2s^2}\right)} \quad (11)$$

The GRNN structure has four layers shown in Figure 3.5. The first layer named input layer is where all of the measurement variables X are fully connected to the pattern layer.

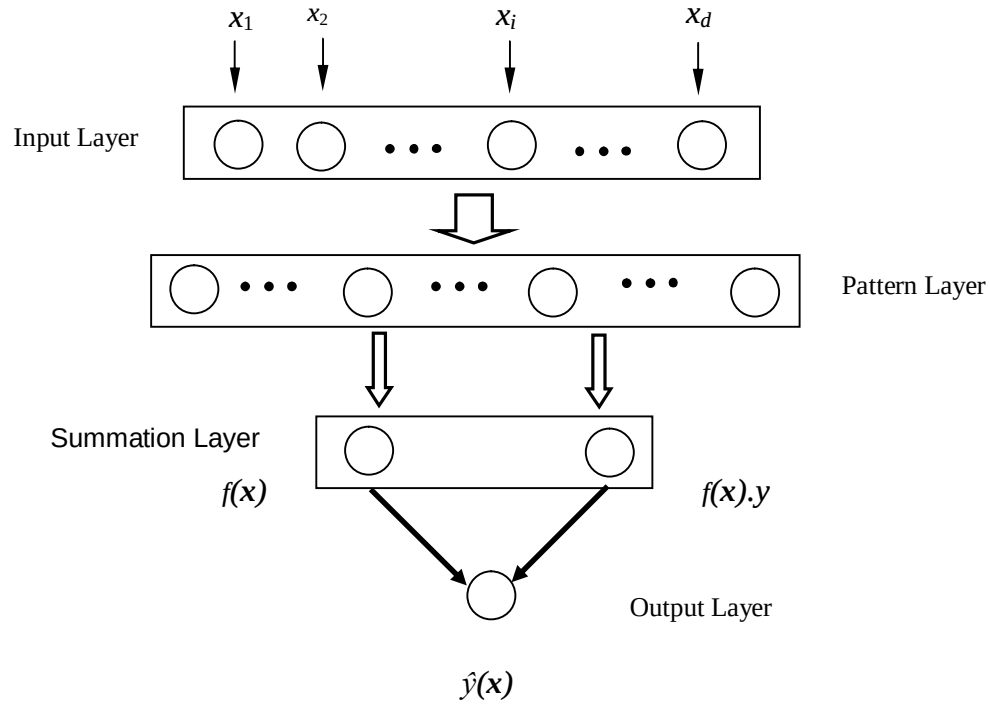


Figure 3.5. Structure of the GRNN network

The pattern layer where a nonlinear transformation is applied on the data from the input space is dedicated to one cluster center for each pattern. There is a subtraction between a new vector X entered into the network and the stored vector representing each cluster. The sum of the squares of the differences is fed into a

nonlinear activation function.

The summation layer holds the pattern layer's outputs. Multiplication of a weight vector and a vector composed of the signals from the pattern layer is performed in the summation layer. This layer generates an estimation of *f(X)K* weighted by the number of observations each cluster center represents [Specht, 1991]. Since K is a data independent constant, there is no need to compute it. The sum of the multiplication of *f(X)K* by Y^j associated with cluster center X^i is estimated in this layer.

The output layer divides *Y f(X)K* by *f(X)K* to provide the desired estimate of *Y*.

3.2.3. Adaptive General Regression Neural Network (adGRNN)

The smoothing parameter, i.e. kernel width, is the only parameter that is used in the GRNN algorithm. Its value is common for each dimension of the GRNN discussed in Section 3.2.2. Specht has suggested that adapting separate s's for each dimension of the probabilistic neural network improves generalization accuracy [Specht, 1992]. In the light of this result, the separate smoothing parameter for each dimension is also applied for the GRNN algorithm. The modified GRNN is called Adaptive GRNN (adGRNN) in this thesis.

The proposed method adGRNN has a separate s_j in each dimension. Adaptation is accomplished by obtaining each s_j according to an optimization criterion which minimizes the mean square error using hold-one-out method of validation. In this thesis the genetic algorithm was used to estimate s_j. Equation 11 given in the previous section is replaced by Equation 12.

$$Y(X) = \frac{\sum_{i=1}^{n} Y^i \exp\left(-\frac{1}{2} \sum_{j=1}^{d} \left(\left(x_j - x_j^i \right) / s_j \right)^2 \right)}{\sum_{i=1}^{n} \exp\left(-\frac{1}{2} \sum_{j=1}^{d} \left(\left(x_j - x_j^i \right) / s_j \right)^2 \right)} \tag{12}$$

To obtain separate s_j for each dimension, the classification accuracy is determined using GRNN with the hold-one-out method. In the hold-one-out method, one input vector of the entire training data set is removed as a testing vector and the remaining data set is used as the training data set. The GRNN and the genetic algorithm (GA) were used together to estimate the optimum s_j for the system. Each time one input vector is chosen as a testing vector while the rest of the training data set is used for training to get the error with the random trial sigma array selected from a population produced by the GA. This chosen trial sigma array is used with each testing vector removed from the original training data set according to the hold-one-out method to obtain the squared difference between the testing value and the predicted value for all the vectors in the data set. This process is repeated for all sigma arrays in the population. Using the hold-one-out method the GRNN is applied to calculate the mean square error for a number of iterations given at the beginning of the application. The sigma array for which the sum of the mean squared difference is the minimum of all the mean squared differences is the optimum sigma array to be used for the predictions.

3.2.4 Genetic Algorithm (GA)

Genetic algorithm, introduced by John Holland in 1975, is a stochastic global search method. It mimics the natural evolution process [Holland, 1975]. This method has been used in various fields, such as analysis of time series, work scheduling, face recognition, etc. [Coley, 1999], [Rothlauf, 2006].

The most important feature of GA is that it investigates many possible solutions simultaneously [So et al., 1996]. In GA, each individual (unknown) parameter is encoded as strings, i.e. chromosomes. A chromosome can be represented with binary, integer or real values shown in Figure 3.6. In this study real-valued representation has been used.

Chromosome A	1 0 0 1 0 1 1 0
Chromosome B	1 0 1 1 1 0 0 0

(a)

Chromosome A	7 8 9 4 1
Chromosome B	8 7 9 1 4

(b)

Chromosome A	1.2324 3.5354 4.6465 3.5556
Chromosome B	3.5354 4.6465 1.2324 3.5556

(c)

Figure 3.6. Representation of chromosomes (a) binary representation, (b) integer representation,(c) real-valued representation

A population is a set of chromosomes. The chromosome data structure stores population in a single matrix of size *N X D*, where N is the number of individuals in the population, and D is the length of the genotypic representation of the individuals. In this study, D is the dimension of an input vector in data set.

An objective function determines the performance, or fitness, of individual members of population, i.e. characterizes individual's performance in the problem domain. This resembles that an individual's ability to survive in its environment in the natural world. The value of the objective function for each individual in each generation is considered to be the fitness of the individual. A new population is obtained through selecting the more fit individuals [Altunkaynak et al., 2004]. In this thesis, the objective function's task is to calculate mean square errors produced by a certain neural network with each individual in the population. The fitness is a function that is used to transform the objective function value into a measure of relative fitness shown in Equation 13.

$$F(x) = g(f(x)) \tag{13}$$

where *f(x)* is the objective function, *g(.)* transforms the value of the objective function to a non-negative number and *F(x)* is the resulting relative fitness.

Selection is the process of determining the individuals from the population to be worked on in order to apply the genetic operators, crossover and mutation. It also decides the number of trials and offsprings, i.e. new individuals, that each individual will produce. The individuals are selected through a fitness-based process. Thus the selection strategy is related to the level of fitness values for individuals actually existing in the population. To find fitness level, ranking or scaling functions can be used.

There are various selection strategies based on the fitness level, such as rank-based selection, roulette wheel selection, stochastic universal selection, local selection, truncation selection, tournament selection, etc. In this study, stochastic universal selection was used [Chipperfield et al., 1994]. It rates the fitness of each solution and decides the selection according to the best solution.

The Darwin's principle of "survival of the fittest" is the evolutionary concept in GA. This process is carried out with genetic operators [Altunkaynak et al., 2004]. The most widely used genetic operators are reproduction, crossover and mutation.

The reproduction operator simply copies a chromosome from the parent to the next generation. Thus, after reproduction the offspring is identical with its parent:

Parent = 0 1 0 1 0 1 1 0

↓ reproduction

Offspring = 0 1 0 1 0 1 1 0

The purpose of the crossover operation is to create new individuals that have some parts of both parent's genetic materials. That is, crossover operation generates two individual chromosomes from two existing chromosomes. If the chromosomes are encoded by binary representation, the crossover process is as in Figure 3.7.

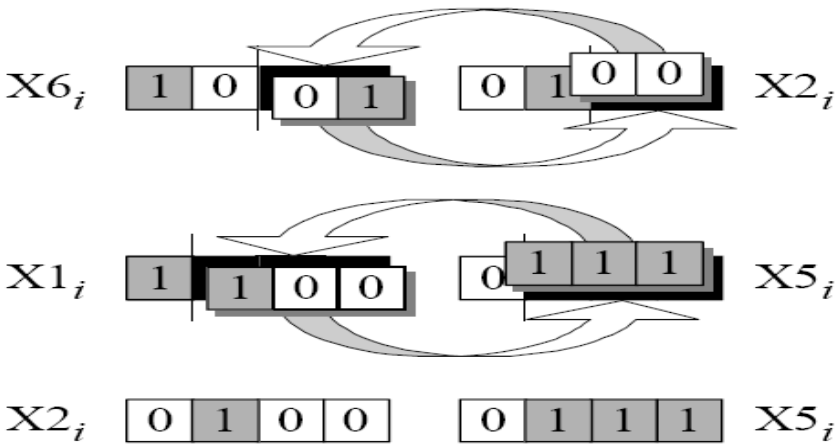


Figure 3.7. Crossover operations for the binary encoded chromosomes (Negnevitsky, 2005)

The crossover operations are used to swap genetic information between pairs of individuals. The simplest crossover operation is one-point crossover. Let P_1 and P_2 be the two parent binary strings with length *l*, and *i* be the position index of these strings. If *i* is selected uniformly at random between 1 and [*l*-1] , and the genetic information is swapped between the individuals with reference to this point, then two new offspring strings are produced. For example let the crossover point *i = 5*.Then we get the following:

P_1 =	1	0	0	1	0	1	1	0
P_2 =	1	0	1	1	1	0	0	0
I	*1*	*2*	*3*	*4*	***5***	*6*	*7*	*8*
O_1 =	1	0	0	1	0	**0**	**0**	**0**
O_2 =	1	0	1	1	1	**1**	**1**	**0**

Typical crossover operations are one-point crossover, two-point crossover, and cut and splice crossover shown in Figure 3.8.

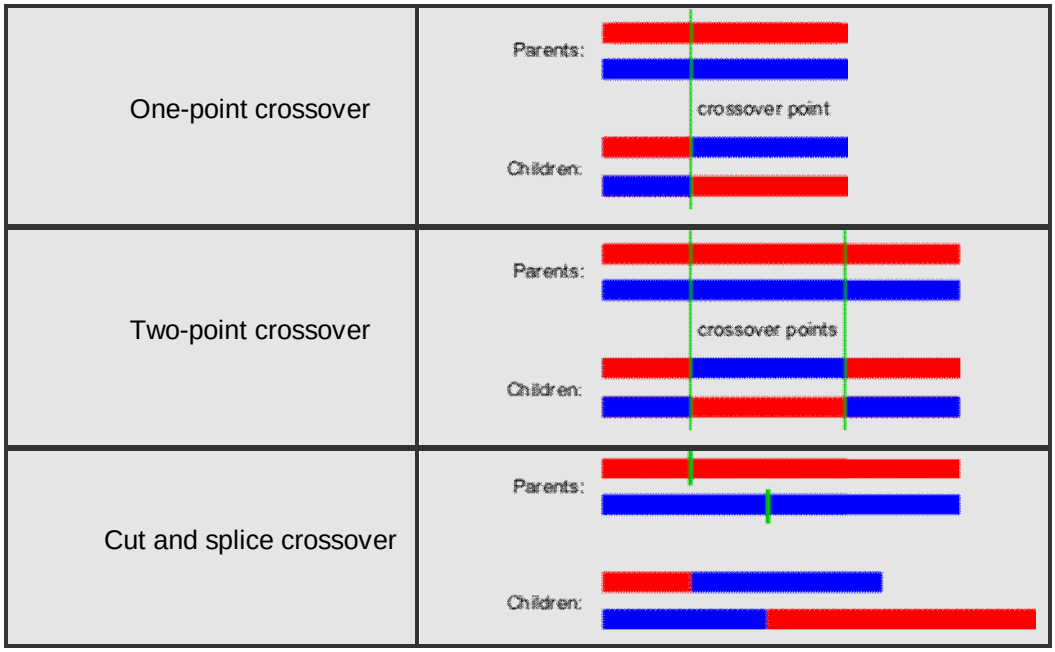


Figure 3.8. Typical crossover operations (Wikipedia, 2007)

Another genetic operation is mutation. In natural evolution, mutation is a random process where one string of gene is replaced by another to produce a new generic structure. The example for mutation is shown in Figure 3.9.

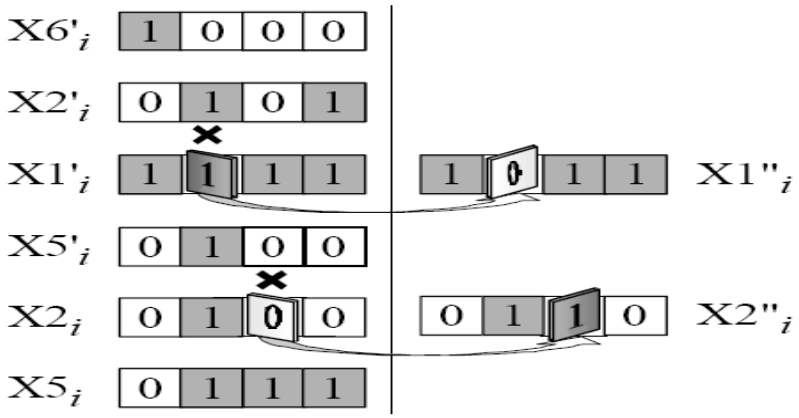


Figure 3.9. Mutation operations for the binary encoded chromosomes (Negnevitsky, 2005)

The basic design of the genetic algorithm is summarized in the flow diagram shown in Figure 3.10

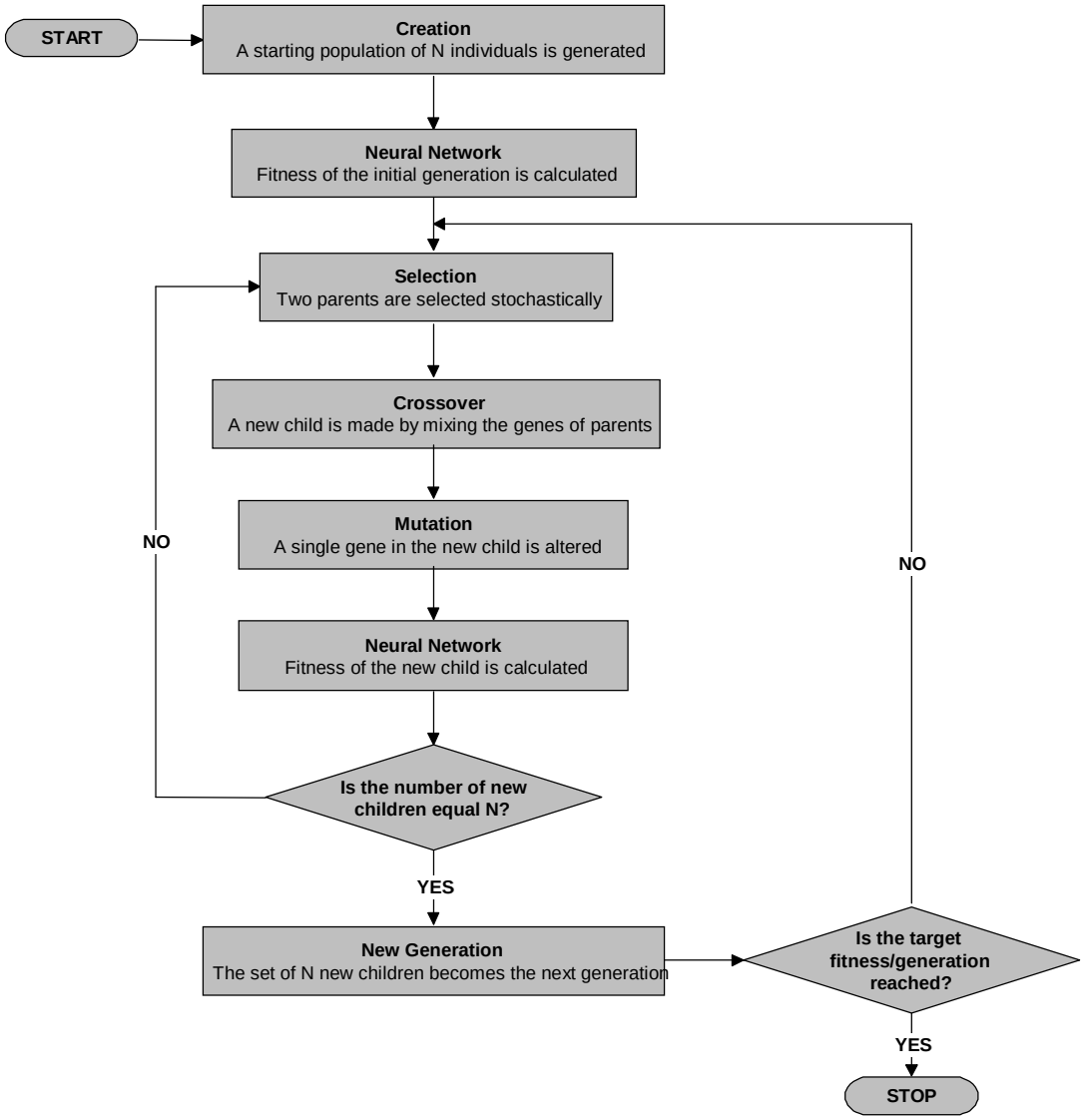


Figure 3.10. Flow diagram of the GA.

