

**DEVELOPMENT AND APPLICATION OF LIGAND-BASED AND STRUCTURE-
BASED COMPUTATIONAL DRUG DISCOVERY TOOLS BASED ON
FREQUENT SUBGRAPH MINING OF CHEMICAL STRUCTURES**

Raed Saeed Khashan

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the School of Pharmacy (Division of Medicinal Chemistry and Natural Products).

Chapel Hill
2007

Approved by:

Dr. Alexander Tropsha

Dr. Weifan Zheng

Dr. Alexander Golbraikh

Dr. Wei Wang

Dr. Andrew Lee

©2007
Raed Saeed Khashan
ALL RIGHTS RESERVED

ABSTRACT

RAED KHASHAN: Development and Application of Ligand-based and Structure-based Computational Drug Discovery Tools Based on Frequent Subgraph Mining of Chemical Structures

(Under the direction of Alexander Tropsha)

Recent development in subgraph mining tools resulted in faster and more efficient algorithms that facilitate exploring the information encoded in data that can be represented by graphs. In this dissertation, we apply the graph mining technique to design ligand-based and structure-based computational drug discovery tools. For ligand-based drug design, molecules in a dataset will be represented by graphs, and subgraph mining tools will be used to find the frequent subgraphs (chemical fragments) that occur in at least a certain percentage of the ligands in the dataset. These chemical fragments will be used as molecular descriptors for the quantitative structure-activity relationship (QSAR) studies. They will also be used for identifying the pharmacophores responsible for the activity as well as the toxicophores responsible for the toxicity of a datasets of molecules. For the structure-based drug design, interacting atoms at the interface of a set of protein-ligand complexes will be represented by graphs. Frequent subgraphs identified will define the patterns of chemical interactions at the interface, which will be used to pose-score docked complexes to identify the correct docking pose.

*This dissertation is dedicated to
my Mother, Father, and all my family members,
whose support, encouragement, and personal sacrifice
have made this research possible.*

ACKNOWLEDGEMENTS

Dr. Alexander Tropsha for his scientific guidance, support and encouragement during my graduate studies.

Dr. Weifan Zheng, for his concern, scientific guidance as well as his advices, especially those related to my future career.

Drs. Alexander Golbraikh, Wei Wang, and Andrew Lee for their time and effort in assisting, and guiding this research project.

Labmates in 301 Beard Hall; Dr. Min Shen, Dr. Scott Oloff, Dr. Shuxiang (King) Zhang, Chrisopher Grulke, Rima Hajjo, and all my labmates for their friendship, support, and scientific discussion over the years.

My former adviser, Professor Robert Pearlman for his valuable advice and guidance in the early stage of developing my research skills.

My family and friends for their continuous support and encouragement without which I could not have made it.

TABLE OF CONTENTS

LIST OF TABLES.....	ix
LIST OF FIGURES.....	xi
ABBREVIATIONS.....	xiii
Chapter	
I. INTRODUCTION.....	1
Introduction to Frequent Subgraph Mining	3
Fast frequent subgraph mining (FFSM) algorithm	4
Overview of Chapter 2.....	7
Overview of Chapter 3.....	7
Overview of Chapter 4.....	8
II. DEVELOPMENT OF FRAGMENT-BASED CHEMICAL	
DESCRIPTORS	10
Introduction.....	10
Computational Methods.....	13
Application of FFSM to chemical datasets to generate chemical fragment	
descriptors.	13
Removing redundant chemical fragments	17
Experimental datasets	20
QSAR model development and validation methods.....	20
Results and discussions.....	28

Generating the fragment-based chemical descriptors	28
Building kNN-classification models	32
Comparison with other molecular descriptors	35
Conclusions	38
 III. IDENTIFYING TWO-DIMENSIONAL (TOPOLOGICAL)	
PHARMACOPHORES/TOXICOPHORES	39
Introduction	39
Computational Methods	41
Application of FFMS to chemical datasets to generate closed subgraphs and use them as chemical fragments	41
Classification based on association (CBA) method	43
Experimental datasets	47
Method validation	48
Results and Discussion	51
Examples of associated fragments for the Ames Mutagenicity dataset	55
Examples of associated fragments for the MRTD dataset	64
Examples of associated fragments for the PGP dataset	77
Weaknesses and strengths of the descriptors and methodology	86
Conclusions	86
 IV. DEVELOPMENT OF POSE-SCORING FUNCTION FOR PROTEIN- LIGAND BINDING BASED ON FREQUENT PATTERNS OF INTER-ATOMIC INTERACTIONS AT THEIR INTERFACES	
Introduction	88

Computational Methods.....	92
Dataset of Protein-Ligand Complexes.....	92
Graph Representation of the Protein-Ligand Interface.....	93
Application of Frequent Subgraph Mining to Identifying Frequent Atomic Interaction Patterns at the Protein-Ligand Interface.	96
Deriving the Scoring Function Using Frequent Protein-Ligand Interaction Patterns.....	99
Validation of the Scoring Function.....	100
Results and Discussions.....	104
Identification of “classical” interaction patterns in the internal training set and external test set scoring.	104
Applying more stringent external test: switching the internal training and external test sets.	113
Conclusions.....	117
Acknowledgments.....	119
V. SUMMARY AND FUTURE DIRECTIONS.....	120
Summary and Future Directions of Chapter 2	122
Summary and Future Directions of Chapter 3	124
Summary and Future Directions of Chapter 4	126
REFERENCES.....	131

LIST OF TABLES

Table

3.1	Internal dataset for Salmonella, has a prediction a total error of 19.6% using the fingerprints descriptors.....	44
3.2	External validation for Salmonella, has a total error of 28.2% using fingerprints descriptors.	44
3.3	Internal dataset for Salmonella, has a prediction a total error of 14.6% using the fragment-based chemical descriptors.	46
3.4	External validation for Salmonella, has a total error of 22.0% using fragment-based chemical descriptors.....	46
3.5	Example of rules used in the classifier built by CBA.	48
3.6	Internal dataset for MRTD, has a prediction a total error of 11.7% using the fingerprints descriptors.	54
3.7	External validation for MRTD, has a total error of 28.1% using fingerprints descriptors.	54
3.8	Internal dataset for MRTD, has a prediction a total error of 8.2% using the fragment-based chemical descriptors.....	55
3.9	External validation for MRTD, has a total error of 26.2% using fragment-based chemical descriptors.....	55
3.10	Example of rules used in the classifier built by CBA.	57
3.11	Internal dataset for PGP, has a prediction a total error of 1.5% using the fingerprints descriptors.	65
3.12	External validation for PGP, has a total error of 30.2% using fingerprints descriptors.	65
3.13	Internal dataset for PGP, has a prediction a total error of 3.8% using the fragment-based chemical descriptors.....	66
3.14	External validation for PGP, has a total error of 23.8% using fragment-based chemical descriptors.	66

3.15 Internal dataset for PGP, has a prediction a total error of 3.0% using the fragment-based chemical descriptors derived from the whole PGP dataset.	68
3.16 External validation for PGP, has a total error of 17.7% using fragment-based chemical descriptors derived from the whole dataset.	68
3.17 Example of rules used in the classifier built by CBA.	70
3.18 Internal dataset for PGP, has a total error of 13.6% using the fragment-based chemical descriptors.	75
3.19 External validation for PGP, has a total error of 27.0% using fragment-based chemical descriptors and simple rules built by CBA.	75
3.20 External validation for PGP, has a total error of 23.8% using fragment-based chemical descriptors and closed rules in place of the simple rules built by CBA.	75
3.21 Screening Maybridge database seeded with the external dataset of PGP gave a total error of 14.3% using fragment-based chemical descriptors and simple rules built by CBA.	77
3.22 Screening Maybridge database seeded with the external dataset of PGP gave a total error of 12.5% using fragment-based chemical descriptors and closed rules in place of the simple rules built by CBA.	77

LIST OF FIGURES

Figure

1.1	<u>Top</u> : Examples of three labeled graphs (referred to as a graph database). The labels of the nodes are specified within the circle and the labels of the edges are specified along the edge. The mapping $q_1 \rightarrow p_2, q_2 \rightarrow p_1, q_3 \rightarrow p_3$ represents an induced subgraph isomorphism from graph Q to P. <u>Bottom</u> : All the frequent induced subgraphs with support $\geq 2/3$ for the graph database.....	6
2.1	Conversion of each molecule in the dataset into undirected, labeled graph.....	14
2.2	Using FFSM to find common subgraphs in at least a subset of molecules of size 2 out of 3 molecules ($\sigma = 2/3$).....	15
2.3	Matrix of counts (number of occurrences) for each subgraph (chemical fragments) in each molecule in the dataset.	16
2.4	Matrix of counts (number of occurrences) for closed subgraphs (chemical fragments) in each molecule in the dataset.	19
2.5	kNN QSAR modeling approach (a) and predictive QSAR modeling workflow (b).	27
2.6	Number of subgraphs as a function of the support value (σ).	29
2.7	Distribution of the size (number of nodes) of the subgraphs using support value $\sigma = 1\%$ before and after removing redundant subgraphs.	31
2.8	Model fitness as a function of support σ (%) for PGP.....	33
2.9	Model fitness as a function of support σ (%) for MRTD.	33
2.10	Model fitness as a function of support σ (%) for Ames genotoxicity.....	34
2.11	External sets prediction accuracies for each dataset.	34
2.12	External sets prediction accuracies for each dataset using fragment-based and MolConnZ descriptors.	37
3.1	Matrix of 1's and 0's for the occurrence or not, respectively, of the closed subgraphs (chemical fragments) in each molecule in the dataset.	42