3.2.5 Self Organizing Map (SOM)

The pioneering studies on the Self Organizing Map (SOM) were carried out by Kohonen in the early 1980s [Kohonen, 1990]. The SOM is a type of unsupervised neural network that can be used in various areas that especially require visualization and dimension reduction of large, high-dimensional data sets. The SOM produces a mapping from a multidimensional input space onto a one or several dimensional topology-preserving map of nodes. In order that neighboring nodes respond to

similar input patterns, the main feature of the SOM is to preserve a topological order in the map.

The SOM algorithm has a set of non-interconnected nodes that are ordered according to a certain topology. These nodes are represented by n-dimensional vectors, arranged in a 2D array. Each node is surrounded by other neighborhood nodes arranged in a hexagonal or rectangular lattice [Haykin, 1994]. These arrangements are shown in Figure 3.11.

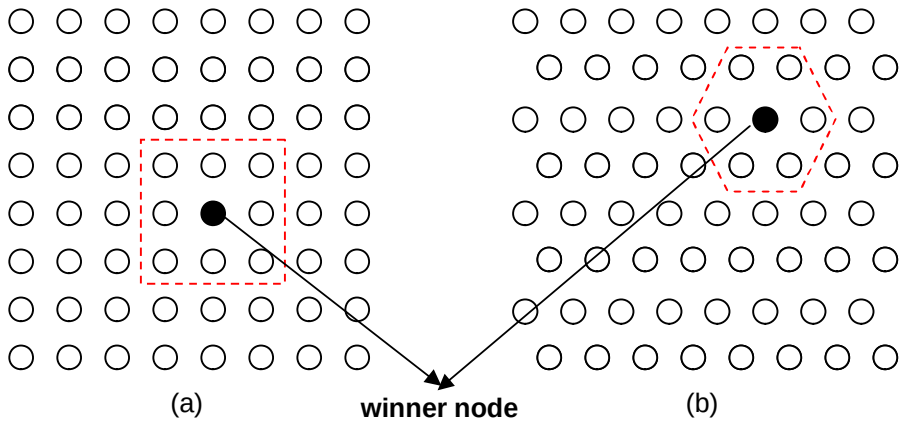


Figure 3.11. 2D lattice shapes (a) rectangular, (b) hexagonal

The training procedure of the SOM involves two stages, first finding a winner node, and next the adaptation of the winner and its neighborhood nodes. The first stage is similar to competitive learning. The output nodes of the network compete among themselves to be activated. The result is only one output node, called winner, at any one time. During training of competitive learning, the winner node is determined with the highest activation to a given input vector, and it is moved closer to the input vector. To decide the winner node, the distance is calculated between the input vectors and the codebook vectors. The other nodes are left unchanged in competitive learning. Thus, only the codebook vector belonging to the winning node is updated at each iteration in competitive learning. In the SOM learning process, besides the codebook vectors belonging to the winning node, its neighbors are also updated at each iteration.

Every node is represented by a codebook vector $c_j = \left[c_{j1}, c_{j2}, ..., c_{jd} \right]$ in the d-dimensional lattice shaped topology. A sample input vector x(t) is randomly chosen from the training set at each training step t. During the first step of the learning process, the winning node m is determined according to the minimum distance computed by the following equation.

$$m = \arg\min_j \left\| x(t) - c_j \right\|, \quad j \in \{1, ..., N\} \tag{14}$$

In Equation 14, N represents number of nodes and c_j is the codebook vector of the j^{th} node.

Subsequently, at the second stage of the training, the winner node, and the winner-surrounded neighborhood nodes are updated by the learning rule as shown in Equation 15.

$$c_j(t+1) = \begin{cases} c_j(t) + a(t)\left[x(t) - c_j(t) \right], & \forall j \in N_m \\ c_j(t), & otherwise \end{cases} \tag{15}$$

where $c_j(t+1)$ is the updated codebook vector, m is the winning node, N_m is its neighborhood, and $a(t)$ is the learning rate decreasing with time.

These two iterative stages are repeated until all vectors belonging to the training set are processed.

The basic SOM algorithm is an unsupervised learning algorithm. In this study, the SOM algorithm has been used as supervised SOM by adding a new labeling module explained in Section 3.2.7.

3.2.6. Self Organizing Global Ranking (SOGR)

The Self Organizing Global Ranking (SOGR) algorithm is based on a new concept of global neighborhood generated by ranking. The SOGR was designed to reduce computational load and complexity of neighborhood concepts of the SOM algorithm [Saglam et al., 2004]. The SOGR learning is very similar to the SOM learning. The main difference between them is the neighborhood concept. The SOGR algorithm is not characterized by the formation of a topographic map of the input vectors. The neighbor nodes are chosen globally based on similarity ranking. Hence, to be chosen as a neighbor, it is not necessary to be geometrically close to the winning node, i.e. neighbor nodes can be everywhere.

The SOGR algorithm can be evaluated in two consecutive stages. In the first stage, the SOGR acts as an unsupervised learning algorithm, i.e. no information about the class labels of the input vectors. To present clusters of input vectors, nodes are moved in the feature space by updating dynamically. In the second stage for supervised learning, the SOGR algorithm assigns class labels to the reference vectors. Then the SOGR algorithm acts as a supervised learning algorithm. The supervised SOGR algorithm includes a labeling module as given in Section 3.2.7.

The first stage generally consists of initialization, identifying the closest node and adaptation steps. The details of these steps are given in the following algorithm.

The SOGR Algorithm

It is assumed that the size of the codebook vectors is predefined as the number of nodes N. Let the initial codebook vector be $c_j(t)$ at time *t*, and the input vectors in the training set be represented by $x(t)$ at time *t*.

1. Initialization: In this step, codebook vectors can be chosen randomly among the input vectors. The initial codebook vectors are $c_j(t)$ at $t = 0$, $j = 1, 2, ..., N$.

2. All the input vectors in the training set are uploaded to the network randomly one at a time. The time index t is increased by 1 when each $x(t)$ is uploaded to the network. For each input vector x_i in the training set, one or more nodes are adjusted as follows:

- **Identifying Winning Nodes:** In the SOGR algorithm, the closest node is not enough to update codebook vectors. Besides the closest node, global R closest nodes are determined. According to the distance values between the randomly selected input vector and the codebook vectors, the R nodes with the smallest value of $\|x_i - c_j(t)\|$ are found. The preferred distance measure in this study is Euclidean distance. The neighborhood of the input vector is constituted by these R nodes.
- **Adaptation of Codebooks:** In this step, codebook vectors are adjusted using the update rule. The adaptation is applied to the R closest nodes according to the following update rule.

$$c_j(t+1) = \begin{cases} c_j(t) + b_j(t)\left[x(t) - c_j(t)\right], & \forall j \in \Gamma \\ c_j(t), & otherwise \end{cases} \quad (16)$$

In Equation 16, Γ represents the set of indices of the R closest nodes. R is the neighborhood size. $b_j(t)$ is the learning rate which decreases with time. It is suggested that the learning rate should be smaller for larger neighborhood size R [Saglam et al., 2004].

3. Assigning Labels to Nodes: In order to use the SOGR in supervised learning, it is necessary to label the codebook vectors with appropriate classes. This labeling process is given in the following section.

3.2.7 Labeling Module

In order to use the SOGR and the SOM learning algorithms in supervised learning, it is necessary to know the labels of the codebook vectors with appropriate

classes. For this purpose, counters are added to each codebook vector. The counters are attached at the end of the result matrix of the codebook vectors shown in Equation 17.

$$C = \begin{bmatrix} c_{11} & c_{12} & c_{13} & \mathbf{L} & c_{1D} & 0 & 0 \\ c_{21} & c_{22} & c_{23} & \mathbf{L} & c_{2D} & 0 & 0 \\ \mathbf{M} & \mathbf{M} & \mathbf{M} & \mathbf{M} & \mathbf{M} & \mathbf{M} & \mathbf{M} \\ c_{P1} & c_{P2} & c_{P3} & \mathbf{L} & c_{PD} & 0 & 0 \end{bmatrix} \tag{17}$$

In the result matrix, the last two columns represent the counters initialized to zero. The total number of counter columns should be equal to the number of classes.

Then, the distances between the input vectors and the all updated codebook vectors are calculated. The nearest codebook vector is decided and a counter for the corresponding class is increased by one in the result matrix. This process is applied for all the input vectors in the training set. The class label of the maximum count is assigned as the label of the codebook vector.

The following numerical example is given for illustration. Let training data have 4 classes, which is the number of the elements of the counter.

- Initialize the count to zero for each node in the result matrix as shown in Equation 18.

$$\begin{matrix} & & & & & 1 & 2 & 3 & 4 \end{matrix}$$

$$C = \begin{bmatrix} c_{11} & c_{12} & c_{13} & \mathbf{L} & c_{1D} & 0 & 0 & 0 & 0 \\ c_{21} & c_{22} & c_{23} & \mathbf{L} & c_{2D} & 0 & 0 & 0 & 0 \\ c_{31} & c_{32} & c_{33} & \mathbf{L} & c_{3D} & 0 & 0 & 0 & 0 \\ c_{41} & c_{42} & c_{43} & \mathbf{L} & c_{4D} & 0 & 0 & 0 & 0 \end{bmatrix} \tag{18}$$

- Find the closest node to the input vector for all input vectors in the training data, that is, find the node with the index $j^* = \arg\min_j \left\| x_i - c_j(t) \right\|$ and add 1 to the

column of the corresponding class at the corresponding row of the winning codebook vector. For example, if the closest codebook vector of the input vector x is c_1, and x belongs to class 2, the counter in the position of the first row and the second column is increased by 1.

- After all input vectors in the training data are processed, the label of the counter which is the maximum for a codebook is assigned as the label of the codebook vector. According to the final result matrix given in Equation 19, the class labels of the codebook vectors are 1, 4, 3, and 4, respectively.

$$C = \begin{bmatrix} c_{11} & c_{12} & c_{13} & \mathbf{L} & c_{1D} & 8 & 2 & 0 & 0 \\ c_{21} & c_{22} & c_{23} & \mathbf{L} & c_{2D} & 0 & 1 & 0 & 9 \\ c_{31} & c_{32} & c_{33} & \mathbf{L} & c_{3D} & 1 & 0 & 6 & 3 \\ c_{41} & c_{42} & c_{43} & \mathbf{L} & c_{4D} & 0 & 2 & 1 & 7 \end{bmatrix} \quad (16)$$

3.2.8 Consensual Classification with Preprocessing

The basic idea in the consensus approach is to collect the individual results and obtain a group decision which is expected to have a higher classification performance and more reliable results [Xu et al., 1992], [Kittler et al., 1998], [Lee et al., 2007].

In the field of drug/nondrug classification studies, different classification algorithms can be used. Each of these algorithms can yield different classification accuracy. That is, it is difficult to decide which result is better or which algorithm is suitable for the studies. Because of this reason, the methodology of integrating the results of a number of different classification algorithms is needed, so that a better result from the different results can be obtained [Xu et al., 1992].

Consensual classification includes the results of several different classifiers to improve the classification accuracy. Even though the individual result of classification may be poor, consensus between the individual results could potentially offer complementary information.

As mentioned before, the basic idea of consensual classification is to combine the individual results, i.e. opinions, and obtain a group decision which is better than an individual decision. These individual results can be obtained from the different neural networks with different form of training data, or different topology of nets. In the related literature, this set of neural networks is called neural network ensemble [Hansen et al., 1990].

To decide which neural network will be included to the ensemble is important. The main point is that the members of neural network ensemble should produce independent errors [Lee et al., 2007]. If the statistically different errors are generated by the ensemble members, the improvement of classification can be obtained by combining their results. That is, consensual approach is very useful if the members of neural network ensembles are different or used with different data set, topology etc. In 1990, Sharkey proposed several methods which can be used such as varying the initial random weights, varying the network architecture, varying the network type and varying the training data to achieve this [Sharkey, 1996]. Varying the data is most frequently used for the creation of ensemble members. This process basically involves altering the training data. Sampling data, disjoint training sets, boosting and adaptive resampling, different data sources, and preprocessing are the ways that are used for altering the training data [Sharkey, 1996].

Sampling data methods produce different subsamples to use in neural networks. Disjoint training sets can also be produced as subsamples. The difference between them is for the disjoint case, the subsamples are nonoverlaped, i.e. mutually exclusive. The basis of boosting and adaptive resampling algorithms is that the training sets are adaptively resampled. This is done such that the weights in resampling are increased for the cases which are often misclassified. This method requires a large amount of data. Different data sources is the another method of varying the data. For this method, data from different input sources is used with neural networks. The preprocessing method which is the last one is altering the input data by using different methods such as extracting different feature vectors from the input data, injection noise or nonlinear transformations [Sharkey, 1996].

In this study for consensual classification, instead of different classifiers, a

single classifier was used to create members of an ensemble. For this purpose, a preprocessing method was used on the input data with a single classifier. As mentioned before the major criterion of combining classifiers is producing independent errors with each classifier [Kittler et al., 1998]. In this thesis, to generate independent results with the same classifier, the input data was preprocessed in two stages at the beginning of the each classification process. At the first stage, a randomly selected uniformly distributed matrix between -1 and 1 is used to transform descriptor vectors of a compound. Secondly, 1D median filtering with window size 3 is applied to remove noise [Vertan et al., 1996].

The aim of both procedures is to make each single classifier produce approximately independent random errors even though the same classifier is used. The block diagram of a single classifier with preprocessing is shown in Figure 3.12. The general structure for implementing consensual theory with this approach is shown in Figure 3.13.

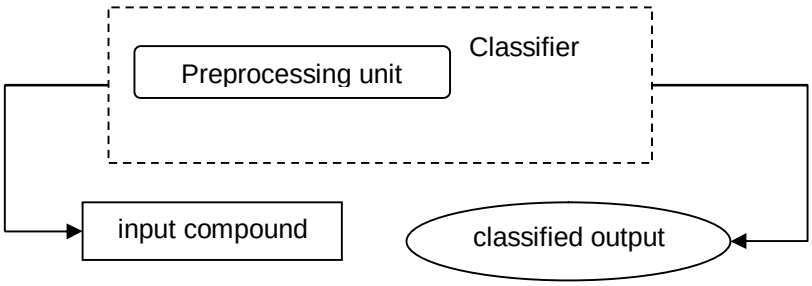


Figure 3.12. Structure of a single classifier.