3.2	Matrix of 1's and 0's for the occurrence or not, respectively, of the closed subgraphs (chemical fragments) in each molecule in the dataset, as well as the class label for the molecule.....	45
3.3	Work flow for the division of the datasets; identifying chemical fragments; generation of class association rules; and external validation.	50
4.1	Inter-atomic interactions at the protein-ligand interface, within a distance cutoff 3.15 Å. Protein is “adenosine deaminase”, and ligand is “PRH“ in the “1a4m” PDB complex.....	95
4.2	Example of 4 different geometries for an interaction pattern between protein and ligand atoms as well as water molecules.	98
4.3	Work flow for the validation of the method.	103
4.4	Comparison between scoring functions using 231 (core set) as external testing set, and the remaining 860 as internal training set.	106
4.5	The number of protein complexes in the external test set as a function of the rank order of the native pose for these complexes.....	108
4.6	The number of protein complexes in the external test set as a function of the rank order of the pose with the smallest RMSD for these complexes.	110
4.7	Rank order as function of RMSD for the protein-ligand complex ”1nc3”.	112
4.8	Comparison between scoring functions using 231 (core set) as internal training set, and the remaining 860 as external test set.....	114
4.9	Comparison between scoring functions using 860 complexes as internal training set, and the remaining 231 as external test set, where sets have different families.	116

ABBREVIATIONS

2D	Two dimensional.
3D	Three dimensional
AUC	Area under the curve.
CADD	Computer-assisted drug design.
CARs	Class Association Rules.
CBA	Classification based on association.
CCR	Correct Classification Rate.
FFSM	Fast frequent subgraph mining.
FP-tree	Frequent patterns tree.
ILP	Inductive logic programming.
ISIDA	In silico design and data analysis.
kNN	k-Nearest Neighbors.
LOO	Leave one out.
MoAD	Mother of All Databases.
MRTD	Maximum Recommended Therapeutic Dose.
PDB	Protein Data Bank.
PGP	P-Glycoprotein.
QSAR	Quantitative structure-activity relationship.
QSPR	Quantitative structure-property relationship.
RMSD	Root mean square deviation.
SE	Sphere exclusion.

CHAPTER 1

INTRODUCTION

Computer-assisted drug design (CADD) techniques have been used successfully to improve the efficiency of the drug discovery process. The combination of computational chemistry concepts, robust software, and high-end computer hardware are used to assist the medicinal chemists identifying or designing ligands that are more likely to interact with the receptor of interest. CADD methods can be categorized based on the availability of the three-dimensional (3D) structure of the target protein. Ligand-based drug design methods are used if the structure of the target protein is not known. A commonly used method is the Quantitative Structure-Activity/Property Relationship (QSAR/QSPR) approach (Martin, Y., 1981). It generates molecular descriptors for all ligands with known target property (i.e. biological activity, toxicity) and uses them in combination with multivariate statistical modeling techniques to arrive at predictive activity or property models. The success of this approach relies on the robustness of the molecular descriptors used, as well as the strength of the statistical technique used to build the predictive models. Many currently available molecular descriptors either lack the mechanistic interpretability or are limited by the pre-defined set of chemical fragments that are used in *chemotyping* of any dataset of interest. The current limitations of molecular descriptors used in modern QSAR and cheminformatics research underline the significance of this project that is focused on developing dataset-specific descriptors based on the frequent sub-structures in the dataset. These frequent sub-structures will be identified using the graph representation of molecules

and the sub-graph mining approach, as we shall explain later. The medicinal chemist can easily interpret these descriptors. In addition, new important fragments that might have not been defined *a priori* can be discovered. The research question that needs to be answered in the course of this project is whether these descriptors can indeed give a better predictive QSAR model as compared to those generated with current descriptors.

A popular ligand-based drug design method is the so-called Active Analog Approach (Sheridan, R., Rusinko, A., Nilakantan, R., Venkataraghavan, R., 1989). It is used to explore active compounds that bind to same target protein in order to identify “pharmacophoric” groups responsible for the specific activity; these groups are subsequently used to screen chemical databases for new leads. In this project, we will answer the question whether the frequent sub-structures can be used as novel means to identify the pharmacophoric groups and then examine their ability to identify new leads in the context of the Active Analog Approach. The significance of this particular study rests on the fast identification of the pharmacophoric groups for database mining. The advantage of our proposed approach is that it does not rely on 3D conformational search of the structures and therefore it is highly efficient computationally.

If the three-dimensional structure of the target protein is available then structure-based drug design methods are used. The most common one and a widely used method is the computational “docking”. Here, a database of compounds is screened to identify compounds that can fit into the active site of the target protein. This approach has been widely used in hit identification and lead optimization. However, there remain significant challenges in the application of this approach, in particular in relation to current scoring schemes. Even when binding conformations are correctly predicted, the calculations

ultimately do not succeed if they do not differentiate correct poses from incorrect ones, and if “true” ligands can not be identified. So, the design of reliable scoring functions and protocols is of fundamental significance.

With the exponential increase in the number of protein-ligand crystal structures in the protein databank (PDB), researchers are more interested in exploring the information that can be gathered from these structures. This project will try to answer the question whether the frequent patterns of inter-atomic interactions at the protein-ligand interface can be used in forming new more precise scoring functions and docking schemes as compared to current methods. The study can be highly significant and of interest to many researchers in that field. The study will also bring insights to the structure based *de novo* design of ligands complementary to the active sites.

INTRODUCTION TO FREQUENT SUBGRAPH MINING

Frequent subgraph mining is a powerful tool that can be used to extract information from different types of databases (Huan, J., Prins, J., and Wang, W., 2003). It is becoming more important in many application areas including cheminformatics, bioinformatics, web mining, video indexing, and sociology, especially with the rapid growth of data available. The the goal is to discover interesting patterns in large collections of data where interestingness is related to the frequency of occurrence. The process starts by the graphical representation of the data; i.e. elements are represented by labeled nodes, and relationships between these elements are represented by labeled edges, followed by frequent subgraph mining to identify the frequent patterns. These patterns can then be used to make class predictions for unseen cases or discovering new classes.

Given a set S of graphs, frequent subgraphs that occur in a fraction (support value) of all graphs S are found. For any frequent subgraph mining algorithm, there are two computationally challenging problems: First, subgraph isomorphism, which is determining whether a given graph is a subgraph of another graph. Second, enumerating all frequent subgraphs efficiently (Huan et al, 2003). There are several efficient subgraph mining algorithms that have been presented in a recent review by Huan et al, 2003. For our study, we have been using Fast Frequent Subgraph Mining (FFSM) algorithm which will be described in the following section. The FFSM algorithm will be applied to mine datasets of small molecules to find frequent patterns (chemical fragments) that can be used for classification purposes as we will see in Chapter 2 and Chapter 3. In addition, it will be applied to find frequent patterns of interactions at the protein-ligand complexes as we will see in Chapter 4.

Fast frequent subgraph mining (FFSM) algorithm

The FFSM algorithm was developed by our collaborators in the Computer Science Department as a general highly efficient tool to find common frequent subgraphs in a family of labeled unidirectional graphs. A *labeled graph* G is defined as a five element tuple $G = \{V, E, \Sigma_v, \Sigma_E, \delta\}$ where V is the set of nodes of G and $E \subseteq V \times V$ is the set of undirected edges of G . Σ_v and Σ_E are a set of labels and the labeling function $\delta: V \rightarrow \Sigma_v \cup E \rightarrow \Sigma_E$ maps nodes and edges in G to their labels. The same label may appear on multiple nodes or on multiple edges, but we require that the set of edge labels and the set of node labels are disjoint.

A labeled graph $G = (V, E, \Sigma_v, \Sigma_E, \delta)$ is *isomorphic* to another graph $G' = (V', E', \Sigma_v', \Sigma_E', \delta')$ if and only if there is a bijection $f: V \rightarrow V'$ such that:

$$\forall u \in V, \delta(u) = \delta(f(u)), \text{ and}$$

$$\forall u, v \in V, ((u, v) \in E \Leftrightarrow (f(u), f(v)) \in E') \wedge \delta(u, v) = \delta'(f(u), f(v))).$$

The bijection f denotes an *isomorphism* between G and G' .

A labeled graph $G = (V, E, \Sigma_v, \Sigma_E, \delta)$ is an *induced subgraph* of graph $G' = (V', E', \Sigma_v', \Sigma_E', \delta')$ if and only if G is subgraph isomorphic to G' and preserves all G' edges connecting nodes in G .

A labeled graph G is *induced subgraph isomorphic* to a labeled graph G' , denoted by $G \subseteq G'$, if and only if there exists an induced subgraph G'' of G' such that G is isomorphic to G'' . Examples of labeled graphs, induced subgraph isomorphism, and frequent induced subgraphs are presented in **Figure 1.1**.

Given a set of graphs GD (referred to as a *graph database*, e.g., a database of molecular graphs), the *support* of a graph G , denoted by sup_G is defined as the fraction of graphs in GD which embeds the subgraph G . Given a threshold σ ($0 < \sigma \leq 1$) (denoted as *minSupport*), we define G to be frequent, iff sup_G is at least σ . All the frequent induced subgraphs in the graph database GD presented in **Figure 1.1 (Top)** (with *minSupport* 2/3) are presented in **Figure 1.1 (Bottom)**. Further details of the development and implementation of the FFSM algorithm are described elsewhere (Huan et al., 2005). The FFSM executable (version 1.0) is available for download at <http://www.cs.unc.edu/~huan/FFSM.html>.

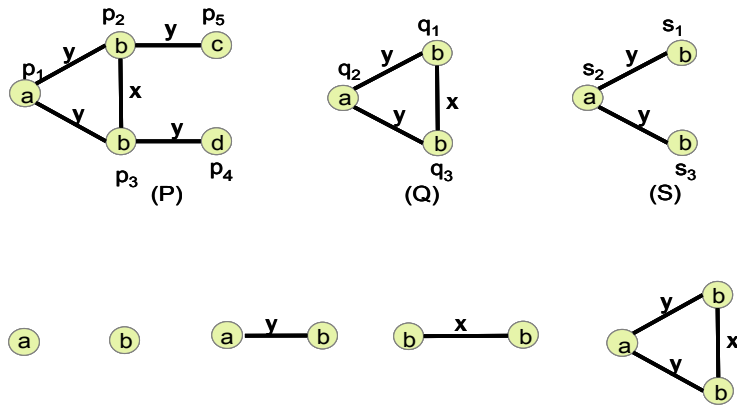


Figure 1.1. Top: Examples of three labeled graphs (referred to as a graph database). The labels of the nodes are specified within the circle and the labels of the edges are specified along the edge. The mapping $q_1 \rightarrow p_2$, $q_2 \rightarrow p_1$, $q_3 \rightarrow p_3$ represents an induced subgraph isomorphism from graph Q to P. Bottom: All the frequent induced subgraphs with support $\geq 2/3$ for the graph database.

OVERVIEW OF CHAPTER 2

In this chapter, we present a novel approach to generating fragment-based molecular descriptors. Using labeled chemical graph representation of molecules, Fast Frequent Subgraph Mining (FFSM) method developed in this group is used to find chemical fragments that occur in at least a subset of molecules in a dataset. The counts of frequent fragments have been used as descriptors in variable selection k Nearest Neighbor (kNN) QSAR modeling. This approach was applied to Maximum Recommended Therapeutic Dose (MRTD), Salmonella Mutagenicity (Ames Genotoxicity), and P-Glycoprotein (PGP) datasets. We followed established protocols for model validation, i.e., randomization of target property and splitting the datasets into training, test, and validation sets. Highly predictive models have been generated with the accuracies for the training and test sets exceeding 0.75, and the accuracy for the external validation sets exceeding 0.72. The accuracy results were comparable to commonly used molecular descriptors and in some cases was better. In addition, fragment-based descriptors implicated in validated models can afford mechanistic interpretation of the results in terms of essential pharmacophoric or toxicophoric elements responsible for the compounds' target property. For interpretation purposes, another classification method will be used as we will see in Chapter 3.

OVERVIEW OF CHAPTER 3

In this chapter we present a novel approach to identify 2D pharmacophores/toxicophores based on frequent subgraph mining. Using labeled chemical graph representation of molecules, Fast Frequent Subgraph Mining (FFSM) method developed in this group is used to find chemical fragments that occur in at least a subset of

molecules in a dataset. These chemical fragments are used as binary descriptors for the dataset. Then, Classification-Based Association (CBA) algorithm is used to identify associated chemical fragments responsible for the activity as well as the toxicity (mutagenicity) for datasets of compounds and provide interpretation for these results. The method is validated for its ability to predict the activity/toxicity of an external dataset. This approach was applied to a dataset of P-Glycoprotein substrates (PGP), Maximum recommended therapeutic dose dataset (MRTD), and to a dataset of mutagenic compounds (Salmonella Ames Mutagenicity dataset). The prediction ability of the method using the chemical fragments identified was compared to that when using Fingerprints descriptors. The results show a significant improvement in the predictive ability when using the chemical fragments identified in this method over the Fingerprints descriptors.

OVERVIEW OF CHAPTER 4

Developing a scoring function that identifies the correct docking pose is very important in understanding the binding mode of a ligand to its receptor, and consequently, in the design of new lead compounds. In this chapter, we present a study for a novel knowledge-based scoring function that has been developed based on the frequent geometric and chemical patterns of inter-atomic interactions at the interface of a representative dataset of x-ray characterized protein-ligand complexes. The approach includes the following steps. First, the protein-ligand interfaces of each complex in the internal training set are represented by labeled chemical graphs where nodes are atoms and edges connect protein and ligand atoms within certain distance of each other. Second, subgraph mining techniques are used to find frequent subgraphs that occur in no less than a certain percentage of the complexes in the internal training set, and these frequent subgraphs

identify the patterns that are used in the scoring function. Thus, the external test protein-ligand complexes are scored based on the similarity between interaction patterns identified at the protein-ligand interface of an external test protein-ligand pair to those found frequently in the internal training set of x-ray characterized complexes. The scoring function has been tested for its ability to accurately recognize the native pose of a ligand in the X-ray crystal structure of the protein-ligand complexes vs. non-native poses produced by computational docking. We have demonstrated that this novel scoring function affords higher accuracy of scoring than five commonly used scoring functions and their consensus provided by commercial docking software.

CHAPTER 2

DEVELOPMENT OF FRAGMENT-BASED CHEMICAL DESCRIPTORS

INTRODUCTION

QSAR modeling is fundamentally based on the similarity principle implying that similar compounds have similar biological properties. Consequently one can predict the biological target property of a molecule from that of chemically similar compounds for which the property is known. To build quantitative predictive models a similarity metric is required; therefore a unit of measurement such as molecular descriptors needs to be identified. Once the descriptors are defined, QSAR techniques can be used to relate the chemical structure of a molecule to its target property.

There are many types of molecular descriptors that can be used for QSAR studies. While some are based on describing molecules at atomic level (e.g. electro-negativity, partial charges, hydrogen bond acceptor and donor ability, etc.), others are based on describing them at the molecular level (e.g. molecular weight, logP, surface area, etc.). While three-dimensional (3D) descriptors based on the conformational structure of the molecule are capable of distinguishing stereo-isomers and changes in structural conformations, 2D descriptors offer the advantage of conformation-independence and much speedier computation. In this study, we present a novel approach to generate fragment-based molecular descriptors. Unlike molecular descriptors based on physicochemical properties and distances of atoms in the molecule, fragment-based

descriptors could potentially provide a mechanistic explanation of the dependence of the target property on molecular structure. Such explanation especially with respect to the differences between active and inactive molecules could provide useful guidance to medicinal chemists with respect to rational design of new biologically active chemical entities.

Fragment-based descriptors have been used in QSAR modeling. Popular examples include fingerprints (e.g., Daylight), atom pairs, and ISIDA. A common trait to all these methods is that chemical fragments are identified a priori; thus frequently the total number of such descriptors generated for a molecular dataset is exceedingly large (e.g., hundreds or thousands fingerprints are generated typically) and/or fragment descriptors are generic. This makes it difficult to build robust and statistically predictive QSAR models that uniquely describe the relationship between structure and activity of specific datasets such that the derived QSAR models could successfully identify novel unique computational hits.

In this study we propose a novel approach to fragment descriptor generation where unique chemical fragments are identified based on the dataset of interest. To this end, we use a labeled chemical graph representation of molecules and employ Fast Frequent Subgraph Mining (FFSM) method developed in our group. Chemical graphs have been used very successfully and for a long time in cheminformatics and QSAR studies giving rise to popular molecular descriptors such as connectivity indices. Algorithms for finding maximum common subgraphs in chemical structures have been developed by other groups (Willett et al., 2002; Bradshaw et al., 2003) and used to study similarity/diversity of chemical structures.

Finding patterns from graphs has long been an interesting topic in the data mining/machine learning community. For instance, Inductive Logic Programming (ILP) has been widely used to find patterns from graph dataset (Dehaspe, Toivonen, and King, 1999). However, ILP is not designed for large databases. Other pioneer methods focused on approximation techniques such as SUBDUE (Holder, Cook, and Djoko, 1994) or on heuristics such as the greed based algorithm (Yoshida and Motoda, 1995). Several algorithms have been recently developed by the data mining community to solve the so-called frequent subgraph mining problem which reports all frequent subgraphs of a group of general graphs (Huan, Prins, and Wang, 2003; Huan et al., 2004; Kuramochi and Karypis, 2001; Yan and Han, 2002). These techniques have been successfully applied in cheminformatics where compounds are modeled by undirected graphs. Recurring substructures in a group of chemicals with similar activity are identified by finding frequent subgraphs in their related graphical representations. The recurring substructures can implicate chemical features responsible for compounds' biological activities (Deshpande, Kuramochi, Wale, and Karypis, 2005).

Our fragment-based descriptors are derived based on frequent common substructures that are found in at least a subset of molecules (this fraction is defined as a *support value*) in the dataset. Once these frequent substructures are identified, the counts of each substructure in each molecule in the dataset is calculated; thus each frequent common substructure serves as a chemical descriptor type and the frequency becomes a descriptor value. This representation affords the application of conventional QSAR modeling techniques to any chemical dataset with measured biological activity leading to a novel fragment descriptor based QSAR modeling approach. The objectives of this study include:

(a) provide a detailed description of the frequent subgraph mining approach as applied towards developing the fragment-based descriptors; (b) validate these descriptors by developing predictive QSAR models (using k-Nearest Neighbor (kNN) QSAR techniques) for several experimental datasets, and (c) finally, discuss the applications of these descriptors in the QSAR analysis for drug design and development.

COMPUTATIONAL METHODS

Application of FFSM to chemical datasets to generate chemical fragment descriptors.

The molecules are described in the SYBYL MOL2 file format, which considers 33 atom types and 5 bond types. Chemical structures are then represented as hydrogen suppressed graphs, where atoms are considered as labeled nodes and bonds are labeled edges. Then, the FFSM algorithm described earlier in Chapter 1 is used to find the frequent (chemical) subgraphs for a given a support value (σ), which is one of the model variables defined by the user. **Figure 2.1** shows an example for representing three molecules comprising a small dataset as labeled unidirectional graphs and **Figure 2.2** presents a simple example of the output generated as a result of applying FFSM to this small dataset with the support value of 66.7% (i.e. $\sigma = 2/3$).