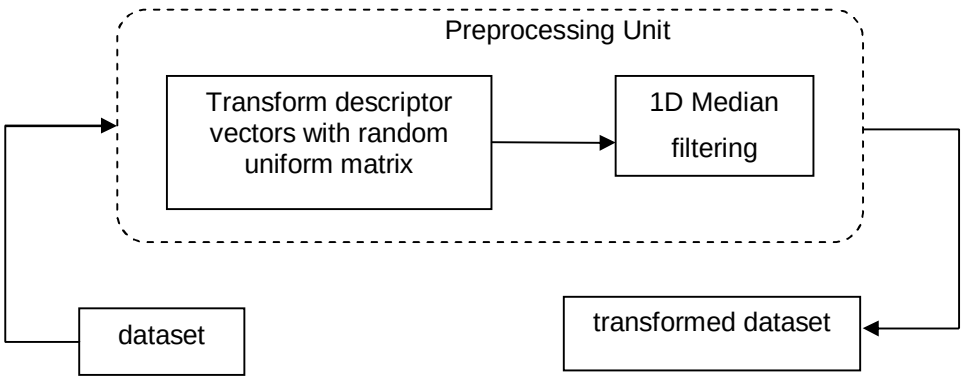


Figure 3.13. The preprocessing unit.

3.2.9 Combining Classification Results

It was assumed that a classifier *cl* only outputs a unique label *j* [Xu et al, 1992]. Then, a majority vote rule is applied to combine the output results.

In the proposed combination scheme, there are *K* different networks to be combined. Each of them produces output results such as $cl_k(x_i) = j$ where *k=1,…,K*, x_i is the i^{th} pattern and *j* is the class label. *K* different networks are applied to each input pattern to obtain classification labels; each label corresponds to one of the *C* data classes. The majority vote rule combines the results of the *K* classifiers. If there are *P* patterns which are compounds in this study in the data set, the size of multiple-classification result matrix is expressed as *P X K*. Both train and test datasets have counter matrices C_{train} and C_{test}, respectively, with the size of *P* X *C*, where *P* is the number of patterns, and *C* is the number of classes for obtaining the group decision of the *K* classifiers. In addition, each classifier result is assigned a corresponding weight l_k, where *k=1,…,K*. For each vector in the dataset, the column of counter matrices C_{train} and C_{test}, corresponding to the result of each single classifier is increased by the corresponding weight l_k. According to the majority vote rule, the class label of each vector is decided by the maximum value among the class labels.

Thus, to generate consensual result, classification results are combined with the weight vector $l = [l_1, l_2, ..., l_k]$. Weight values are based on the individual performance of the classifiers [Benediktsson et al., 1997]. The block diagram of combining *K* classifiers with the corresponding l_k is shown in Figure 3.14.

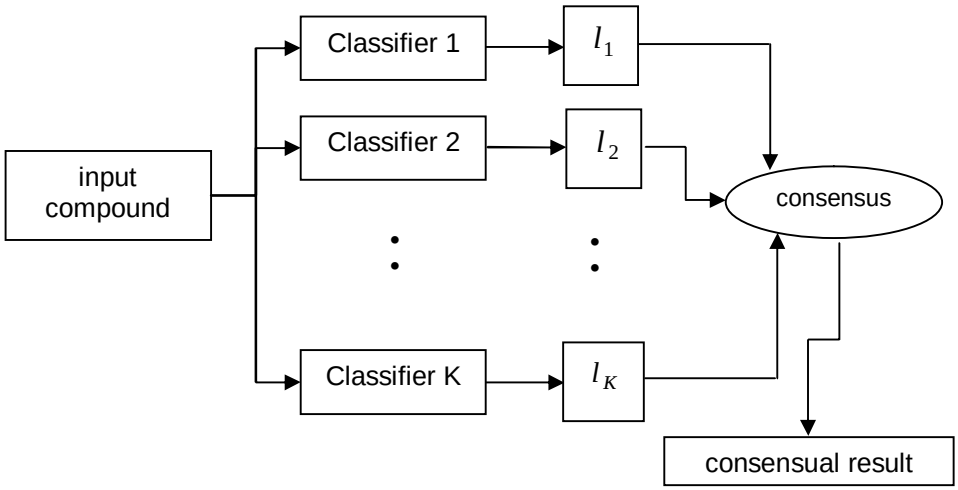


Figure 3.14. Scheme of consensual results with the same classification algorithm

In the study by Lee at el., two methods were mentioned for selecting l_k values, one is equal weights, and the other method is based on the delta rule, giving the weights as a function of the performance of each single classification result [Lee et al., 2007].

For obtaining weight values by the delta rule, *P X K* result matrix of multiple classifiers and the known desired outputs of the training data are used. Assume that *R* is the trained results of the single classifiers and *T* is the target output vector. In order to find the optimal weights by the delta rule, *R* and l are defined as follows:

$$R = [R_1 R_2 ... R_k] \qquad l = \begin{bmatrix} l_1 \\ l_2 \\ \mathbb{M} \\ l_k \end{bmatrix}$$

By solving the following equation, the weights can be obtained by the delta rule:

$$Rl \cong T \tag{20}$$

l is obtained by least square estimation minimizing the square error as follows

[Benediktsson et al., 1997].

$$l_{opt} = \min_{l} \left\| R\,l - T \right\|^2 \tag{21}$$

Solution for optimal l given in Equation 21 is sought by using the pseudoinverse of R under the assumption that R has a full column rank. The formula for l by pseudoinverse of R is given in Equation 22.

$$l_{opt} = \left(R^T R \right)^{-1} R^T T \tag{22}$$

where R^T is the transpose of R, and $\left(R^T R \right)^{-1} R^T T$ is the pseudoinverse of R.

In this study, in addition to these two methods, the genetic algorithm (GA) was used to obtain the optimal weight values. With the use of a starting population, which consists of random weight values and $P\ X\ K$ trained result matrix, adaptation of the l array for each network was obtained by GA using a modification the Matlab Genetic Algorithm Toolbox [Chipperfield et al., 1994].

4. RESULTS AND DISCUSSION

The main aim of this thesis was to distinguish a molecular group with a proven pharmacological activity from another molecular group without any activity. For this purpose, all experiments were done according to the flow chart shown in Figure 4.1.

As mentioned before, there were two data sets, Murcia-Soler and Cherkasov, used for the experiments. Classifications were performed around 20-30 times by using the same classification algorithm with the multiple different training data which is generated by the preprocessing unit to produce independent errors. This preprocessing unit, which includes transformation of the data set by random uniform matrix and then 1D median filtering to remove the possible noise, was performed at the beginning of each computation. Hence even though the same classification algorithm was used, this was equivalent to 30 different classifiers because of the preprocessing process.

All the results were measured according to the evaluation parameters given in Section 4.1. In the following sections, after explaining the model evaluation parameters, the results will be given according to the data sets used in this thesis.

There were a few parameters for the methods used in the experiments. In the application of the GRNN, 0.2 was chosen as the smoothing factor, sigma. The SOGR has the learning rate and neighborhood size. While the learning rate was decreased, neighborhood size was increased. Furthermore, when the number of the iteration increased, neighborhood size was decreased. A reason to choose dynamic values for the parameters was to avoid the local minima. Starting point to learning rate was chosen as 0.9 and the max value of neighborhood size was chosen as 5 in all experiments with the SOGR. For the experiments with the SOM network, the hexagonal lattice was used.

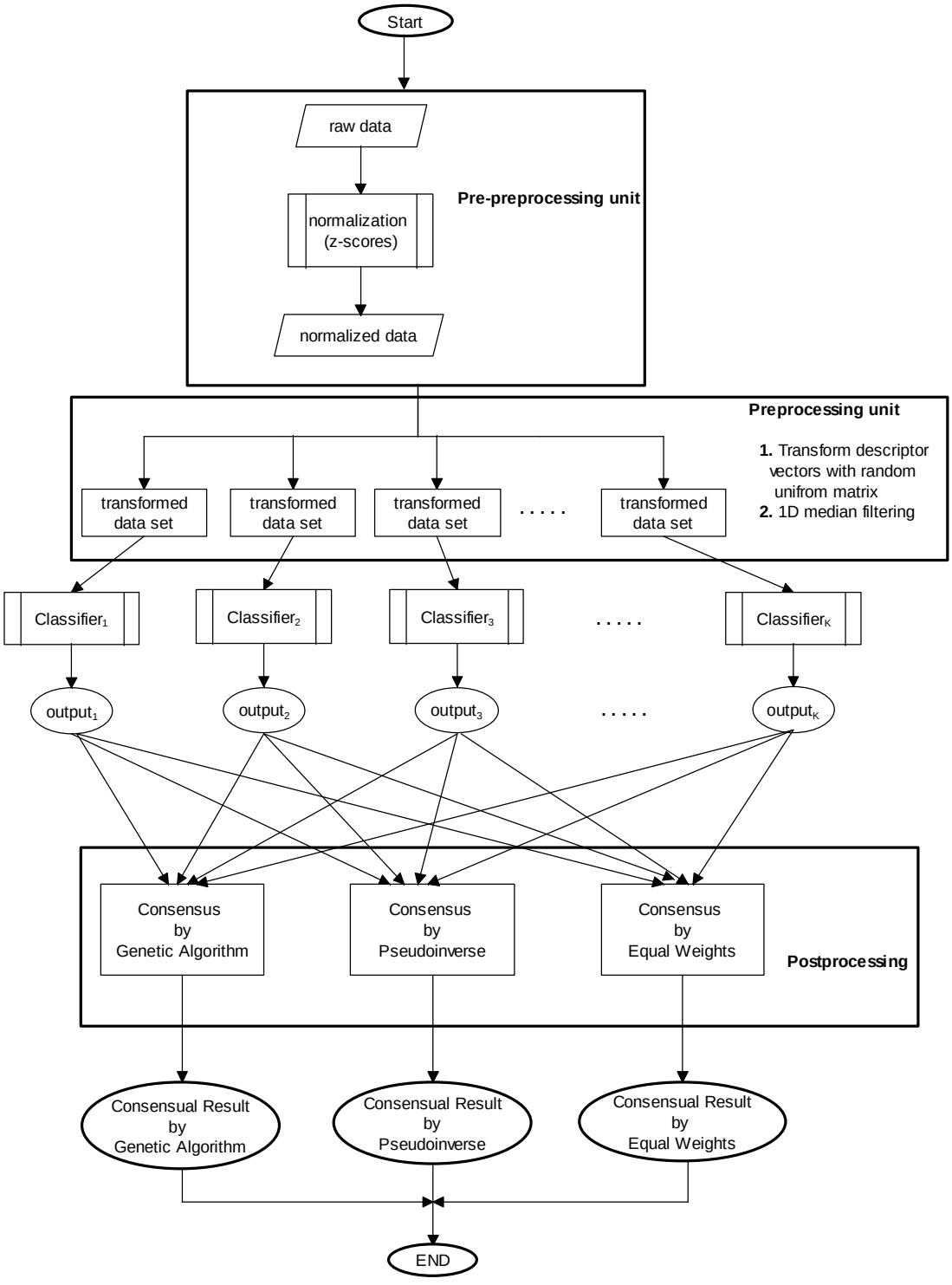


Figure 4.1. General flow diagram of the experiments

4.1. Model Evaluation

The performance quality of the models was examined by 3-fold cross-validation analysis. In the cross-validation test, the whole data set was normalized and then it was split into three blocks. Each block was singled out in turn for testing, and the remaining two blocks of the data set were used for the training model.

The results were quantified with the following statistical parameters: sensitivity (Q_d), specificity (Q_n) and overall accuracy (Q_a). These parameters were used to measure the accuracy of positive prediction, i.e. the percentage of drugs predicted correctly, negative prediction, i.e. the percentage of nondrugs predicted correctly, and the overall accuracy of the model, respectively. These measurements were calculated by using the following values:

TP (true positive): The number of correctly classified drug molecules,

FP (false positive): The number of nondrug molecules incorrectly classified as drug,

TN (true negative): The number of correctly classified nondrug molecules,

FN (false negative): The number of drug molecules incorrectly classified as nondrug.

The metrics were calculated by using these values with the following equations:

$$Q_d = TP/(TP+FN)$$
$$Q_n = TN/(TN+FP)$$
$$Q_a = (TP+TN)/(TP+TN+FP+FN)$$

In general, the overall accuracy Q_a is always used to measure the prediction power of a model [Li et al., 2007]. The Matthew's correlation coefficient (Q_{mcc}) and probability excess (Q_{pe}) values were also calculated for comparison.

The Q_{pe} that varies between 0 and 1 is calculated by (sensitivity + specificity – 1). While accuracy is affected by the relative frequencies of the two classes, the Q_{pe}

value is independent of the relative class frequency in the dataset [Yang et al., 2005].

The Q_{mcc}, given in the following equation, can vary from -1 to +1, with higher values indicating better predictions, represents the predictive power of the method.

$$Q_{mcc} = \frac{TP.TN - FN.FP}{\sqrt{(TP+FN)(TP+FP)(TN+FN)(TN+FP)}}$$

4.2. Results with the Murcia-Soler Data Set

641 input patterns corresponding to 2D MOE descriptors for the studied general therapeutics and nondrug compounds were used to train GRNN, adGRNN, SOM and SOGR methods. As mentioned previously, the network outputs were assigned to 1.0 for drugs and 0.0 for all nondrugs in the data set.

4.2.1 Model (a): Model with Topological and Structural MOE Descriptors

Total of 61 features were selected among the topological and structural properties for this model. 12 of these descriptors included simple information on the molecule: total number of atoms of certain elements (carbon, nitrogen, oxygen, sulfur, fluorine, chlorine, and bromine), total number of simple, double and triple bonds, and number of vertex degrees. The remaining descriptors contained different topological and structural properties. The 61 descriptors shown in Table 4.1 were chosen based on the study of Murcia-Soler et al. [Murcia-Soler et al., 2003].

The GRNN, adGRNN and SOGR algorithms were applied for Model (a). Each algorithm was run independently 30 times with the preprocessing unit to have group decisions. These consensual results were obtained by 3 different ways; GA, pseudoinverse of result matrix and equal weights. The combined classification results of sensitivity, specificity, probability excess and accuracy generated by GRNN, adGRNN and SOGR for the testing data were shown in Table 4.2.

Table 4.1. The list of QSAR Descriptors for Model (a)

Categories	Descriptors	Number of Descriptors
Atom and Bond Counts	a_nC, a_nBr, a_nN, a_nO, a_nS, a_nF, a_nCl, b_single, b_double, b_triple, VAdjMa, VAdjEq	12
Kier&Hall Connectivity and Kappa Shape Indices	chi0, chi0_C, chi1, chi1_C, chi0v, chi0v_C, chi1v, chi1v_C, Kier1, Kier2, Kier3, KierA1, KierA2, KierA3, KierFlex, zagreb	16
Adjacency and Distance Matrix	balabanJ, BCUT_PEOE_0, BCUT_PEOE_1, BCUT_PEOE_2, BCUT_PEOE_3, BCUT_SLOGP_0, BCUT_SLOGP_1, BCUT_SLOGP_2, BCUT_SLOGP_3, BCUT_SMR_0, BCUT_SMR_1, BCUT_SMR_2, BCUT_SMR_3, diameter, petitjean, GCUT_PEOE_0, GCUT_PEOE_1, GCUT_PEOE_2, GCUT_PEOE_3, GCUT_SLOGP_0, GCUT_SLOGP_1, GCUT_SLOGP_2, GCUT_SLOGP_3, GCUT_SMR_0, GCUT_SMR_1, GCUT_SMR_2, GCUT_SMR_3, petitjeanSC, radius, VDİstEq, VDistMa, weinerPath, weinerPol	33

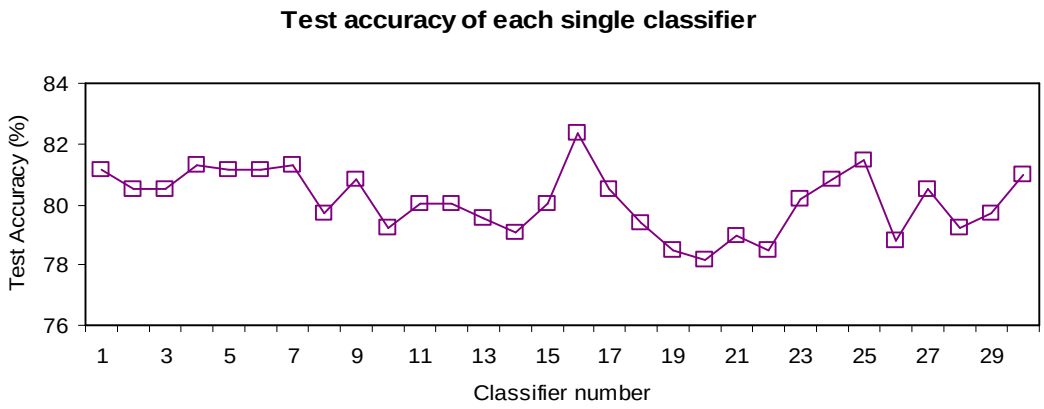
Table 4.2. The Consensual Testing Accuracy of GRNN, adGRNN and SOGR.