To continue with this example, **Figure 2.3** shows a matrix of chemical fragment descriptors where all frequent subgraphs with the support of $\sigma = 2/3$ serve as descriptors and each descriptor's count represent the descriptor's value for each molecule.

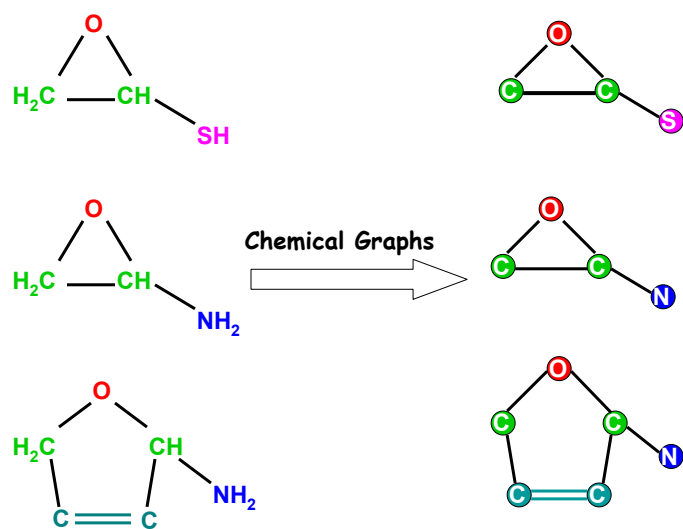


Figure 2.1 Conversion of each molecule in the dataset into undirected, labeled graph.

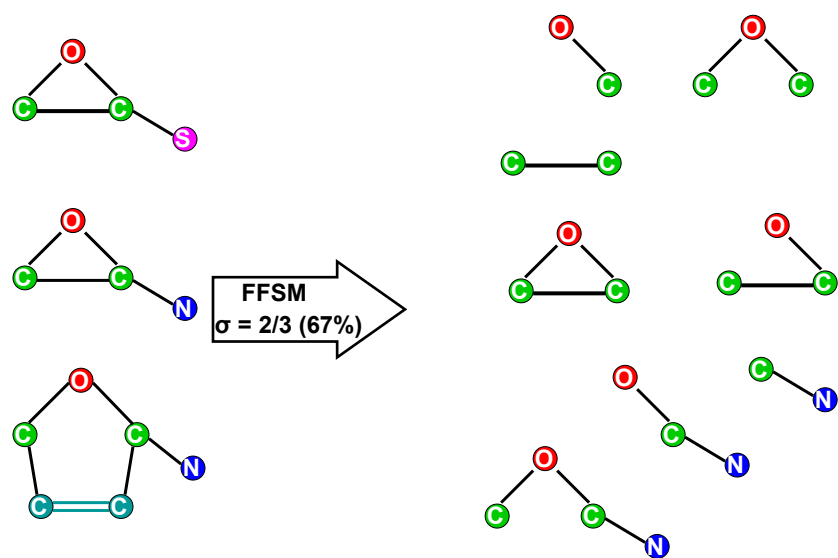


Figure 2.2 Using FFSM to find common subgraphs in at least a subset of molecules of size 2 out of 3 molecules ($\sigma = 2/3$).









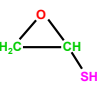
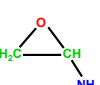
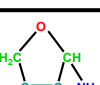
Fragment \ Dataset								
	1	1	1	1	3/2	0	0	0
	1	1	1	1	3/2	1	1	1
	0	0	0	1	3/2	1	1	1

Figure 2.3 Matrix of counts (number of occurrences) for each subgraph (chemical fragments) in each molecule in the dataset.

Removing redundant chemical fragments

The application of the FFSM algorithm to chemical datasets may result in the identification of redundant features. For example, if an aromatic group is found to be frequent, then all the sub-structures within such aromatic group will also be frequent in spite of having any of these sub-structures present independently in the molecular dataset. This problem of subgraph redundancy is well known in graph mining, and the resulting subgraphs after removing redundant ones are called *closed* subgraphs. A subgraph g is closed in a database if there exist no proper supergraph of g that has the same support as g (Yan, X., and Han, J., 2003). In our studies reported herein to eliminate the redundancy in the frequent subgraphs (chemical fragments) leaving only closed ones, the following criteria was used:

For each two frequent subgraphs SG_i and SG_j : If $(SG_i \subseteq SG_j)$ and $\text{support}(SG_i) = \text{support}(SG_j)$, then remove SG_i .

However, a subgraph SG_i that is embedded in SG_j (i.e. $SG_i \subseteq SG_j$) and has the same support value as SG_j will not be deleted if it also occurs by itself (not as part of the SG_j) in the graph database of molecules. This is important since it will retain subgraphs that can be useful.

In the sample dataset and its features (descriptors) shown in **Figures 2.1, 2.2, and 2.3**, we find that the first 3 subgraphs have the same support value of 3 out of 3 graph molecules in the dataset (i.e., $\sigma = 3/3$). Consequently, the first subgraph will stay while the second and the third ones will be eliminated. Similar considerations are applied to the next two subgraphs, i.e., fourth and fifth: the fourth stays and the fifth will be removed. Finally, the same analysis is applied to the last three subgraphs leading to the elimination of the last

two subgraph descriptors. Therefore, for our toy example we will end up with only three closed subgraphs that will be used as our unique descriptors (see **Figure 2.4**).

Removing redundant subgraphs (fragments) will reduce the number of subgraphs drastically and therefore make the subsequent processes faster and more efficient.



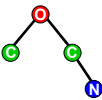
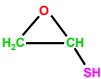
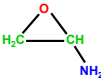
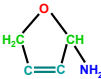
Fragment \ Dataset			
	1	1	0
	1	1	1
	0	1	1

Figure 2.4 Matrix of counts (number of occurrences) for closed subgraphs (chemical fragments) in each molecule in the dataset.

Experimental datasets

Three datasets were used in this study. The first one included 1217 drug-like molecules with the MRTD (Maximum Recommended Therapeutic Dose) as their target property. This dataset was recently analyzed by the FDA modeling group (Contrera et al., 2004). Following the approach described in the original publication all molecules were divided into two classes based on the MRTD cutoff value. This results in having 576 molecules with toxicological effect (adverse or undesirable pharmacological effect), and 641 molecules without toxicological effect. The second dataset is composed of 3434 drug-like molecules with the Salmonella mutagenic activity score as the target property. The score ranged from 10 to 80; molecules with no mutagenic activity have a score of 10, and the most mutagenic molecules have a score of 80. A cutoff value is used to divide the dataset into 2 classes: mutagenic and non-mutagenic, and thus resulting in 1615 mutagenic molecule versus 1819 non-mutagenic molecules. This dataset was described in a paper by Votano et al., 2004. The third dataset included 195 molecules shown to be substrates (108 molecules) or non-substrates (87 molecules) of the P-Glycoprotein Protein (PGP). This dataset was analyzed previously in our group using several modeling techniques and descriptor sets and its molecules were taken from a paper by Penzotti et al., 2002. Thus, all experimental datasets have a binary value as their target property.

QSAR model development and validation methods

Dataset Division into External, Training, and Test Sets. It is commonly accepted that the internal validation of QSAR models built from training sets is sufficient to confirm their predictive power (Benigni et al., 2000; Oloff et al., 2006; Trohalaki, Gifford, and Pachter, 2000; Zhang, Golbraikh, and Tropsha, 2006; Zhang et al., 2006). However,

previous studies in this as well as several other laboratories demonstrated that no correlation exists between leave-one-out (LOO) cross-validated R^2 (q^2) for the training set and the correlation coefficient R^2 between the predicted and observed activities for the test set (Golbraikh and Tropsha, 2002; Kubinyi, Hamprecht, and Mietzner, 1998). These findings indicated that in order to obtain QSAR models with high predictive ability, external validation was critical. Thus, each dataset of compounds was divided randomly into external and internal sets. Then, the internal set was divided into multiple chemically diverse training and test sets with a rational approach implemented in our group (Golbraikh and Tropsha, 2002) based on the Sphere Exclusion (SE) algorithm (Snarey et al., 1997). SE is a general procedure that is typically applied to molecules characterized by multiple descriptors of their chemical structures. The entire dataset can then be treated as a collection of points (each point corresponding to an individual compound) in the multidimensional descriptor space. The goal of the SE method is to divide a dataset into two subsets (training and test sets) using a diversity sampling procedure (Golbraikh and Tropsha, 2002).

The SE algorithm used in this study included the following steps. The algorithm starts with the calculation of the distance matrix D between points representing compounds in the multidimensional descriptor space. Let D_{\min} and D_{\max} be the minimum and maximum elements of D , respectively. N probe sphere radii are defined by the following formulas: $R_{\min}=R_1=D_{\min}$, $R_{\max}=R_N=D_{\max}/4$, $R_i=R_1+(i-1)*(R_N-R_1)/(N-1)$, where $i=2, \dots, N-1$. Each probe sphere radius corresponds to one division into the training and test sets. Once the sphere size is defined the subsequent calculations include the following steps: (i) Select randomly a point in the descriptor space. (ii) Include it in the training set. (iii) Construct a

probe sphere around this point. (iv) Select points from this sphere and include them alternatively into test and training sets. (v) Exclude all points within this sphere from further consideration. (vi) If no more compounds left, stop. Otherwise let m be the number of probe spheres constructed and n be the number of remaining points. Let d_{ij} ($i=1,\dots,m$; $j=1,\dots,n$) be the distances between the remaining points and probe sphere centers. Select a point corresponding to the lowest d_{ij} value and go to step (ii). The training sets were used to build models and the test sets were used for model validation.

Correct classification rate (CCR). Typically, CCR is defined as the ratio of compounds classified correctly to the total number of compounds. This definition of CCR has a major drawback, if the counts of compounds belonging to different classes are significantly different. Suppose there are two classes, class 1 contains 75 compounds and class 0 contains 23 compounds. Assume that some hypothetical "model" will assign all compounds to class 1. Then $CCR=0.76$, since $75/(75+23)=0.76$, i.e. we would believe that our "model" is very good contrary to the common sense.

To avoid artificial overrating of the classification model accuracy, in this study CCR was defined as follows. Let N be the total number of compounds in a dataset, and N_1 and N_0 be the number of compounds in class 1 and the number of compounds in class 0, respectively (i.e., $N_0+N_1=N$). Let T_1 and T_0 be the number of compounds predicted as class 1 and the number of compounds predicted as class 0, respectively. Then

$$CCR=0.5(T_1/N_1+T_0/N_0). \quad (1)$$

In this case, for the hypothetical example described above we obtain $CCR=0.5$, and our "model" assigning all compounds to class 1 does not seem to be more accurate than the random assignment of each molecule with probability 0.5 to a class 1 or 0.

kNN-Classification. The stochastic variable selection kNN classification method is based on the idea that assigning a compound to a class can be defined by the class membership of its nearest neighbors (in a multi-dimensional chemistry space) taking into account weighted similarities between a compound and its nearest neighbors as follows (see **Figure 2.5**). Let N be the number of compounds in a dataset. In the simplest case of binary classification, these compounds are distributed between classes a or b . Let n_a and n_b be the number of compounds in classes a and b , respectively, and m be the number of descriptors (composing the multi-dimensional chemistry space) selected by the variable selection kNN classification procedure. The Tanimoto coefficient can be used as a similarity measure between two classes as follows:

$$T(a,b) = \frac{\sum_{i=1}^m \bar{D}_i^a \bar{D}_i^b}{\sum_{i=1}^m (\bar{D}_i^a)^2 + \sum_{i=1}^m (\bar{D}_i^b)^2 - \sum_{i=1}^m \bar{D}_i^a \bar{D}_i^b}, \quad (2)$$

Where \bar{D}_i^a and \bar{D}_i^b are average values of descriptor i for classes a and b , respectively:

$$\bar{D}_i^a = \frac{\sum_{j=1}^{n_a} D_{ij}^a}{n_a} \text{ and } \bar{D}_i^b = \frac{\sum_{j=1}^{n_b} D_{ij}^b}{n_b},$$

Where D_{ij}^a is the descriptor value for compound j of class a . Evidently, $T(a, a) = 1$. Let k be the number of nearest neighbors of compound i . Weighted similarities between each compounds i and each class C (i.e., a , or b) are calculated as follows:

$$S_{i,C} = \sum_{p=1}^k \left[\frac{\exp(-\alpha d_{ip} / \sum_{p'=1}^k d_{ip'})}{\sum_{q=1}^k \left[\exp(-\alpha d_{iq} / \sum_{p'=1}^k d_{ip'}) \right]} T(a_p, C) \right], \quad (3)$$

Where a_p in $T(a_p, C)$ is the class of compound p , α is a parameter, which in this study was set to 1, and d_{ip} is the distance between compound i and its p -th nearest neighbor. In the leave-one-out cross-validation procedure, the similarity between compound i and each class C is calculated according to the following expression:

$$S'_{i,C} = \sum_{j=1}^k \left[\frac{\exp(-d_{ij})}{\sum_{j'=1}^k \exp(-d_{ij'})} S_{j,C} \right] \quad (4)$$

Compound i is assigned to the class which corresponds to the highest value of $S'_{i,C}$.

The CCR for the training set (CCR_{train}) is calculated using formula (1).

Applicability Domain of kNN QSAR Models. For assigning an external compound (which was not included in the training set) to a class, its representative point in the descriptor space must be not too far from its nearest neighbors of the training set. The similarity threshold was defined as the maximum squared distance between a compound, for which the prediction is made and its nearest neighbors of the training set. This squared distance can be defined as a sum of the average squared distance between nearest neighbors within the training set and a number Z of standard deviations of the squared distances from the average: $D^2_{\text{max}} = \langle D^2_{\text{near.neighb}} \rangle + Z\sigma_{\text{near.neighb}}$. The threshold is referred to here as Z -cutoff.

Classification accuracy of the model is estimated using the test set as follows. (1)

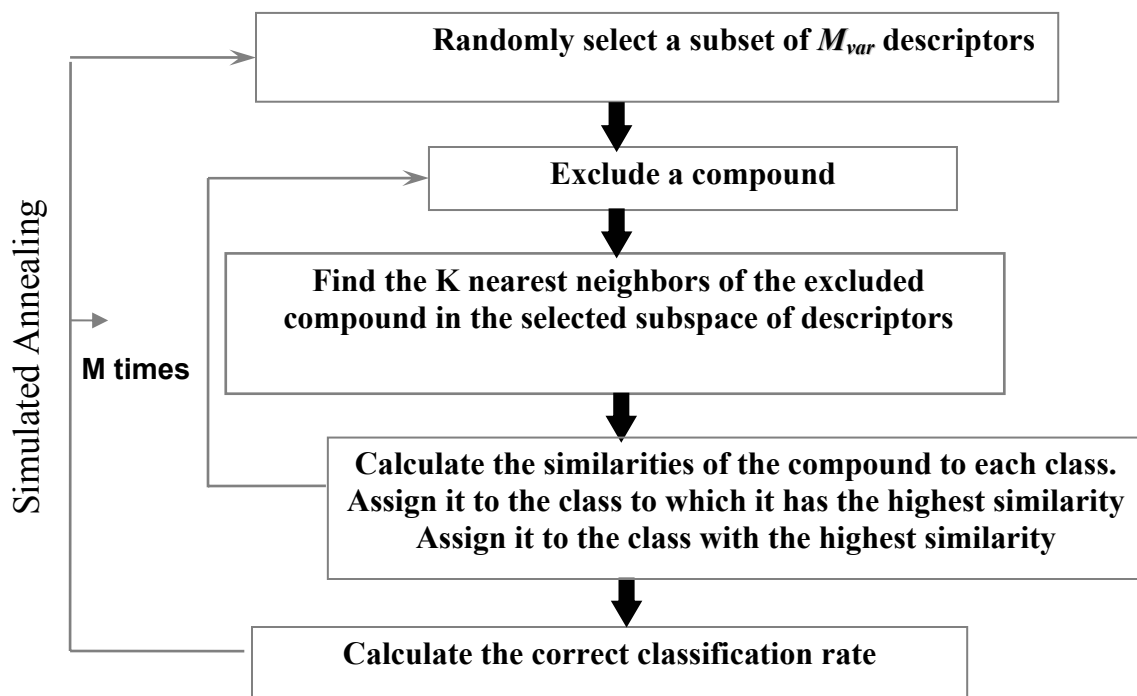
For each compound of the test set, k nearest neighbors from the training set are found. (2)

All compounds of the test set are selected, for which the distances to their nearest neighbors in the training set were within the defined Z-cutoff. (3) Similarity of each compound chosen in step (2) to each class is calculated using formula (4). The compound is assigned to a class, to which it has the highest similarity. (4) Classification accuracy of the model is characterized by the *CCR* for the test set (CCR_{test}) calculated with the formula (1). Maximum Z-cutoff value, for which reliable prediction of new compounds can be obtained, is a characteristic of the applicability domain of a QSAR model. In this study, Z-cutoff was set to 1.0.

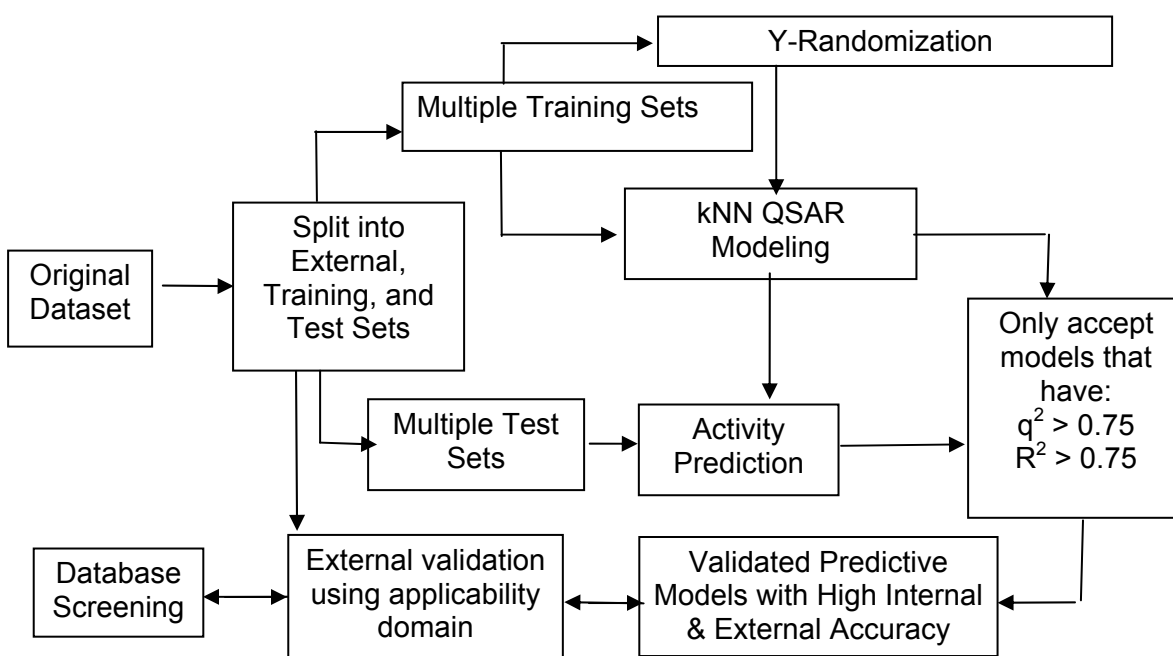
Classification *k*NN QSAR is a stochastic variable selection procedure based on the simulated annealing approach. The procedure is aimed at the development of a model with the highest fitness [CCR_{train}]. The procedure starts with the random selection of a predefined number of descriptors out of all descriptors. Compound excluded in LOO CV procedure is assigned to a class corresponding to a highest S_{iC} (see formula (3)), where *i* is the number of the excluded compound. After each run, cross-validated CCR_{train} is defined (see formula (1)) and a predefined number of descriptors are randomly changed (mutated). The new value of CCR_{train} is obtained using the modified subset of descriptors. If $CCR_{\text{train}}(\text{new}) > CCR_{\text{train}}(\text{old})$, the new subset of descriptors is accepted. If $CCR_{\text{train}}(\text{new}) \leq CCR_{\text{train}}(\text{old})$, the new subset of descriptors is accepted with probability $p = \exp(CCR_{\text{train}}(\text{new}) - CCR_{\text{train}}(\text{old}))/T$, and rejected with probability (1-*p*), where *T* is a simulated annealing parameter, “temperature”. During the process, *T* is decreasing until the predefined value. Thus, CCR_{train} is optimized. In the prediction process, the final set of descriptors selected is used, and formula (4) is applied to predict compounds in the test set.