Method	Weight Method	Sensitivity	Specificity	Probability Excess	Accuracy
GRNN	**Genetic algorithm**	**87.74**	**75.56**	**63.30**	**83.46**
	Pseudoinverse	83.90	65.33	49.23	77.38
	Equal Weight	90.63	65.78	56.41	81.91
adGRNN	**Genetic algorithm**	**86.54**	**82.67**	**69.21**	**85.18**
	Pseudoinverse	87.98	68.00	55.98	80.97
	Equal Weight	92.30	63.56	55.86	82.21
SOGR	**Genetic algorithm**	**82.34**	**85.78**	**68.12**	**83.46**
	Pseudoinverse	89.52	68.00	57.52	81.90
	Equal Weight	94.46	64.89	59.35	84.08

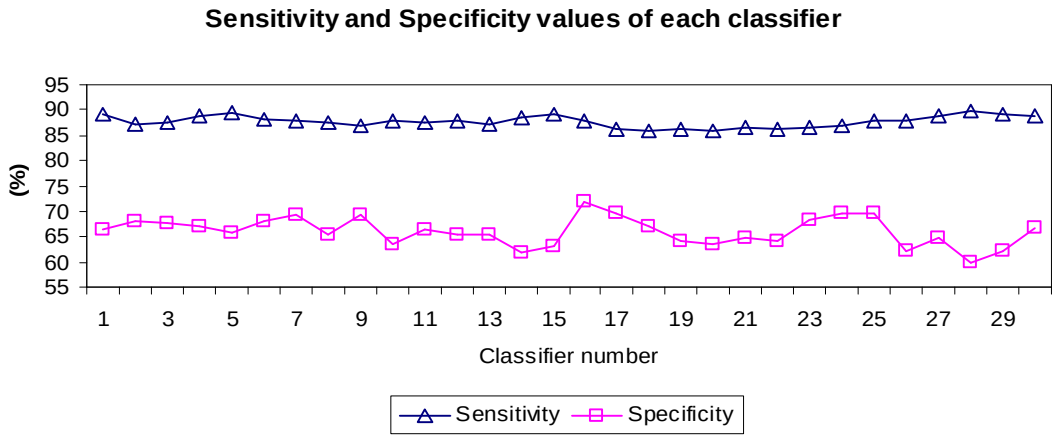
The consensual sensitivity, specificity, probability excess and accuracy of methods classification results according to different weight methods. All values are given in %.

The consensual results illustrated in Table 4.2 were averaged statistics of 3-fold runs for testing data (%33-35 of the inputs used). All the related parameters were calculated with training data (%68-65 of inputs used) for SOGR methods. The GRNN and adGRNN parameters were calculated by the validation set which was 20% of the training data set.

As mentioned above, each method was run 30 times before postprocessing, i.e. consensus. The individual testing accuracy, sensitivity and specificity results of GRNN, adGRNN and SOGR are shown in Figure 4.2, 4.3 and 4.4, respectively.

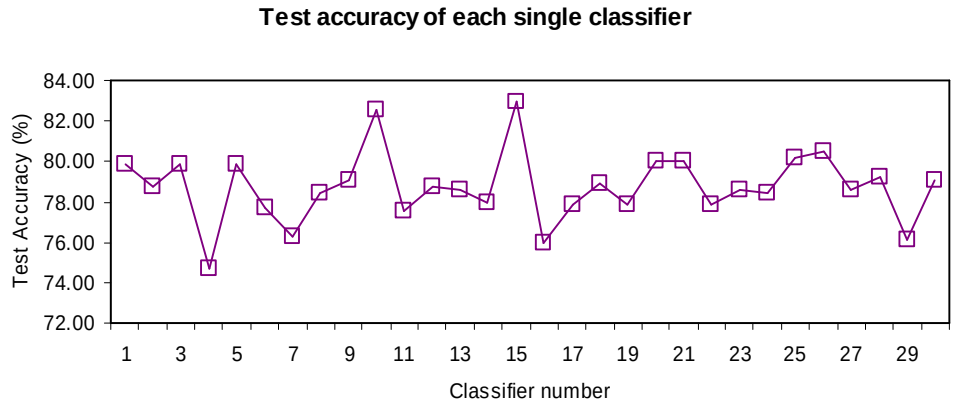


(a)

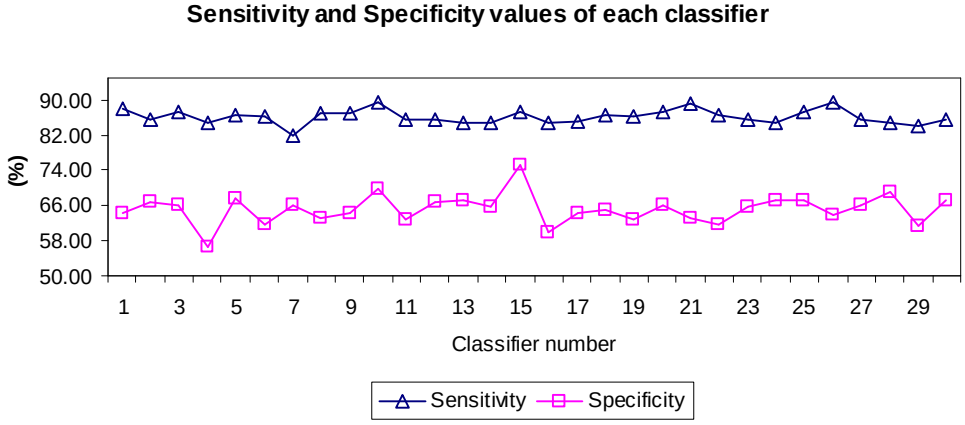


(b)

Figure 4.2. (a) Individual testing accuracy, (b) sensitivity and specificity results of each single classifier GRNN.

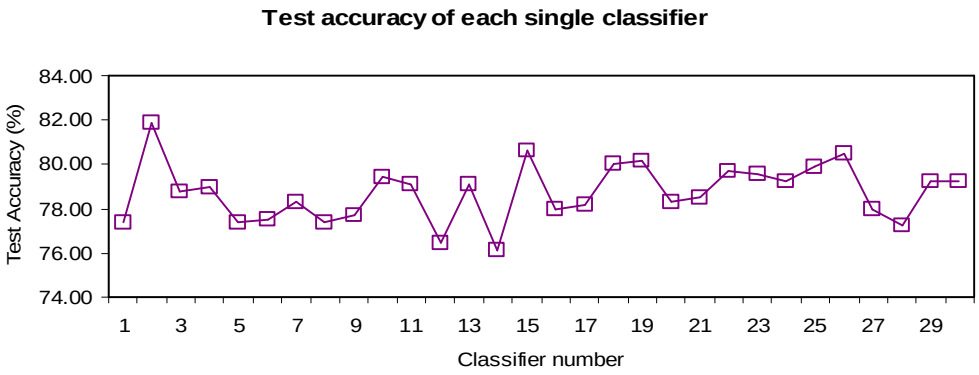


(a)

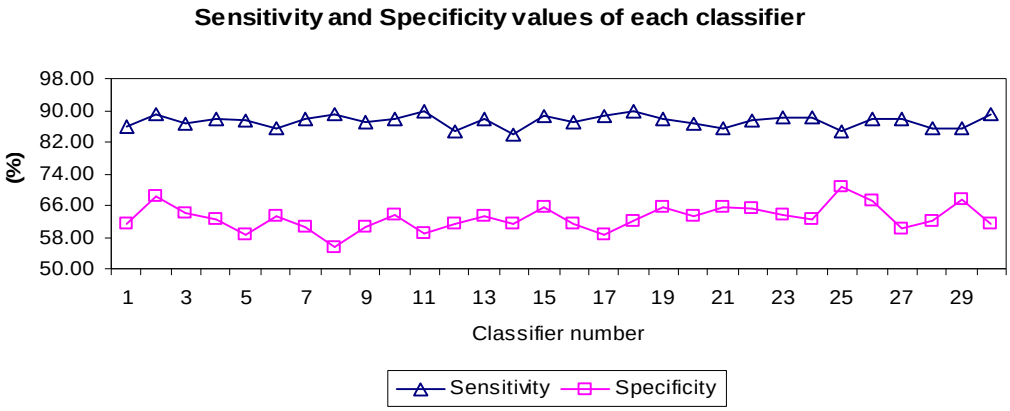


(b)

Figure 4.3. (a) Individual testing accuracy, (b) sensitivity and specificity results of each single classifier adGRNN.



(a)



(b)

Figure 4.4. (a) Individual testing accuracy, (b) sensitivity and specificity results of each single classifier SOGR.

According to complete individual experimental results shown in Figure 4.2, 4.3 and 4.4, the best accuracy of GRNN was 82.37%, adGRNN was 82.99% and SOGR was 81.90% among those of single classifiers. By combining with consensus rule, the test accuracies were improved to 83.46%, 85.18% and 83.46 for GRNN, adGRNN and SOGR, respectively.

Another observation derived from Table 4.2 was the best consensus results produced by GA for all methods. The consensus approach with GA provided better performance, especially in the success of the inactive compounds prediction. For instance, while the individual nondrug classification results of the adGRNN varied between 56.44%-75.11% shown in Figure 4.3(b), Table 4.2. illustrates that combining results by GA provided 7%-26% better prediction success. The similar improvement was also observed for the SOGR consensus result with GA shown in Table 4.2.

In the original study of Murcia-Soler *et al.*, they used a feedforward neural network method to classify compounds into potentially drug and nondrug compounds with 76% and 70% of success rate, respectively [Murcia-Soler *et al.*, 2003]. In this thesis the model (a), with the consensual classification algorithm proposed, prediction success of 86.54% and 82.67% were obtained for drugs and nondrugs, respectively. It is obvious that there are significant improvements

achievable with the proposed consensual method to separate active compounds from inactive compounds.

4.2.2 Model (b): Model with All 2D MOE Descriptors

For this model, all 2D MOE descriptors included the physical properties, subdivided surface areas, atom and bond counts, kier and hall connectivity and kappa shape indices, adjacency and distance matrix, pharmacophore features and partial charges, were calculated for all the input patterns.

The total of 162 descriptors with input patterns matrix was normalized and splitted into three training-testing pairs. SOM and SOGR methods applied on each pairs for 30 times. Individual testing accuracy results of both methods are given in the following figure (Figure 4.5).

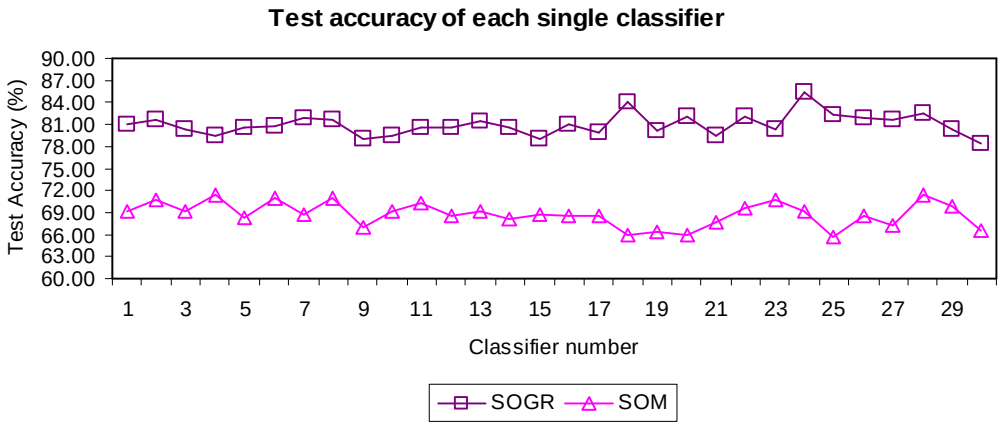
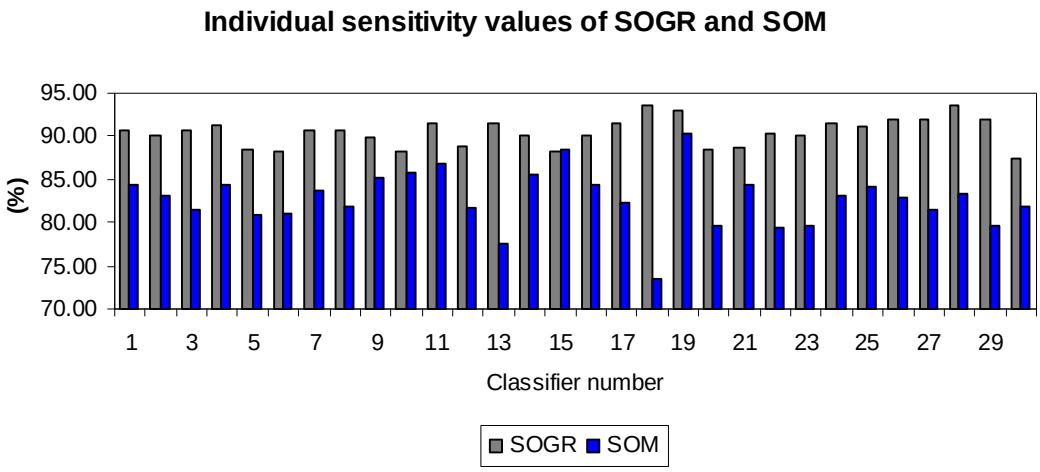
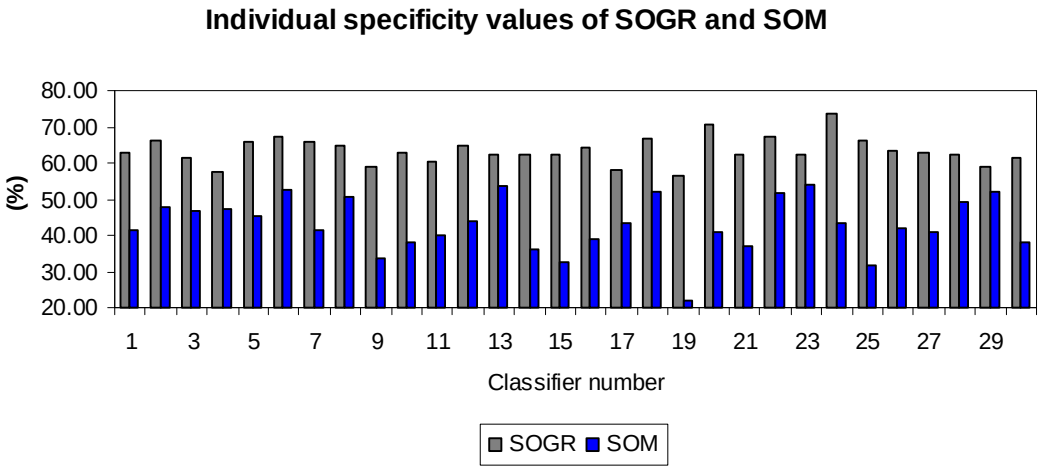


Figure 4.5. Individual testing accuracy of SOGR and SOM classifiers

At first glance, it was observed that individual SOGR results were significantly better than supervised SOM results. On the other hand both methods provided high success in classification of drug compounds, but not with nondrug compounds. In Figure 4.6 individual results of sensitivity and specificity values are given for both methods.



(a)



(b)

Figure 4.6. (a)Individual sensitivity results and (b) Individual specificity results of SOGR and SOM classifiers

In addition to the individual results, consensus results for both classifiers were also calculated. Table 4.3 illustrates the consensus classification results for the methods. According to the complete experimental results given in Figure 4.5, the best test accuracy of SOM was 71.45% among those of single classifiers. By combining with consensus rule, the test accuracy improved to 74.10% as shown in Table 4.3. This result was obtained by consensus approach with GA. In Table 4.3, it is also observed that the prediction of inactive compounds (specificity) is quite low when compared to the success of active compounds prediction (sensitivity). Low

probability excess values also show these unbalanced results between the successes of two classes.