This implementation is similar to that reported for the continuous kNN QSAR method developed in our laboratory earlier (Zheng, W. and Tropsha, A., 2000).

In all calculations reported in this work, the maximum number of nearest neighbors used (k) was 5, T_{\max} = 1000, T_{\min} = 10^{-6} , temperature decrement was 0.90, and the number of mutations was 2. For all descriptor types, the number of descriptors selected by the procedure was varied from 20 to 100 with step 5. For each number of descriptors selected, 10 models were built. Thus, the total number of models built for one division into training and test sets was 170. And since we have 50 pairs of training and test sets, the total number of models generated would be 8500.



(a)



(b)

Figure 2.5. kNN QSAR modeling approach (a) and predictive QSAR modeling workflow (b).

Comparison with other molecular descriptors. In order to demonstrate that the fragment-based chemical descriptors perform just as good as the other molecular descriptors, we compare it with the commonly used MolConnZ molecular descriptors (Kellogg, G., Kier, L., Gaillard, P., and Hall, L., 1996), and the fingerprints descriptors (as we will see in Chapter 2). Models were built for the same datasets using the same techniques and sets of parameters.

RESULTS AND DISCUSSIONS

There are many parameters that are playing a role in finding models with the best predictive ability. In this section we study these parameters and show how they affect the model development process.

Generating the fragment-based chemical descriptors

The support value (σ) determines the set of subgraphs generated as a result of using FFSM, these subgraphs will then form the fragment-based chemical descriptors. Obviously, the larger the value of the support, the smaller the number of subgraphs found. And as support value decreases, the number of subgraphs increases exponentially. **Figure 2.6** shows the number of subgraphs as a function of the support value for the Ames Genotoxicity dataset (3,434 drug-like molecules).

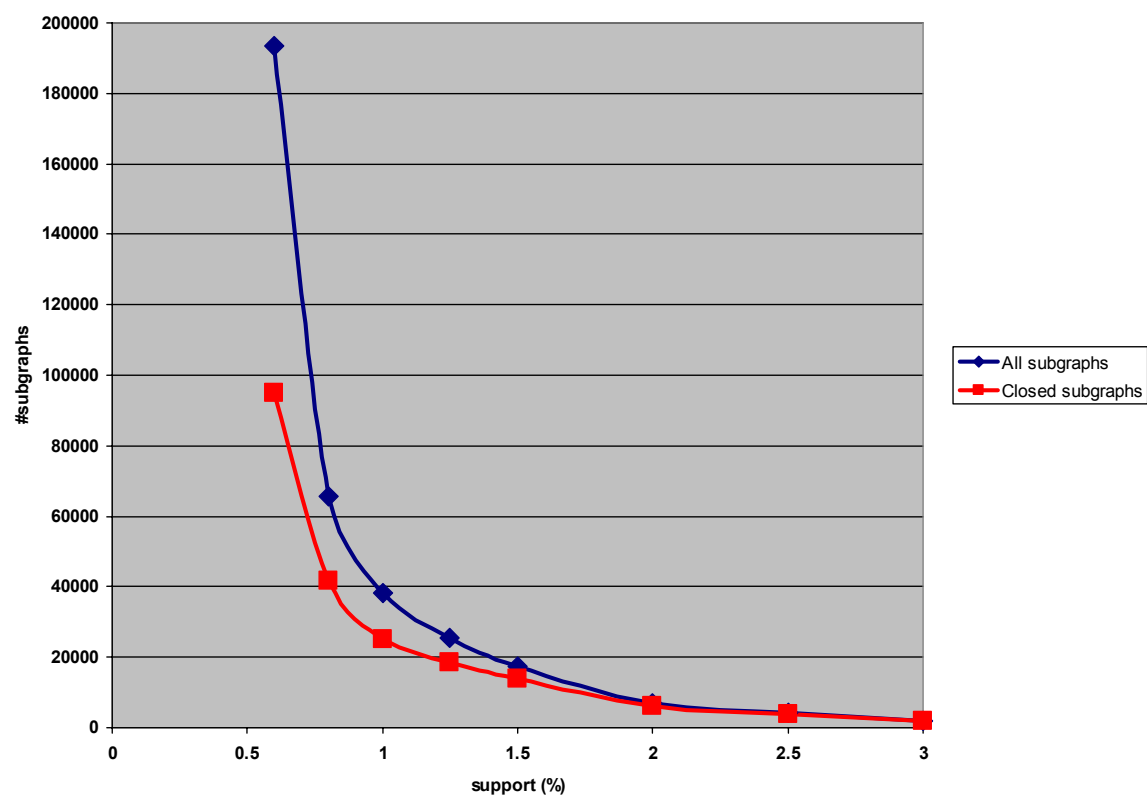


Figure 2.6. Number of subgraphs as a function of the support value (% σ).

The dark blue line shows the raw number of subgraphs generated by FFSM; the red line shows the number of subgraphs after removing redundant subgraphs leaving only the closed ones (cf. Methods). Notice how the number of subgraphs increases exponentially as the support value decreases, and notice the large drop in the number of closed subgraphs.

Figure 2.7 shows the size distribution of the subgraphs before and after removing correlated subgraphs for a single support value of 1.0 %. The size of a subgraph is simply the number of nodes in that subgraph.

Notice that the red curve is shifted to the right, implying that smaller subgraphs correlated with their parent subgraphs are removed leaving only closed subgraphs.

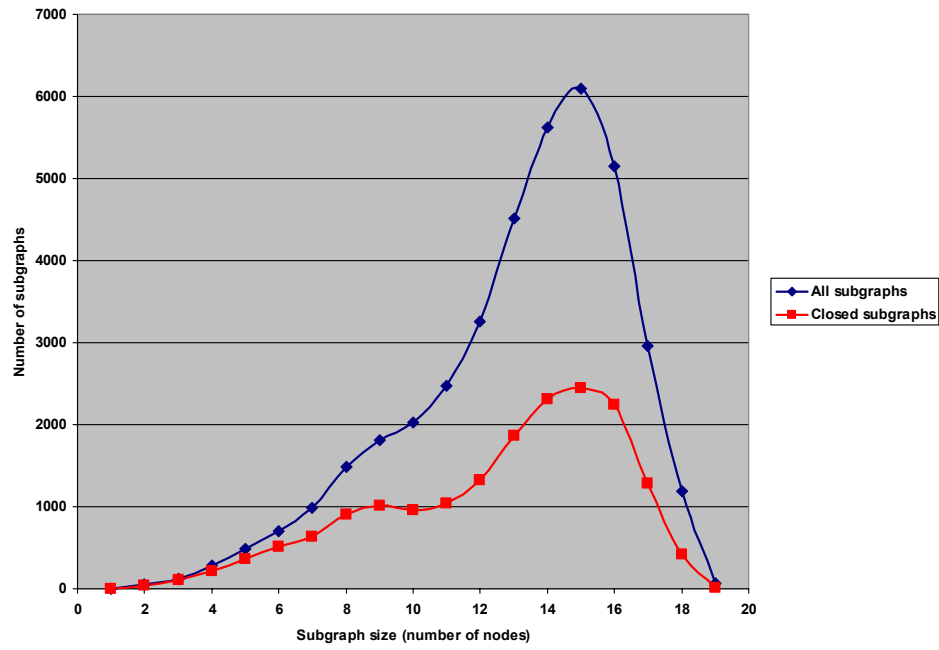


Figure 2.7. Distribution of the size (number of nodes) of the subgraphs using support value $\sigma = 1\%$ before and after removing redundant subgraphs.

Building kNN-classification models

Using the sets of closed subgraphs generated for a range of support values as descriptors classification kNN QSAR was used to build models for the three aforementioned datasets. **Figures 2.8-2.10** show the model fitness for each dataset as a function of the support value.