Table 4.3. The Consensual Testing Accuracy of SOM and SOGR**.**

Method	Weight Method	Sens.	Spec.	Mcc.	ProbEx	Acc.
SOM	**GA**	**86.05**	**52.00**	**40.81**	**38.05**	**74.10**
	PseInv.	83.40	54.67	40.05	38.06	73.32
	Equal	86.28	44.45	34.54	30.72	71.63
SOGR	**GA**	**90.63**	**80.44**	**71.63**	**71.08**	**87.05**
	PseInv.	94.48	68.44	67.17	62.92	85.34
	Equal	95.92	64.44	66.34	60.36	84.87

The consensual classification performances of the SOM and SOGR methods according to the different weight methods. All values are given in %. (GA = Genetic Algorithm, PseInv = Pseudoinverse, Sens. = Sensibility, Spec. = Specificity, Mcc = Mathew's Correlation Coefficient, ProbEx. = Probability Excess, Acc = Accuracy)

While individual results of SOGR varied between 78.32%-85.34% as shown in Figure 4.5, Table 4.3 illustrates that combining results by GA provided 2%-9% better prediction success. This improvement can also be observed by checking probability excess and Mathews' correlation values which depend on sensitivity and specificity results shown in Figure 4.6.

It was clear from Table 4.3 that the SOGR results were significantly better for classification of active and inactive compounds. The most obvious improvement was observed in prediction of inactive compounds, i.e. specificity value. While the SOM algorithm classified nondrug compounds with 52% accuracy as given in Table 4.3, the consensus result of specificity value provided by SOGR was 80.44%. Consequently, this improvement also resulted in higher probability excess and Mathews' correlation coefficient values.

4.2.3 Model (c): Model with Principle Components

To reduce dimensionality of the feature space, principle component analysis (PCA) was performed using the MOE [MOE; 2006]. Principle components (PCs) were calculated in two ways. In the first way all 2D MOE descriptors were projected

onto principle component space, and first 60 principle components (PCs) were selected for analysis. In the second way, PCs were calculated within the groups: physical properties, subdivided surface areas, atom and bond counts, kier and hall connectivity and kappa shape indices, adjacency and distance matrix, pharmacophore features and partial charges. According to the second way there were 103 PCs from those groups given in Table 4.4.

Table 4.4. Number of PCs Within the Descriptor Groups

Categories	# of PCs
Physical Properties	8
Subdivided Surface Areas	15
Atom and Bond Counts	22
Kier&Hall Connectivity and Kappa Shape Indices	10
Adjacency and Distance Matrix	20
Pharmacophore Feature	10
Partial Charges	18

All methods were applied to both data sets. Consensual results of the 20 individual experiments for each method are given in the following table (Table 4.5). According to the results given in Table 4.5, the SOGR results of the 103 PCs produced slightly better classification performance as compared to the SOGR results of the 60 PCs. On the other hand, adGRNN and GRNN provided significantly better results than the results given in Table 4.2. The obvious improvement was observed especially in nondrug classification results. This improvement is quite important to this study. The false positive compounds which were incorrect classification of nondrug compounds can be considered as drug-like compounds.

According to individual results, PCs within the groups provided better testing accuracy shown in Figure 4.7, 4.8 and 4.9 for GRNN, adGRNN and SOGR, respectively.

Table 4.5. The Consensual Testing Results of GRNN, adGRNN and SOGR with PCs

PCs	Method	Weight Method	Sens.	Spec.	Mcc.	ProbEx	Acc.
60 PCs (full)	GRNN	**GA**	**95.67**	**87.11**	**83.81**	**82.78**	**92.67**
		PseInv.	90.15	80.00	70.44	70.15	86.58
		Equal	96.39	84.45	82.71	80.84	92.20
	adGRNN	**GA**	**94.24**	**85.33**	**80.55**	**79.57**	**91.11**
		PseInv.	81.97	73.78	55.40	55.75	79.09
		Equal	97.83	78.22	80.13	76.06	90.95
	SOGR	**GA**	**91.57**	**73.33**	**67.39**	**64.91**	**85.18**
		PseInv.	93.74	66.67	64.63	60.41	84.24
		Equal	96.39	59.11	62.91	55.50	83.31
103 PCs (within groups)	GRNN	**GA**	**93.51**	**88.89**	**82.53**	**82.40**	**91.89**
		PseInv.	89.66	80.89	70.61	70.55	86.58
		Equal	94.72	86.67	82.47	81.38	91.89
	adGRNN	**GA**	**96.87**	**81.78**	**81.39**	**78.65**	**91.58**
		PseInv.	89.43	82.67	71.84	72.10	87.05
		Equal	96.63	79.11	78.98	75.74	90.48
	SOGR	**GA**	**93.27**	**76.89**	**72.18**	**70.16**	**87.52**
		PseInv.	91.09	71.55	65.01	62.65	84.24
		Equal	95.19	66.67	66.89	61.86	85.18

The consensual classification performances of the GRNN, adGRNN and the SOGR methods according to the different weight methods with principle components. All values are given in %. (PC = Principle Component, GA = Genetic Algorithm, PseInv = Pseudoinverse, Sens. = Sensibility, Spec. = Specificity, Mcc = Mathew's Correlation Coefficient, ProbEx. = Probability Excess, Acc = Accuracy)

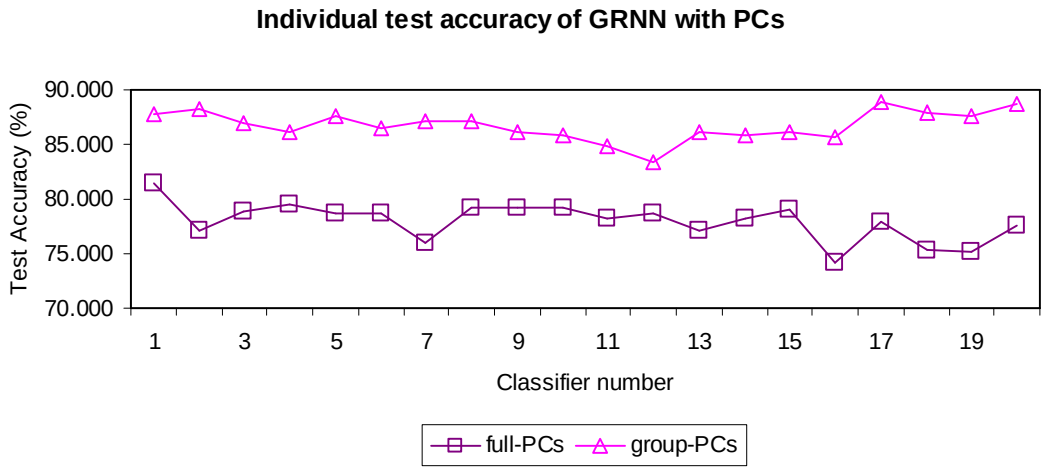


Figure 4.7. 20 Individual testing accuracies of GRNN with full PCs and group PCs.

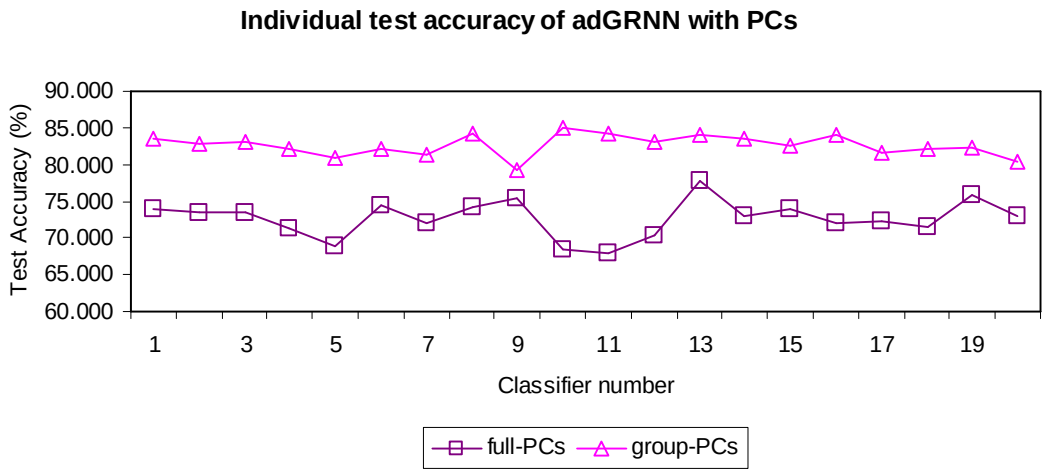


Figure 4.8. 20 Individual testing accuracies of adGRNN with full and group PCs.

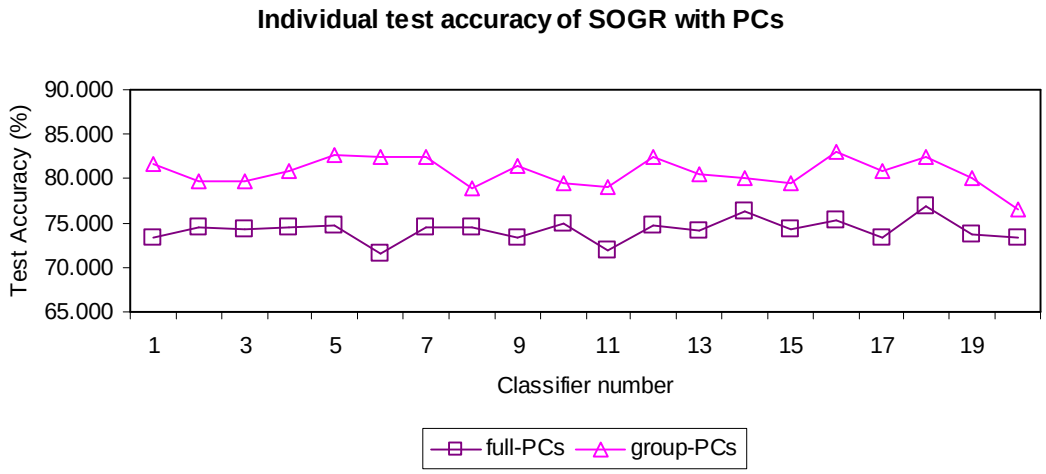


Figure 4.9. 20 Individual testing accuracies of SOGR with full and group PCs

In the following table, minimum and maximum ranges were given for the individual results of this model (c) (Table 4.6). Comparison of tables 4.5 and 4.6 demonstrates that the consensual approach with GA produced better results.

Table 4.6. The Min-Max Ranges of 20 individual GRNN, adGRNN and SOGR Results.

PCs	Method	Ranges	Sens.	Spec.	ProbEx.	Mcc.	Acc.
60 PCs (full)	GRNN	Min	72.358	66.667	47.511	47.082	74.257
		Max	83.903	81.778	61.176	60.410	81.431
	adGRNN	Min	69.230	61.778	33.181	32.390	68.022
		Max	84.133	74.667	50.355	50.859	77.841
	SOGR	Min	79.326	53.333	35.433	36.288	71.600
		Max	87.749	68.000	48.909	49.176	76.909
103 PCs (within groups)	GRNN	Min	83.662	78.667	66.773	65.465	83.463
		Max	92.559	86.667	75.750	75.761	88.928
	adGRNN	Min	80.294	68.000	53.341	54.064	79.256
		Max	88.950	80.889	68.142	67.533	85.020
	SOGR	Min	83.881	61.333	46.548	47.874	76.442
		Max	91.106	71.556	59.767	62.348	83.153

The min-max ranges of the individual GRNN, adGRNN and the SOGR methods according to the different weight methods with principle components. All values are given in %. (PC = Principle Component, GA = Genetic Algorithm, PseInv = Pseudoinverse, Sens. = Sensibility, Spec. = Specificity, Mcc = Mathew's Correlation Coefficient, ProbEx. = Probability Excess, Acc = Accuracy)

4.3. Results of Cherkasov Data Set

4.3.1 Model with 2 classes

As mentioned previously, the Cherkasov data set included 523 antimicrobial compounds, 959 general drugs and 1202 drug-like chemicals which are not in any therapeutic category. All 2D MOE descriptors were calculated with the MOE software for all input data. Over these all descriptors 80 PCs and 127 group PCs were calculated. Data set with all 2D MOE has 164 descriptors that were normalized with the z-score mentioned before. For all data sets, 3 nonoverlaped training-testing pairs were used. The applications on these data sets were similar to applications on the Murcia-Soler data sets. Data set with all descriptors were run with the GRNN, adGRNN, SOGR and SOM, 20 individual times each. According to the results, the best consensual classification was obtained by adGRNN. As in Murcia-Soler data set applications, combining the individual results with GA produced the best results. Table 4.7 summarizes the consensual classification performance of the four methods.

Table 4.7. The Consensual Results of 20 individual GRNN, adGRNN and SOM and SOGR Methods for Cherkasov Data Set.

Method	Weight Method	Sens.	Spec.	Mcc.	ProbEx	Acc.
GRNN	**GA**	**80.97**	**85.78**	**66.52**	**66.75**	**83.12**
	PseInv.	74.63	84.70	60.60	59.34	79.13
	Equal	75.98	87.36	64.21	63.34	81.07
adGRNN	**GA**	**90.22**	**80.03**	**71.00**	**70.25**	**85.65**
	PseInv.	72.47	85.12	59.50	57.59	78.13
	Equal	77.80	89.03	67.88	66.83	82.83
SOM	GA	77.67	58.31	38.06	35.98	69.00
	PseInv.	**70.17**	**69.97**	**40.49**	**40.15**	**70.09**
	Equal	66.80	65.97	33.27	32.78	66.43
SOGR	**GA**	**77.94**	**81.28**	**59.17**	**59.21**	**79.43**
	PseInv.	77.55	80.20	58.18	57.80	78.76
	Equal	79.55	78.54	58.67	58.10	79.10

The consensual classification performances of the GRNN, adGRNN and the SOGR methods according to the different weight methods with principle components. All values are given in %. (PC = Principle Component, GA = Genetic Algorithm, PseInv = Pseudoinverse, Sens. = Sensibility, Spec. = Specificity, Mcc = Mathew's Correlation Coefficient, ProbEx. = Probability Excess, Acc = Accuracy)

In addition to the consensual results, individual testing accuracies are also shown in Figure 4.10. There are obvious differences between the individual results and the consensual results. Group decisions of 30 individual runs for each method produced better testing accuracy performance than the maximum results of the individual runs. These improvements were obtained with all the methods because of the independent errors.

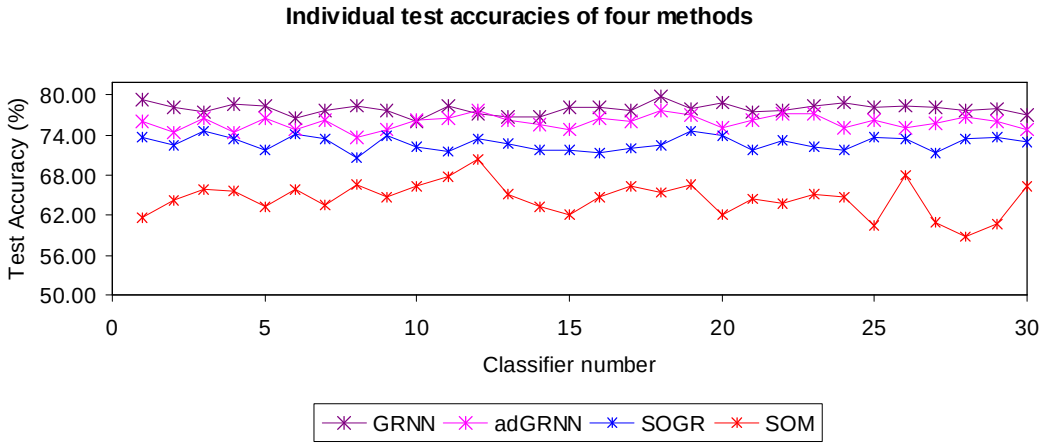


Figure 4.10. 30 Individual testing accuracies of GRNN, adGRNN, SOM and SOGR

Another experiment with Cherkasov data set was done calculating PCs. PCs were calculated in two ways as in Murcia-Soler data set, 80 PCs calculated over all descriptors and 127 PCs calculated within seven groups of MOE descriptors. Although consensual approach provided better results than the individual results for data sets with PCs, there were no improvements, besides calculation time due to the smaller data sets. On the other hand, due to the calculation time of PCs, PCA was not preferred method with the Cherkasov data set. The following result table demonstrates the SOGR results with PCs and original descriptors as an example (Table 4.8).

Table 4.8. The Consensual Results of SOGR with Variations of Cherkasov Data Set.