The analysis of the data presented in **Figures 2.8-2.10** leads to the following conclusions. The models serve to validate the fragment-based descriptors since high accuracy (>75%) was achieved using actual target property whereas models built with randomized target property gave accuracies <60% (keeping in mind that all three datasets have a binary type target property, meaning that the worst model you can get will have a 50% accuracy).

As the support value increases, the accuracies of models decrease. These observations can be explained easily because smaller number of generic common subgraphs is found and they are not useful in distinguishing between molecules' target property.

On the other hand, as the support value decreases, the number of subgraphs increases exponentially, and even though we have more subgraphs to use (i.e. higher chance of finding better models), we are limited by the ability of the simulated annealing-based kNN to find the right subgraphs among the huge number of subgraphs generated. Thus, model accuracies decrease again. That explains why best models are found when the number of subgraphs used is in the range of few hundreds marked by vertical dotted line in each figure. In theory, if kNN runs for some time that is long enough to find the right subgraphs, model accuracies should keep increasing.

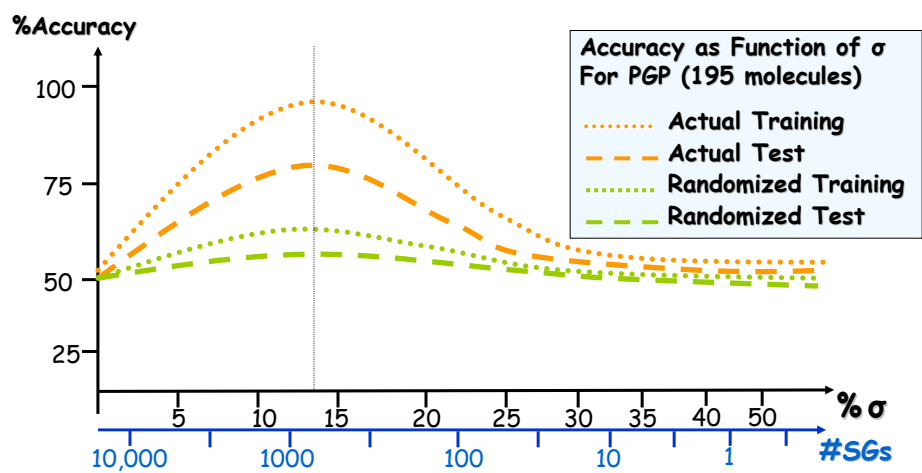


Figure 2.8. Model fitness as a function of support σ (%) for PGP.

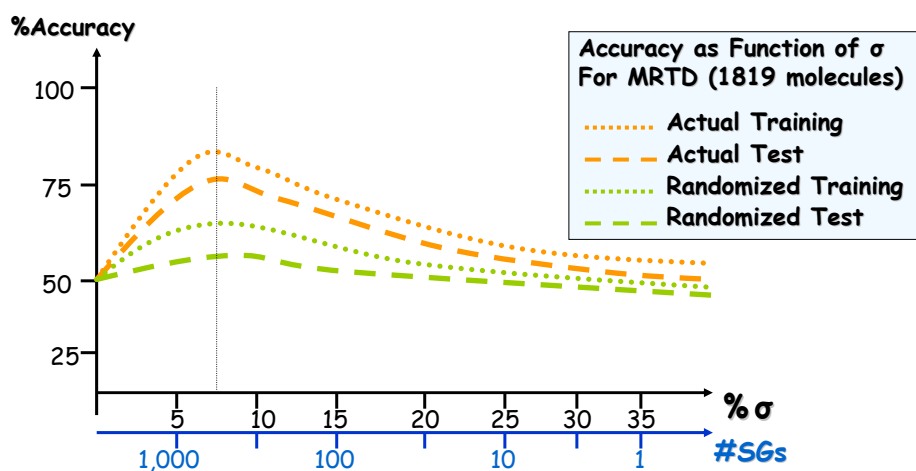


Figure 2.9. Model fitness as a function of support σ (%) for MRTD.

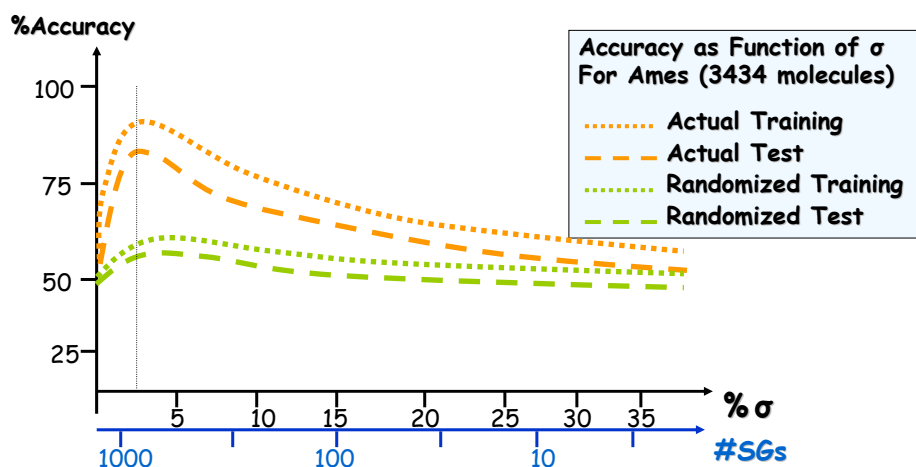


Figure 2.10. Model fitness as a function of support σ (%) for Ames genotoxicity.

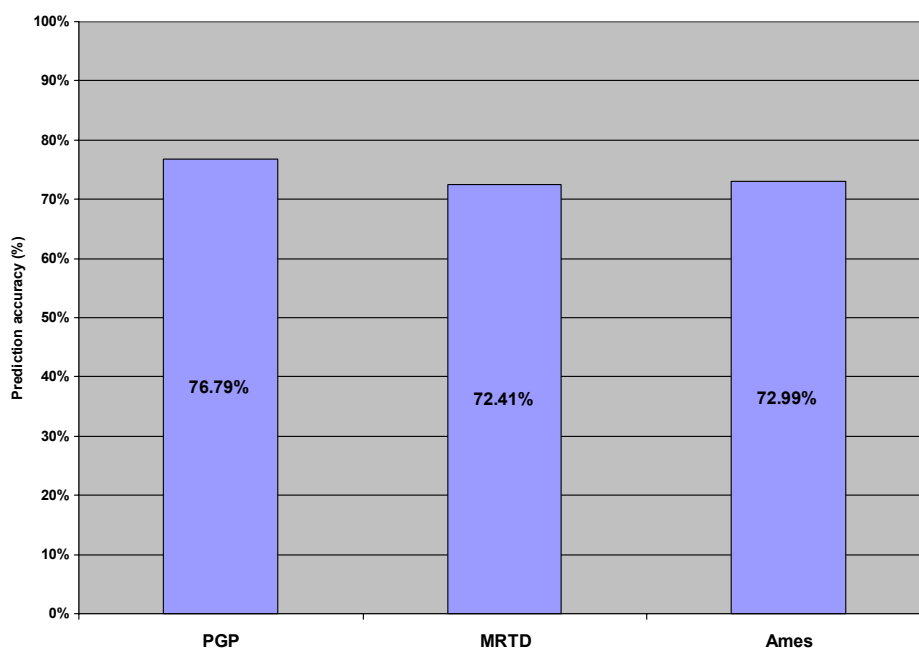


Figure 2.11. External sets prediction accuracies for each dataset.

Using the support values that give the best training and test sets' prediction accuracies for each of the three dataset, and using models with accuracies higher than 75% for both the training and test sets, an external validation prediction was performed. The accuracies for each of the datasets were above 72%, see **Figure 2.11**.

Comparison with other molecular descriptors

Finally, to compare between descriptors, subgraphs offers a direct interpretation of the features important in determining the target property that is easily understood and utilized by medicinal chemists, as we will address in Chapter 3. In addition, with variable selection kNN, branched features (and disconnected features) are taken care of. Also, since subgraph descriptors are not Boolean descriptors, but counts of subgraphs in the molecule, it should give a better description than structural alert descriptors and fingerprints that are based only on the presence or absence of such sub-structure. In addition to the fact that subgraph descriptors are dataset-derived and not predefined, this will open the door to finding new sub-structures that are not defined *a priori*.

In this section, we will show the results of comparing the fragment-based descriptors with one of the commonly used molecular descriptors, MolConnZ descriptors. Then in Chapter 3 of the dissertation, we will compare these descriptors with the fingerprints descriptors in terms of their ability to derive accurate predictive models.

Figure 2.12 shows the results of comparing fragment-based descriptors with MolConnZ descriptors. Using the same parameters' setting of kNN-classification modeling technique, the total number of models generated was 8500. For each of the three datasets, the models that have training and test sets' prediction accuracies higher than 75% were selected to be used to predict the external dataset. The accuracy of the prediction of the external

dataset will be compared to that obtained using the MolConnZ molecular descriptors, see **Figure 2.12**.

As **Figure 2.12** shows, models generated using fragment-based descriptors are comparable to those generated using MolconnZ descriptors and can perform better than the MolConnZ descriptors. In addition, the fragment-based descriptors provide a better interpretation to the medicinal chemist than MolConnZ descriptors do, see Chapter 3.