Data Set	Sens.	Spec.	Mcc.	ProbEx	Acc.
All Descriptors	77.94	81.28	59.17	59.21	79.43
80 PCs (full)	79.15	72.21	51.99	51.36	76.04
127 PCs (within group)	81.78	75.13	57.35	56.91	78.80

The consensual classification performances of the SOGR method according to the different weight methods with all descriptors, full PCs., and group PCs. All values are given in %. (PCs = Principle Components, Sens. = Sensibility, Spec. = Specificity, Mcc = Mathew's Correlation Coefficient, ProbEx. = Probability Excess, Acc = Accuracy)

4.3.2 Model with 3 Classes: Antimicrobials, Drugs and Druglikes

For distinguishing antimicrobials, drugs and drug-likes of Cherkasov data set from each other, 523 compounds were assigned activity value 1.0 corresponding to their classification as antimicrobials, 959 compounds were associated with 2.0 outputs reflecting the general drugs and 1202 druglike chemicals were assigned 3.0 output value. The data set was split into three training-testing pairs. Each pair was run 20 times with preprocessing. Figure 4.11 illustrates the individual results for each fold. As shown in Figure 4.11, classification performance of antimicrobials was slightly better than classification performance of the others, general drugs and drug-likes.

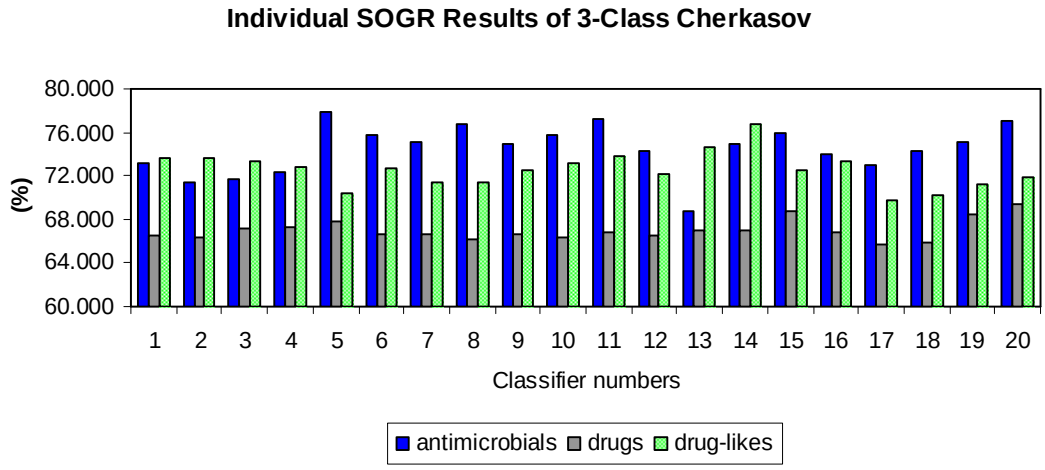


Figure 4.11. 20 Individual SOGR testing classification results of antimicrobials versus drugs versus druglikes

The misclassified druglike chemicals, i.e. classified as antimicrobials or general drugs, can be considered as a starting point for drug design studies. In this experiment, druglike chemicals were divided into 3 equal parts for each testing fold. For the first fold, 66 druglike chemicals were classified as drug or antimicrobials, 60 of second fold's druglike chemicals were classified incorrectly and in the last fold, 90 of the druglike chemicals were misclassified. These total 216 misclassified druglikes out of 1202 druglikes in the data set were obtained with the consensual

approach with GA. Table 4.9 summarizes the consensual results provided with the SOGR for each fold.

Table 4.9. 3 –fold Consensual Results of SOGR with 3-class Cherkasov Data Set.

Fold	Weight Method	Antimicrobials	Drugs	Drug-likes	Overall
1	GA	78.29	80.25	83.50	81.32
	PseInv.	65.71	75.55	86.50	78.52
	Equal	79.43	75.86	86.00	81.10
2	GA	81.03	75.00	85.54	80.89
	PseInv.	67.82	74.69	84.29	77.65
	Equal	79.89	75.31	85.54	80.78
3	GA	90.23	76.88	77.56	79.78
	PseInv.	82.76	80.31	73.32	77.65
	Equal	89.08	81.25	74.06	79.55
Average Results	**GA**	**83.18**	**77.38**	**82.20**	**80.66**
	PseInv.	72.10	76.85	81.37	77.94
	Equal	82.80	77.47	81.87	80.48

GA = Genetic Algorithm, PseInv = Pseudoinverse. All results are given in %.

5. CONCLUSIONS

This thesis presented a novel consensual classification approach using a single classifier which effectively increased the discrimination between the drug-like and the nondrug-like compounds. In order to generate consensual result, i.e. group decision, GRNN, adGRNN, SOM and SOGR methods were used as the individual classification results.

Lee et al. applied consensual classification to remotely sense multispectral images by using weights generated by the pseudoinverse method and equal weights [Lee et al., 2007]. In this thesis, the GA was used to obtain optimal weights in addition to these two weight methods. All the experiments indicated that the best classification results were obtained with the proposed GA approach.

In addition to the consensus approach with GA, the SOGR method was used to separate drug compounds from nondrug compounds for the first time in literature. This method gives the equal chance for all neurons without considering the topological neigborhood concept.

The aim of this study was contribute automation tools to the drug designers at the preprocessing stage of the design process. Since it may be important to start with the correct drug-like chemicals, we considered the nondrugs which were classified as the drugs, i.e. false positives. Thus, it may be desirable to keep these incorrectly classified drug-like compounds. These misclassified drug-like compounds can lead the study of drug designs.

There were two different data sets, Murcia-Soler and Cherkasov, used in the experiments. While the Murcia-Soler data set consisted of drug and nondrug compounds, the Cherkasov data set involved drug (general drug and antimicrobials) and drug-like compounds. All the experiments were done with both data sets with 2D MOE descriptors in order to find the potential drugs. For the Murcia-Soler data set, if a compound is structurally more similar to examples of known drugs that to nondrugs, it can be assumed that it might be a potential drug. For the Cherkasov data set, the false positive predictions might be considered as a potential drug.

Murcia-Soler et al. used a feedforward neural network method to classify

compounds into potentially drug and nondrug compounds with 76% and 70% of success rate, respectively [Murcia-Soler et al, 2003]. In this thesis, with the consensual classification algorithm proposed, prediction SOGR success of 90.63% and 80.44% were obtained for drugs and nondrugs, respectively. These results were obtained with full 2D MOE descriptors. On the other hand, with smaller descriptor set which is more similar to the original data set of the study of Murcia-Soler, prediction adGRNN success of 86.54% and 82.67% were obtained for drugs and nondrugs, respectively. It is obvious that there are significant improvements achievable with the proposed consensual method to separate active compounds from inactive compounds to decide the potential drugs.

One of the other experiments done in this thesis were done with 3-class Cherkasov data set and 216 compounds were classified incorrectly and might be consider as potential drugs.

In conclusion, this thesis presented that the best results to discriminate between the drug-like and the nondrug-like compounds were obtained with the proposed consensus approach with genetic algorithm which is used for the first time for combining. On the other hand, the self organizing global ranking algorithm which is applied for the first time on chemical data set also produced high classification results. On the basis of results obtained from these experiments, new approach was provided for choosing potential drug candidates by using molecular data automatically. Hence, it can be stated that this narrowing considerably reduces the time and effort for the discovery of novel pharmaceuticals.

REFERENCES

AGATONOVIC-KUSTRIN, S., BERESFORD, R., 2000. Basic concepts of artificial neural networks modelling and its application in pharmaceutical research. J. Pharmaceutcal and Biomedical Analysis, 22, 717-727.

AJAY, W., WALTERS, P. AND MURCKO, M. A., 1998. Can We Learn To Distinguish between "Drug-like" and "Nondrug-like" Molecules?, J. Med. Chem.**,** 41, 3314-3324.

ALTUNKAYNAK B., ESİN, A., 2004, The genetic algortihm method for parameter estimation in nonlinear regression. G.U. Journal of Science, 17(2), 43-51

ANZALI, S., BARNICKEL, G., CEZANNE, B., KRUG, M., FILIMONOV, D., POROIKOV, V., 2001. Discriminating between drugs and nondrugs by prediction of activity spectra for substances (PASS). J. Med. Chem., 44(15), 2432-2437.

ARCINIEGAS, F., BENNETT, K., BRENEMAN, C., EMBRECHTS, M.J., 2000, Molecular database mining using self-organizing maps for the design of novel pharmaceuticals, Smart Engineering System Design, 10, ASME Press, 477 – 482

ASSINEX GOLD Collection, 2004, Assinex Ltd., Moscow

BAYRAM, E., SANTAGO II, P., HARRIS, R., XIAO, Y., CLAUSET, A., SCHMITT, J.D., 2004. Genetic algorithms and self-organizing maps: a powerful combination for modeling complex QSAR and QSPR problemsJ. Computer-Aided Molecular Design. 18, 483-493.

BEMIS, G.W., MURCKO, M.A., 1996, Properties of known drugs 1: Molecular frameworks. Jour. Med. Chem., 39, 2887-2893.

BEMIS, G.W., MURCKO, M.A., 1999. Properties of known drugs 2: Side chains. Jour. Med. Chem., 42, 5095-5099.

BENEDIKTSSON, J. A., SVEINSSON, J. R., ERSOY, O. K., AND SWAIN, P. H., 1997, Parallel Consensual Neural Networks, IEEE Trans. Neural Network, 8(1).

BENEDIKTSSON, J.A., SWAIN, P.H., ERSOY, O.K., 1990, Neural network approaches versus statistical methods in classification of multisource remote sensing data. IEEE Trans. Geoscience and Remote Sensing, 28(4), 540-552.

BROWN, F.K., 1998, Chemoinformatics: What is it and how does it impact drug discovery. Annu. Rep. Med. Chem., 33, 375-384.

BRÜSTLE, M, BECK, B., SCHINDLER, T., KING, W., MITCHELL, T., CLARK, T., 2002. Descriptors, physical properties, and drug-likeness. J. Med. Chem., 45(16), 3345-3355.

BUDAVARI, S. The Merck Index, 12th Edition, [CD-ROM], Merck & Co., Inc.: Whitehouse Station, NJ, 1996.

BYVATOV, E., FECHNER, U., SADOWSKI, J., SCHNEIDER, G., 2003. Comparison of Support Vector Machine and Artificial Neural Networks Systems for Drug/Nondrug Classification. J. Chem. Inf. Comput. Sci., 43, 1882-1889.

CACOULLOS, T., 1966, Estimation of a multivariate density. Annals of the Institute of Statistical Mathematics, 18(2), 179-189.

CEDENO, W., AGRAFIOTIS, D., 2005. Particle swarms for drug design. IEEE, 1218-1225.

CHEMIDPLUS database, 2006, http://chem.sis.nlm.nih.gov/chemidplus/

CHERKASOV, A., 2006, Can ‘bacterial-metabolite-likeness’ model improve odds of ‘in-slico’ antibiotic discovery? J. Chem. Inf. Model., 46, 1214-1222.

CHIPPERFIELD, A. J., FLEMING, P. J., AND FONSECA, C. M., 1994, Genetic Algorithm Tools for Control Systems Engineering, Proc. Adaptive Computing in Engineering Design and Control, Plymouth Engineering Design Centre, 128-133.

CLARDY, J., WALSH, C., 2004, Lessons from natural molecules, Nature, 432, 829-837.

CLARK, D.E., PICKETT, S.D., 2000, Computational methods fort he prediction of ‘drug-likeness’, Drug Discovery Today, 5(2), 49-58.

COLEY, A.D., 1999, An Introduction to Genetic Algorithms for Scientists and Engineers. World Scientific, Singapore.

DAYLIGHT, 2007, Chemical Information Systems, Inc., Canada, http://www.daylight.com/smiles/index.html

EMBRECHTS, M. J., ARCINIEGAS, F., ÖZDEMIR, M., MOMMA, M., BRENEMAN, C.M., LOCKWOOD, L., BENNETT, K. P., KEWLEY, R.H., 2002, Stripmining For Molecules. *IEEE* 305-310

ENGEL, T., 2006, Basic overview of chemoinformatics. J. Chem. Inf. Model. 46, 2267-2277.

FRIMURER, T. M., BYWATER, R., NÆRUM, L., NØRSKOV LAURITSEN L., AND BRUNAK, S., 2000. Improving the Odds in Discriminating "Drug-like" from "Non Drug-like" Compounds. J. Chem. Inf. Comput. Sci., 40, 1315-1324.

GASTEIGER, J., ENGEL, T., 2003, Chemoinformatics: A textbook. Wiley-VCH, Weinheim.

GASTEIGER, J., TECKENTRUP, A., TERFLOTH, L., SPYCHER, S., 2003, Neural Networks as Data Mining Tools in Drug Design. J. Phy. Org. Chem.16, 232-245.

GUHA, R., 2005, Methods to improve the reliability, validity and interpretability of QSAR models, Phd Thesis, The Penn State Univ.

HANSEN, L. K., SALAMON, P., 1990, Neural Network Ensembles, IEEE Trans. on Pattern Analysis and Machine Intelligence, 12(10).

HAYKIN, S., 1994, Neural Networks, A Comprehensive Foundation, Prentice Hall, Upple Saddle River, N.J.

HOLLAND, J.H., 1975, Adaptation in Natural and Artificial Systems, The University of Michigan Press, Ann Arbor, MI.

HUTTER, M. C., 2007. Separating drugs from nondrugs: A statistical approach using atom pair distribution. J. Chem. Inf. Model, 47, 186-194.

J. ANTIBIOT. database, 2006, http://www.nih.go.jp/~jun/NADB/byname.html

JÓNSDÓTTIR, S. Ó., JØRGENSEN, F. S., AND BRUNAK, S., 2005, Prediction methods and databases within chemoinformatics: emphasis on drugs and drug candidates, Bioinformatics, 21(10), 2145-2160.

KADAM, R.U., ROY, N., 2007, Recent trends in drug-likeness prediction: A comprehensive review of *in slico* methods. Indian Journal of Pharmaceutical Science, 609-615.

KARAKOC, E., SAHINALP, C., CHERKASOV, A., 2006. Comparative QSAR and Fragments Distribution Analysis of Drugs, Drug Likes, Metabolic Substances and Antimicrobial Compounds. Journal of Chemical Information and Modeling, 46(5), 2167-2182.

KELLER, H.T., PICHOTA, A., YIN, Z., 2007, A practical view of 'druggability', Current Opinion in Chemical Biology, 10, 357-361.

KITTLER, J., HATEF, M., DUIN, R. P. W., AND MATAS, J., 1998. On Combining Classifiers, IEEE Trans. on Pattern Analysis and Machine Intelligence, 20(3), 226-239.

KOHONEN, T., 1990. The Self Organizing Map. Proc. IEEE, 78(9), 1464-1480

LEE, J., ERSOY, O. K., 2007. Consensual and Hierarchical Classification of Remotely Sensed Multispectral Images, IEEE Trans. Geosci. Remote Sens., 45(9), 2953-2963.

LI, A.P., 2005. Preclinical in vitro screening assays for drug-like properties. Drug Discovery Today: Technologies. 2, 2, 179-185.

LI, Q., BENDER, A., PEI, J., LAI, L., 2007. A large descriptor set and a probabilistic kernel-based classifier significantly improve druglikeness classification. J. Chem. Inf. Model.

LIPINSKI, C.A, LOMBARDO, F., DOMINY, B.W., FEENEY, P.J., 1997. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. Advanced Drug Delivery Reviews, 46, 3-26

LIPINSKI, C.A., 2003, Physicochemical properties and the discovery of orally active drugs: technical and people issues., Molecular Informatics Confronting Complexity.Proceedings of the Beilstein International Workshop, Logos Verlag, Berlin.

LIPINSKI, C.A., 2004, Lead- and drug-like compounds: the rule-of-five revolution., Drud Discovery Today: Technologies, 1(4), 337-341.

LIU, Y., 2005. Drug design by machine learning: Ensemble learning for QSAR modeling. IEEE Proceed. the 4th Int. Conference on Machine Learning and Applications (ICMLA 05)

MANALLACK, D.T., TEHAN, B.G., GANCIA, E., HUDSON, B.D., FORD, M.G., LIVINGSTONE, D.J., WHITLEY, D.C. AND PITT, W.R., 2003. A consensus neural network-based technique for discriminating soluble and poorly soluble compounds. J. Chem. Inf. Comput. Sci. 43, 674-679.

MATTER H., BARIGHAUS, K.H., NAUMANN T., KLABUNDE T., PIRARD B., 2001, Computational approaches towards the rational design of drug-like compound libraries. Comb. Chem. & High Throughput Screen., 4(6), 453-475

MERCK INDEX, 2004, 13.4th Edition, [CD-ROM], Merck & Co., Inc. and CambridgeSoft

MERCK INDEX, 14th Edition, The web accessible version of the Merck Index is co-published by Merck & Co., Inc. and CambridgeSoft.

MOE, Molecular Operational Environment, 2006, Chemical Computing Group Inc., Montreal, Canada.

MUEGGE, I., HEALD, S.L., BRITELLI, D., 2001. Simple selection criteria for drug-like chemical matter. J. Med. Chem., 44(12), 1841-1846.

MURCIA-SOLER, M., PE´REZ-GIMENEZ, F., GARCIA-M., J., SALABERT-SALVADOR, M. T., DIAZ-VILLANUEVA, W., CASTRO-BLEDA, M. J., 2003. Drugs and Nondrugs: An Effective Discrimination with Topological Methods and Artificial Neural Networks. J. Chem. Inf. Comput. Sci., 43, 1688-1702.

NCBI, National Center for Biotechnology Information, 2007. PubChem, http://www.ncbi.nlm.nih.gov/PubMed/.

NEGNEVITSKY, M., 2005. Artificial Intelligence: A Guide to Intelligent Systems. 2/E. Addison-Wesley.

PARIS, G., 1999. Meeting of the American Chemical Society, quoted by W.Warr at http://www.warr.com/warrzone2000.html

PARZEN, E., 1962. On Estimation of a Probability Density Function and Mode. Ann. Math. Stat., 33, 1065-1076.

RICHON, A.B., YOUNG, S.S., 1997. An introduction to QSAR methodology. Network Science. www.netsci.org/Science/Compchem/feature19.html.

ROTHLAUF, F., 2006. Representations for Genetic and Evolutionary Algorithms. Springer, Netherlands.

SADOWSKI, J. AND KUBINYI, H., 1998. A scoring scheme for discriminating between drugs and nondrugs. J. Med. Chem., 41, 3325-3329.

SAGLAM, M.I, ERSOY, O.K., ERER, I., 2004, Self Organizing Global Ranking and its Applications. Proceedings of the Conference on Smart Engineering Design (ANNIE 04), St. Louis.

SCHNEIDER, G., WREDE, P., 1998, Artificial neural networks for computer-based molecular design. Progress in Biophysics & Molecular Biology, 70, 175-222.

SHARKEY, A. J. C., 1996, On combining artificial neural nets, Connect. Sci., 8(3), 299-314

SO, S., KARPLUS, M., 1996, Evolutionary optimization in quantitive structure-activity relationship: an aplication of genetic neural networks. J. Med. Chem, 39, 1521-1530.

SPECHT, D.F., 1991, A General Regression Neural Network, IEEE Trans. Neural Networks, 2(6), 568-576.

SPECHT, D.F., 1992, Enhancements to Probabilistic Neural Networks, IEEE, I-761-768.

TAKAOKA, Y., ENDO, Y., YAMANOBE, S., KAKINUMA, H., OKUBO, T., SHIMAZAKI, Y., OTA, T., SUMIYA, S., YOSHIKAWA, K., 2003. Development of a method for evaluating drug-likeness and ease of synthesis using a data set in which compounds are assigned scores based on chemists' intuition. J. Chem. Inf. Comput. 43(4), 1269-1275.

TERFLOTH, L., GASTEIGER, J., 2001. Neural networks and genetic algorithms in drug design. Drug Discovery Today. 6, 15, 102-107.

TOMANDL, D., SCHOBER, A., 2001, A Modified General Regression Neural Network (MGRNN) with New, Efficient Training Algorithms as a Robust 'black box'-tool for Data Analysis, Neural Networks, 14, 1023-1034.

VEBER, D.F:, JOHNSON, S.R., CHENG, H.Y., SMITH, B.R:, WARD, K.W:, KOPPLE, K.D., 2002, Molecular properties that influence the oral bioavailability or drugs, J. Med. Chem., 45, 2615-2623

VERTAN, C., MALCIU, M., BUZULOIU, V., POPESCU, V., 1996, Median Filtering Techniques for Vector Valued Signals, Proc. IEEE Intern. Conf. on Image Processing ICIP' 96, 1, 977-980, Lausanne, Switzerland.

WAGENER, M., van GEERESTEIN, V.J., 2000. Potential drug and nondrugs: prediction and identification of important structural features. J. Chem. Inf. Comput. Science, 40, 280-292.

WALTERS, W.P., MURCKO, M.A., 2002, Prediction of 'drug-likeness', Advanced Drug Delivery Reviews, 54, 255-271.

WISHART, D., 2005. The bioinformatics of small molecules. University of Alberta, Canada.

XU, L., KRZYZAK, A., AND SUEN, C. Y., 1992. Methods of Combining Multiple Classifier and Their Applications to Handwriting Recognition. IEEE Trans. Syst., Man, Cybern., 22(3), 418-435.

Yang, Z. R., Thomson, R., McNeil P., and Esnouf, R. M., 2005. RONN: the Bio-basis Function Neural Network Technique Applied to the Detection of Natively Disordered Regions in Proteins. Bioinformatics, 21(16), 3369–3376.

BIOGRAPHY

Ayça ÇAKMAK PEHLİVANLI received the B.S. degree in statistics from Hacettepe University in 1997 and the M.S. degree in 2001 from Syracuse University, Syracuse, USA in the field of computer and information science. She worked as a research assistant in the Department of Electrical Electronic Engineering at Cukurova University, between December 2001 and September 2006. She is currently working as a research assistant in Computer Engineering Department at Istanbul Kultur University since 2006. She has been involved in research in statistical and computational intelligence, molecular modeling, chemoinformatics and bioinformatics.

APPENDIX

1. Physical Properties

The following physical properties can be calculated from the connection table (with no dependence on conformation) of a molecule(*):

Code	Description
AM1_dipole	The dipole moment calculated using the AM1 Hamiltonian [MOPAC].
AM1_E	The total energy (kcal/mol) calculated using the AM1 Hamiltonian [MOPAC].
AM1_Eele	The electronic energy (kcal/mol) calculated using the AM1 Hamiltonian [MOPAC].
AM1_HF	The heat of formation (kcal/mol) calculated using the AM1 Hamiltonian [MOPAC].
AM1_IP	The ionization potential (kcal/mol) calculated using the AM1 Hamiltonian [MOPAC].
AM1_LUMO	The energy (eV) of the Lowest Unoccupied Molecular Orbital calculated using the AM1 Hamiltonian [MOPAC].
AM1_HOMO	The energy (eV) of the Highest Occupied Molecular Orbital calculated using the AM1 Hamiltonian [MOPAC].
Apol	Sum of the atomic polarizabilities (including implicit hydrogens) with polarizabilities taken from [CRC 1994].
Bpol	Sum of the absolute value of the difference between atomic polarizabilities of all bonded atoms in the molecule (including implicit hydrogens) with polarizabilities taken from [CRC 1994].
Density	Molecular mass density: Weight divided by vdw_vol.
FCharge	Total charge of the molecule (sum of formal charges).
MNDO_dipole	The dipole moment calculated using the MNDO Hamiltonian [MOPAC].
MNDO_E	The total energy (kcal/mol) calculated using the MNDO Hamiltonian [MOPAC].
MNDO_Eele	The electronic energy (kcal/mol) calculated using the MNDO Hamiltonian [MOPAC].
MNDO_HF	The heat of formation (kcal/mol) calculated using the MNDO Hamiltonian [MOPAC].
MNDO_IP	The ionization potential (kcal/mol) calculated using the MNDO Hamiltonian [MOPAC].
MNDO_LUMO	The energy (eV) of the Lowest Unoccupied Molecular Orbital calculated using the MNDO Hamiltonian [MOPAC].
MNDO_HOMO	The energy (eV) of the Highest Occupied Molecular Orbital calculated using the MNDO Hamiltonian [MOPAC].
Mr	Molecular refractivity (including implicit hydrogens). This property is calculated from an 11 descriptor linear model [MREF 1998] with r2 = 0.997, RMSE = 0.168 on 1,947 small molecules.
PM3_dipole	The dipole moment calculated using the PM3 Hamiltonian [MOPAC].
PM3_E	The total energy (kcal/mol) calculated using the PM3 Hamiltonian [MOPAC].
PM3_Eele	The electronic energy (kcal/mol) calculated using the PM3 Hamiltonian [MOPAC].
PM3_HF	The heat of formation (kcal/mol) calculated using the PM3 Hamiltonian [MOPAC].
PM3_IP	The ionization potential (kcal/mol) calculated using the PM3 Hamiltonian [MOPAC].
PM3_LUMO	The energy (eV) of the Lowest Unoccupied Molecular Orbital calculated using the PM3 Hamiltonian [MOPAC].
PM3_HOMO	The energy (eV) of the Highest Occupied Molecular Orbital calculated using the PM3 Hamiltonian [MOPAC].
SMR	Molecular refractivity (including implicit hydrogens). This property is an atomic contribution model [Crippen 1999] that assumes the correct protonation state (washed structures). The model was trained on ~7000 structures and results may vary from the mr descriptor.
Weight	Molecular weight (including implicit hydrogens) with atomic weights taken from [CRC 1994].
logP(o/w)	Log of the octanol/water partition coefficient (including implicit hydrogens). This property is calculated from a linear atom type model [LOGP 1998] with r2 = 0.931, RMSE=0.393 on 1,827 molecules.
Logs	Log of the aqueous solubility This property is calculated from an atom contribution linear atom type model [Hou 2004] with r2 = 0.90, ~1,200 molecules.
Reactive	Indicator of the presence of reactive groups. A non-zero value indicates that the molecule contains a reactive group. The table of reactive groups is based on the Oprea set [Oprea 2000] and includes metals, phospho-, N/O/S-N/O/S single bonds, thiols, acyl halides, Michael Acceptors, azides, esters, etc.
SlogP	Log of the octanol/water partition coefficient (including implicit hydrogens). This property is an atomic contribution model [Crippen 1999] that calculates logP from the given structure; i.e., the correct protonation state (washed structures). Results may vary from the logP(o/w) descriptor. The training set for SlogP was ~7000 structures.
TPSA	Polar surface area calculated using group contributions to approximate the polar surface area from connection table information only. The parameterization is that of Ertl et al. [Ertl 2000].
vdw_vol	Van der Waals volume calculated using a connection table approximation.
vdw_area	Area of van der Waals surface calculated using a connection table approximation.

(*) CRC Handbook of Chemistry and Physics. CRC Pres, 1994.

ERTL, P., ROHDE, B., SELZER, P. 2000. Fast Calculation of Molecular Polar Surface Area as a Sum of Fragment-Based Contributions and Its Application to the Prediction of Drug Transport Properties. J. Med. Chem. 43, 3714-3717.

LABUTE, P. 1998. MOE LogP(Octanol/Water) Model. unpublished. Source code in $MOE/lib/svl/quasar.svl/q_logp.svl

2. Subdivided Surface Areas

The Subdivided Surface Areas are descriptors based on an approximate accessible van der Waals surface area calculation for each atom, v_i along with some other atomic property, p_i. The v_i are calculated using a connection table approximation. Each descriptor in a series is defined to be the sum of the v_i over all atoms i such that p_i is in a specified range (a,b].
In the descriptions to follow, L_i denotes the contribution to logP(o/w) for atom i as calculated in the SlogP descriptor [Crippen 1999]. R_i denotes the contribution to Molar Refractivity for atom i as calculated in the SMR descriptor [Crippen 1999]. The ranges were determined by percentile subdivision over a large collection of compounds(*).

Code	Description
SlogP_VSA0	Sum of vi such that Li <= -0.4.
SlogP_VSA1	Sum of vi such that Li is in (-0.4,-0.2].
SlogP_VSA2	Sum of vi such that Li is in (-0.2,0].
SlogP_VSA3	Sum of vi such that Li is in (0,0.1].
SlogP_VSA4	Sum of vi such that Li is in (0.1,0.15].
SlogP_VSA5	Sum of vi such that Li is in (0.15,0.20].
SlogP_VSA6	Sum of vi such that Li is in (0.20,0.25].
SlogP_VSA7	Sum of vi such that Li is in (0.25,0.30].
SlogP_VSA8	Sum of vi such that Li is in (0.30,0.40].
SlogP_VSA9	Sum of vi such that Li > 0.40.
SMR_VSA0	Sum of vi such that Ri is in [0,0.11].
SMR_VSA1	Sum of vi such that Ri is in (0.11,0.26].
SMR_VSA2	Sum of vi such that Ri is in (0.26,0.35].
SMR_VSA3	Sum of vi such that Ri is in (0.35,0.39].
SMR_VSA4	Sum of vi such that Ri is in (0.39,0.44].
SMR_VSA5	Sum of vi such that Ri is in (0.44,0.485].
SMR_VSA6	Sum of vi such that Ri is in (0.485,0.56].
SMR_VSA7	Sum of vi such that Ri > 0.56.

(*)WILDMAN, S.A., CRIPPEN, G.M., 1999. Prediction of Physiochemical Parameters by Atomic Contributions. J. Chem. Inf. Comput. Sci. 39, No. 5, 868-873.

3. Pharmacophore Feature Descriptors

The Pharmacophore Atom Type descriptors consider only the heavy atoms of a molecule and assign a type to each atom. That is, hydrogens are suppressed during the calculation. The feature set is Donor, Acceptor, Polar (both Donor and Acceptor), Positive (base), Negative (acid), Hydrophobe and Other. Assignments may take into account implied protonation, deprotonation, keto/enol considerations and tautomerism at a biologically relevant pH. For example, -COOH will be typed in its deprotonated form regardless of how the structure is stored.

Code	Description
a_acc	Number of hydrogen bond acceptor atoms (not counting acidic atoms but counting atoms that are both hydrogen bond donors and acceptors such as -OH).
a_acid	Number of acidic atoms.
a_base	Number of basic atoms.
a_don	Number of hydrogen bond donor atoms (not counting basic atoms but counting atoms that are both hydrogen bond donors and acceptors such as -OH).
a_hyd	Number of hydrophobic atoms.
vsa_acc	Approximation to the sum of VDW surface areas of pure hydrogen bond acceptors (not counting acidic atoms and atoms that are both hydrogen bond donors and acceptors such as -OH).
vsa_acid	Approximation to the sum of VDW surface areas of acidic atoms.
vsa_base	Approximation to the sum of VDW surface areas of basic atoms.
vsa_don	Approximation to the sum of VDW surface areas of pure hydrogen bond donors (not counting basic atoms and atoms that are both hydrogen bond donors and acceptors such as -OH).
vsa_hyd	Approximation to the sum of VDW surface areas of hydrophobic atoms.
vsa_other	Approximation to the sum of VDW surface areas of atoms typed as "other".
vsa_pol	Approximation to the sum of VDW surface areas of polar atoms (atoms that are both hydrogen bond donors and acceptors), such as -OH.

4. Atom Counts and Bond Counts

The atom count and bond count descriptors are functions of the counts of atoms and bonds (subdivided according to various criteria)(*).

Code	Description
a_aro	Number of aromatic atoms.
a_count	Number of atoms (including implicit hydrogens). This is calculated as the sum of (1 + hi) over all non-trivial atoms i.
a_heavy	Number of heavy atoms #{Zi | Zi > 1}.
a_ICM	Atom information content (mean). This is the entropy of the element distribution in the molecule (including implicit hydrogens but not lone pair pseudo-atoms). Let ni be the number of occurrences of atomic number i in the molecule. Let pi = ni / n where n is the sum of the ni. The value of a_ICM is the negative of the sum over all i of pi log pi.
a_IC	Atom information content (total). This is calculated to be a_ICM times n.
a_nH	Number of hydrogen atoms (including implicit hydrogens). This is calculated as the sum of hi over all non-trivial atoms i plus the number of non-trivial hydrogen atoms.
a_nB	Number of boron atoms: #{Zi | Zi = 5}.
a_nC	Number of carbon atoms: #{Zi | Zi = 6}.
a_nN	Number of nitrogen atoms: #{Zi | Zi = 7}.
a_nO	Number of oxygen atoms: #{Zi | Zi = 8}.
a_nF	Number of fluorine atoms: #{Zi | Zi = 9}.
a_nP	Number of phosphorus atoms: #{Zi | Zi = 15}.
a_nS	Number of sulfur atoms: #{Zi | Zi = 16}.
a_nCl	Number of chlorine atoms: #{Zi | Zi = 17}.
a_nBr	Number of bromine atoms: #{Zi | Zi = 35}.
a_nI	Number of iodine atoms: #{Zi | Zi = 53}.
b_1rotN	Number of rotatable single bonds. Conjugated single bonds are not included (e.g., ester and peptide bonds).
b_1rotR	Fraction of rotatable single bonds: b_1rotN divided by b_heavy.
b_ar	Number of aromatic bonds.
b_count	Number of bonds (including implicit hydrogens). This is calculated as the sum of (di/2 + hi) over all non-trivial atoms i.
b_double	Number of double bonds. Aromatic bonds are not considered to be double bonds.
b_heavy	Number of bonds between heavy atoms.
b_rotN	Number of rotatable bonds. A bond is rotatable if it has order 1, is not in a ring, and has at least two heavy neighbors.
b_rotR	Fraction of rotatable bonds: b_rotN divided by b_heavy.

Code	Description
b_single	Number of single bonds (including implicit hydrogens). Aromatic bonds are not considered to be single bonds.
b_triple	Number of triple bonds. Aromatic bonds are not considered to be triple bonds.
chiral	The number of chiral centers.
chiral_u	The number of unconstrained chiral centers.
Lip_acc	The number of O and N atoms.
Lip_don	The number of OH and NH atoms.
Lip_druglike	One if and only if lip_violation < 2 otherwise zero.
Lip_violation	The number of violations of Lipinski's Rule of Five [Lipinski 1997].
nmol	The number of molecules (connected components).
Opr_brigid	The number of rigid bonds bonds from [Oprea 2000].
Opr_leadlike	One if and only if opr_violation < 2 otherwise zero.
Opr_nring	The number of rings bonds from [Oprea 2000].
Opr_nrot	The number of rotatable bonds from [Oprea 2000].
Opr_violation	The number of violations of Oprea's lead-like test [Oprea 2000].
rings	The number of rings.
VAdjMa	Vertex adjacency information (magnitude): 1 + log2 m where m is the number of heavy-heavy bonds. If m is zero, then zero is returned.
VAdjEq	Vertex adjacency information (equality): -(1-f)log2(1-f) - f log2 f where f = (n2 - m) / n2, n is the number of heavy atoms and m is the number of heavy-heavy bonds. If f is not in the open interval (0,1), then 0 is returned.

(*) LIPINSKI, C.A., LOMBARDO, F., DOMINY, B.W. AND FEENEY, P.J.; 1997. Experimental and Computational Approaches to Estimate Solubility and Permeability in Drug Discovery and Development Settings; Adv. Drug Deliv. Rev. 23, 3-25.

OPREA, TUDOR I, 2000.. Property Distribution of Drug-Related Chemical Databases. J. Comp. Aid. Mol. Des. 14, 251-264

5. Kier&Hall Connectivity and Kappa Shape Indices

For a heavy atom *i* let $v_i = (p_i - h_i) / (Z_i - p_i - 1)$ where p_i is the number of s and p valence electrons of atom *i*. The Kier and Hall chi connectivity indices are calculated from the d_i and v_i values. The Kier and Hall kappa molecular shape indices [Hall 1991] compare the molecular graph with minimal and maximal molecular graphs, and are intended to capture different aspects of molecular shape. In the following description, *n* denotes the number of atoms in the hydrogen suppressed graph, *m* is the number of bonds in the hydrogen suppressed graph and *a* is the sum of $(r_i/r_c - 1)$ where r_i is the covalent radius of atom *i*, and r_c is the covalent radius of a carbon atom. Also, let p_2 denote the number of paths of length 2 and p_3 the number of paths of length 3 (*).

Code	Description
chi0	Atomic connectivity index (order 0) from [Hall 1991] and [Hall 1977]. This is calculated as the sum of 1/sqrt(di) over all heavy atoms i with di > 0.
chi0_C	Carbon connectivity index (order 0). This is calculated as the sum of 1/sqrt(di) over all carbon atoms I with di > 0.
chi1	Atomic connectivity index (order 1) from [Hall 1991] and [Hall 1977]. This is calculated as the sum of 1/sqrt(didj) over all bonds between heavy atoms i and j where i < j.
chi1_C	Carbon connectivity index (order 1). This is calculated as the sum of 1/sqrt(didj) over all bonds between carbon atoms i and j where i < j.
chi0v	Atomic valence connectivity index (order 0) from [Hall 1991] and [Hall 1977]. This is calculated as the sum of 1/sqrt(vi) over all heavy atoms i with vi > 0.
chi0v_C	Carbon valence connectivity index (order 0). This is calculated as the sum of 1/sqrt(vi) over all carbon atoms i with vi > 0.
chi1v	Atomic valence connectivity index (order 1) from [Hall 1991] and [Hall 1977]. This is calculated as the sum of 1/sqrt(vivj) over all bonds between heavy atoms i and j where i < j.
chi1v_C	Carbon valence connectivity index (order 1). This is calculated as the sum of 1/sqrt(vivj) over all bonds between carbon atoms i and j where i < j.
Kier1	First kappa shape index: (n-1)2 / m2 [Hall 1991].
Kier2	Second kappa shape index: (n-1)2 / m2 [Hall 1991].
Kier3	Third kappa shape index: (n-1) (n-3)2 / p32 for odd n, and (n-3) (n-2)2 / p32 for even n [Hall 1991].

Code	Description
KierA1	First alpha modified shape index: s (s-1)2 / m2 where s = n + a [Hall 1991].
KierA2	Second alpha modified shape index: s (s-1)2 / m2 where s = n + a [Hall 1991].
KierA3	Third alpha modified shape index: (n-1) (n-3)2 / p32 for odd n, and (n-3) (n-2)2 / p32 for even n where s = n + a [Hall 1991].
KierFlex	Kier molecular flexibility index: (KierA1) (KierA2) / n [Hall 1991].
zagreb	Zagreb index: the sum of di2 over all heavy atoms i.

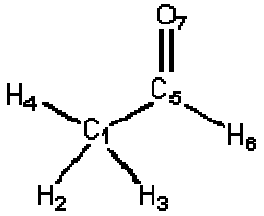
(*)HALL, L.H., KIER, L.B., 1991. The Molecular Connectivity Chi Indices and Kappa Shape Indices in Structure-Property Modeling. Reviews of Computational Chemistry. 2.

HALL, L.H., KIER, L.B., 1977. The Nature of Structure-Activity Relationships and Their Relation to Molecular Connectivity. Eur. J. Med. Chem. 12, 307.

6. Adjacency and Distance Matrix Descriptors

The *adjacency matrix*, *M*, of a chemical structure is defined by the elements [M_{ij}] where M_{ij} is 1 if atoms *i* and *j* are bonded and zero otherwise. The *distance matrix*, *D*, of a chemical structure is defined by the elements [D_{ij}] where D_{ij} is the length of the shortest path from atoms *i* to *j*; zero is used if atoms *i* and *j* are not part of the same connected component. The adjacency matrix of CH3CH=O is displayed on the left and its distance matrix is displayed on the right (below):

```
C1      0 1 1 1 1 0 0      0 1 1 1 1 2 2
H2      1 0 0 0 0 0 0      1 0 2 2 2 3 3
H3      1 0 0 0 0 0 0      1 2 0 2 2 3 3
H4      1 0 0 0 0 0 0      1 2 2 0 2 3 3
C5      1 0 0 0 0 1 1      1 2 2 2 0 1 1
H6      0 0 0 0 1 0 0      2 3 3 3 1 0 2
O7      0 0 0 0 1 0 0      2 3 3 3 1 2 0
```



Petitjean [Petitjean 1992] defines the **eccentricity** of a vertex to be the longest path from that vertex to any other vertex in the graph. The graph **radius** is the smallest vertex eccentricity in the graph and the graph **diameter** as the largest vertex eccentricity. These values are calculated using the distance matrix and are used for several descriptors described below.

The following descriptors are calculated from the distance and adjacency matrices of the heavy atoms(*):

Code	Description
balabanJ	Balaban's connectivity topological index [Balaban 1982].
BCUT_PEOE_0 BCUT_PEOE_1 BCUT_PEOE_2 BCUT_PEOE_3	The BCUT descriptors [Pearlman&1998] are calculated from the eigenvalues of a modified adjacency matrix. Each ij entry of the adjacency matrix takes the value 1/sqrt(bij) where bij is the formal bond order between bonded atoms i and j. The diagonal takes the value of the PEOE partial charges. The resulting eigenvalues are sorted and the smallest, 1/3-ile, 2/3-ile and largest eigenvalues are reported.
BCUT_SLOGP_0 BCUT_SLOGP_1 BCUT_SLOGP_2 BCUT_SLOGP_3	The BCUT descriptors using atomic contribution to logP (using the Wildman and Crippen SlogP method) instead of partial charge.
BCUT_SMR_0 BCUT_SMR_1 BCUT_SMR_2 BCUT_SMR_3	The BCUT descriptors using atomic contribution to molar refractivity (using the Wildman and Crippen SMR method) instead of partial charge.
Diameter	Largest value in the distance matrix [Petitjean 1992].
Petitjean	Value of (diameter - radius) / diameter.

Code	Description
GCUT_PEOE_0 GCUT_PEOE_1 GCUT_PEOE_2 GCUT_PEOE_3	The GCUT descriptors are calculated from the eigenvalues of a modified graph distance adjacency matrix. Each ij entry of the adjacency matrix takes the value 1/sqr(dij) where dij is the (modified) graph distance between atoms i and j. The diagonal takes the value of the PEOE partial charges. The resulting eigenvalues are sorted and the smallest, 1/3-ile, 2/3-ile and largest eigenvalues are reported.
GCUT_SLOGP_0 GCUT_SLOGP_1 GCUT_SLOGP_2 GCUT_SLOGP_3	The GCUT descriptors using atomic contribution to logP (using the Wildman and Crippen SlogP method) instead of partial charge.
GCUT_SMR_0 GCUT_SMR_1 GCUT_SMR_2 GCUT_SMR_3	The GCUT descriptors using atomic contribution to molar refractivity (using the Wildman and Crippen SMR method) instead of partial charge.
petitjeanSC	Petitjean graph Shape Coefficient as defined in [Petitjean 1992]: (diameter - radius) / radius.
Radius	If ri is the largest matrix entry in row i of the distance matrix D, then the radius is defined as the smallest of the ri [Petitjean 1992].
VDistEq	If m is the sum of the distance matrix entries then VdistEq is defined to be the sum of log2 m - pi log2 pi / m where pi is the number of distance matrix entries equal to i.
VDistMa	If m is the sum of the distance matrix entries then VDistMa is defined to be the sum of log2 m - Dij log2 Dij / m over all i and j.
weinerPath	Wiener path number: half the sum of all the distance matrix entries as defined in [Balaban 1979] and [Wiener 1947].
weinerPol	Wiener polarity number: half the sum of all the distance matrix entries with a value of 3 as defined in [Balaban 1979].

(*)BALABAN, A.T., 1979. Five New Topological Indices for the Branching of Tree-Like Graphs; Theoretica Chimica Acta. 53, 355-375.

BALABAN, A.T., 1982. Highly Discriminating Distance-Based Topological Index; Chemical Physics Letters. 89, 5, 399-404.

PETITJEAN, M. Applications of the Radius-Diameter Diagram to the Classification of Topological and Geometrical Shapes of Chemical Compounds. J. Chem. Inf. Comput. Sci. 32, 331-337 (1992).

WIENER, H., 1947. Structural Determination of Paraffin Boiling Points. Journal of the American Chemical Society. 69, 17-20.

7. Partial Charge Descriptors

Descriptors that depend on the partial charge of each atom of a chemical structure require calculation of those partial charges. An unfortunate complication is the fact that there are numerous methods of calculating partial charges. Rather than enforce a particular method, MOE provides several versions of most of the charge-dependent descriptors. The only difference between these variants is the source of the partial charges. The following variants are supported: PEOE, Q

PEOE. The Partial Equalization of Orbital Electronegativities (PEOE) method of calculating atomic partial charges [Gasteiger 1980] is a method in which charge is transferred between bonded atoms until equilibrium.

Q. Descriptors prefixed with Q_ use the partial charges stored with each structure in the database. In other words, no partial charge calculation is made and it is assumed that some external program has been used to calculate the atomic partial charges. This dependence can be a subtle source of error if, for example, the wrong charges are stored when descriptors are recalculated (*e.g.*, when evaluating QSAR models on novel structures).

Partial charges from forcefields can be used by a) energy minimizing the database structures (which will store the charges); b) using the Q_ variant of the descriptors (*).

Code	Description
Q_PC+ PEOE_PC+	Total positive partial charge: the sum of the positive qi. Q_PC+ is identical to PC+ which has been retained for compatibility.
Q_PC- PEOE_PC-	Total negative partial charge: the sum of the negative qi. Q_PC- is identical to PC- which has been retained for compatibility.
Q_RPC+ PEOE_RPC+	Relative positive partial charge: the largest positive qi divided by the sum of the positive qi. Q_RPC+ is identical to RPC+ which has been retained for compatibility.
Q_PRC- PEOE_RPC-	Relative negative partial charge: the smallest negative qi divided by the sum of the negative qi. Q_RPC- is identical to RPC- which has been retained for compatibility.
Q_VSA_POS PEOE_VSA_POS	Total positive van der Waals surface area. This is the sum of the vi such that qi is non-negative. The vi are calculated using a connection table approximation.
Q_VSA_NEG PEOE_VSA_NEG	Total negative van der Waals surface area. This is the sum of the vi such that qi is negative. The vi are calculated using a connection table approximation.
Q_VSA_PPOS PEOE_VSA_PPOS	Total positive polar van der Waals surface area. This is the sum of the vi such that qi is greater than 0.2. The vi are calculated using a connection table approximation.
Q_VSA_PNEG PEOE_VSA_PNEG	Total negative polar van der Waals surface area. This is the sum of the vi such that qi is less than -0.2. The vi are calculated using a connection table approximation.
Q_VSA_HYD PEOE_VSA_HYD	Total hydrophobic van der Waals surface area. This is the sum of the vi such that |qi| is less than or equal to 0.2. The vi are calculated using a connection table approximation.
Q_VSA_POL PEOE_VSA_POL	Total polar van der Waals surface area. This is the sum of the vi such that |qi| is greater than 0.2. The vi are calculated using a connection table approximation.
Q_VSA_FPOS PEOE_VSA_FPOS	Fractional positive van der Waals surface area. This is the sum of the vi such that qi is non-negative divided by the total surface area. The vi are calculated using a connection table approximation.
Q_VSA_FNEG PEOE_VSA_FNEG	Fractional negative van der Waals surface area. This is the sum of the vi such that qi is negative divided by the total surface area. The vi are calculated using a connection table approximation.
Q_VSA_FPPOS PEOE_VSA_FPPOS	Fractional positive polar van der Waals surface area. This is the sum of the vi such that qi is greater than 0.2 divided by the total surface area. The vi are calculated using a connection table approximation.
Q_VSA_FPNEG PEOE_VSA_FPNEG	Fractional negative polar van der Waals surface area. This is the sum of the vi such that qi is less than -0.2 divided by the total surface area. The vi are calculated using a connection table approximation.
Q_VSA_FHYD PEOE_VSA_FHYD	Fractional hydrophobic van der Waals surface area. This is the sum of the vi such that |qi| is less than or equal to 0.2 divided by the total surface area. The vi are calculated using a connection table approximation.
Q_VSA_FPOL PEOE_VSA_FPOL	Fractional polar van der Waals surface area. This is the sum of the vi such that |qi| is greater than 0.2 divided by the total surface area. The vi are calculated using a connection table approximation.
PEOE_VSA+6	Sum of vi where qi is greater than 0.3.
PEOE_VSA+5	Sum of vi where qi is in the range [0.25,0.30).
PEOE_VSA+4	Sum of vi where qi is in the range [0.20,0.25).
PEOE_VSA+3	Sum of vi where qi is in the range [0.15,0.20).
PEOE_VSA+2	Sum of vi where qi is in the range [0.10,0.15).
PEOE_VSA+1	Sum of vi where qi is in the range [0.05,0.10).
PEOE_VSA+0	Sum of vi where qi is in the range [0.00,0.05).
PEOE_VSA-0	Sum of vi where qi is in the range [-0.05,0.00).
PEOE_VSA-1	Sum of vi where qi is in the range [-0.10,-0.05).
PEOE_VSA-2	Sum of vi where qi is in the range [-0.15,-0.10).
PEOE_VSA-3	Sum of vi where qi is in the range [-0.20,-0.15).
PEOE_VSA-4	Sum of vi where qi is in the range [-0.25,-0.20).
PEOE_VSA-5	Sum of vi where qi is in the range [-0.30,-0.25).
PEOE_VSA-6	Sum of vi where qi is less than -0.30.

(*) GASTEIGER, J., MARSILI, M., 1980. Iterative Partial Equalization of Orbital Electronegativity - A Rapid Access to Atomic Charges. Tetrahedron. 36, 3219