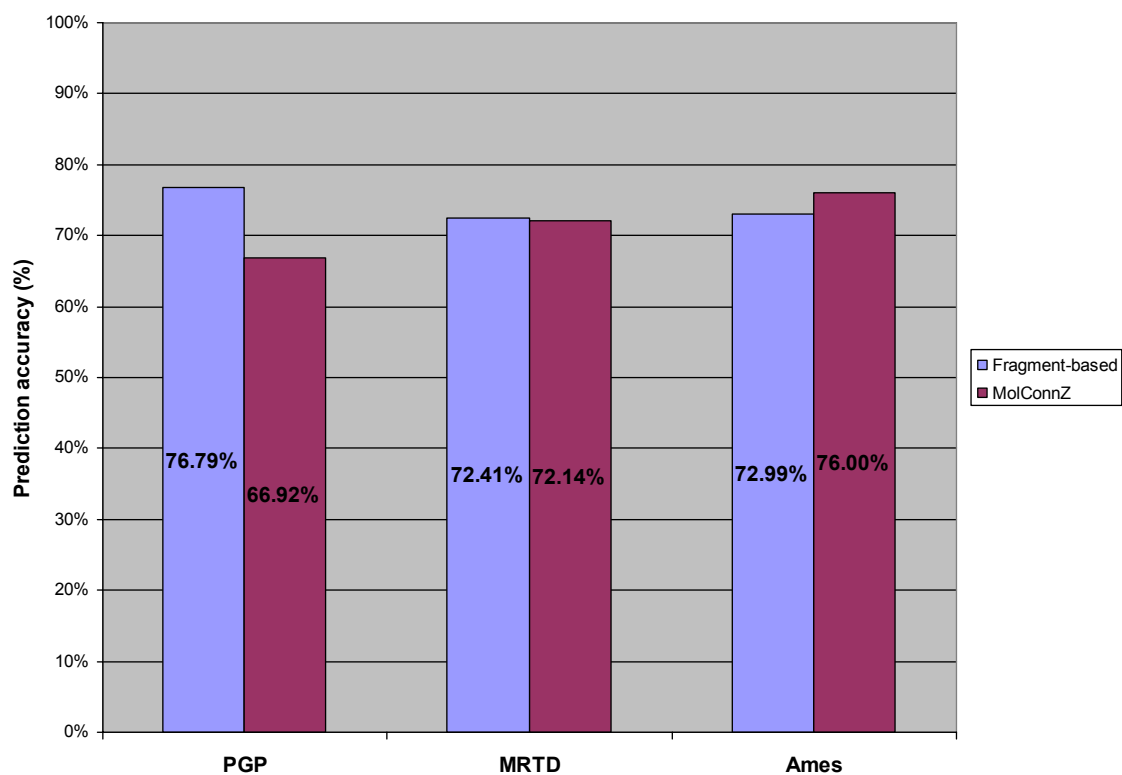


Figure 2.12. External sets prediction accuracies for each dataset using fragment-based and MolConnZ descriptors.

CONCLUSIONS

In this chapter, we present a novel approach to generating fragment-based molecular descriptors. Using labeled chemical graph representation of molecules, Fast Frequent Subgraph Mining (FFSM) method is used to find chemical fragments that occur in at least a subset of molecules in a dataset. The counts of each frequent fragment have been used as descriptors in variable selection k Nearest Neighbor (kNN) QSAR modeling. Highly predictive models have been generated for the datasets used in this study, and were comparable to one of the commonly used molecular descriptors. Frequent subgraphs implicated in validated models can afford mechanistic interpretation of the results that are easily understood by medicinal chemists in terms of essential pharmacophoric or toxicophoric elements responsible for the molecule activity, as we shall demonstrate in Chapter 3 of this dissertation using another classification method that can provide a better way of interpreting the selected descriptors than kNN does. Also, since subgraph descriptors are not Boolean descriptors, but counts of subgraphs in the molecule, it should give a better description than structural alert descriptors and fingerprints that are based only on the presence or absence of such sub-structure. Also, since these fragment-based descriptors are dataset-derived and not predefined, this will open the door to finding new sub-structures that are not defined *apriori*. In addition, they are dataset-specific, and therefore provide a better definition of the model applicability domain than apriori defined fragments.

CHAPTER 3

IDENTIFYING TWO-DIMENSIONAL (TOPOLOGICAL) PHARMACOPHORES/TOXICOPHORES

INTRODUCTION

As discussed earlier in Chapter 2, QSAR modeling is fundamentally based on the similarity principle implying that similar compounds have similar biological properties. Consequently one can predict the biological target property of a molecule from that of chemically similar compounds for which the property is known. To build quantitative predictive models a similarity metric is required; therefore a unit of measurement such as molecular descriptors needs to be identified. Once the descriptors are defined, QSAR techniques can be used to relate the chemical structure of a molecule to its target property.

Variety of molecular descriptors is available for QSAR studies. While some are based on describing molecules at atomic level (e.g. electro-negativity, partial charges, hydrogen bond acceptor and donor ability, etc.), others are based on describing them at the molecular level (e.g. molecular weight, logP, surface area, etc.). While three-dimensional (3D) descriptors based on the conformational structure of the molecule are capable of distinguishing stereo-isomers and changes in structural conformations, 2D descriptors offer the advantage of conformation-independence and much speedier computation. In this study, we present a novel approach to generate fragment-based molecular descriptors. Unlike molecular descriptors based on physicochemical properties and distances of atoms

in the molecule, fragment-based descriptors could potentially provide a mechanistic explanation of the dependence of the target property on molecular structure. Such explanation especially with respect to the differences between active and inactive molecules could provide useful guidance to medicinal chemists with respect to rational design of new biologically active chemical entities.

Having the ideal descriptors by itself is not enough to do QSAR predictions. The descriptors should be combined with the appropriate modeling technique to provide the best prediction. Based on the nature of the molecular descriptors, one modeling technique might perform better than another. In this study we will describe a unique methodology that is used with the fragment-based descriptors we identify.

As explained earlier in Chapter 2, our fragment-based descriptors are derived based on frequent common substructures that are found in at least a subset of molecules (this fraction is defined as a *support value*) in the dataset. Once these frequent substructures are identified, the occurrence of each substructure in each molecule in the dataset is calculated; thus each frequent common substructure serves as a chemical descriptor type and the occurrence becomes a binary descriptor value. In addition, a modeling methodology is developed based on identifying frequently associated chemical fragments responsible for producing the desired class (activity or toxicity) of the molecules studied. These associated fragments are used as rules (Class Association Rules, or simply CARs) that are characterized by confidence and support values. These CARs can then be used to build a classifier for predicting an external dataset of molecules. The objectives of this study include: (a) provide a detailed description of the frequent subgraph mining approach as applied towards developing the fragment-based descriptors; (b) provide a detailed

description of the classification method used to utilize these descriptors; (c) validate these descriptors and methodology by developing predictive models for several experimental datasets; (d) compare the descriptors with the commonly used fingerprints descriptors; (e) provide an example of how the models generated can be interpreted to be useful for a medicinal chemist; and (f) finally discuss the applications of these descriptors in the drug design and development process by providing fragments that can be responsible for the target property such as mutagenicity. These examples should be of an interest to many researchers in the field who are concerned about toxicity and safety issues.

COMPUTATIONAL METHODS

Application of FFSM to chemical datasets to generate closed subgraphs and use them as chemical fragments

In this study, we are only interested in reporting whether the chemical fragment occurs or does not occur in each molecule of the dataset. The reason is that the method used later in developing models and identifying the pharmacophore/toxicophores needs only binary (0 or 1) values for the chemical fragments. Using the same example in Chapter 2 with same support value (66.7%), we will get the matrix in **Figure 3.1**.



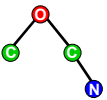
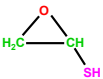
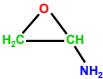
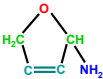
Fragment Dataset			
	1	1	0
	1	1	1
	0	1	1

Figure 3.1 Matrix of 1's and 0's for the occurrence or not, respectively, of the closed subgraphs (chemical fragments) in each molecule in the dataset.

Classification based on association (CBA) method

Classification based on association rules (CBA) is a useful method that can provide an interpretable classification models. The method (described in a paper by Liu, B., Hsu, W., and Ma, Y., 1998) relies on the integration of two powerful data mining techniques: Classification rule mining, which aims to discover a small set of rules in the database to form an accurate classifier (Quinlan, 1992, and Breiman et al., 1984); and Association rule mining which finds all rules in the database that satisfy some minimum support and minimum confidence constraints (Agrawal, and Srikant, 1994).

Let D be the dataset. Let I be the set of all items in D , and Y be the set of class labels. We say that a data case $d \in D$ contains $X \subseteq I$, a subset of items, if $X \subseteq d$. A class association rule (CAR) is an implication of the form $X \rightarrow y$, where $X \subseteq I$, and $y \in Y$. A rule $X \rightarrow y$ holds in D with confidence c if $c\%$ of cases in D that contain X are labeled with class y . The rule $X \rightarrow y$ has support s in D if $s\%$ of the cases in D contain X and are labeled with class y (Liu, B., Hsu, W., and Ma, Y., 1998). In other words:

$$\text{Confidence}(X \rightarrow y) = \frac{\| \{d \in D \mid X \cup y \subseteq d\} \|}{\| \{d \in D \mid X \subseteq d\} \|} \quad (1)$$

$$\text{Support}(X \rightarrow y) = \frac{\| \{d \in D \mid X \cup y \subseteq d\} \|}{\| D \|} \quad (2)$$

CBA consists of two parts: First, generating the complete set of rules (CARs) that satisfy the user-specified minimum support (called minsup) and minimum confidence (called minconf) constraints; Second, building a classifier by selecting CARs that's provide the highest accuracy for the given dataset. The algorithm for each part is described by Liu, B., Hsu, W., and Ma, Y., 1998.

The matrix in **Figure 3.1** is modified such that each molecule in the dataset shows its class (last coloumn) with value of either 1 (indicating activity or toxicity) or 0 (indicating no activity or toxicity), see **Figure 3.2**.



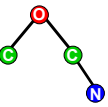
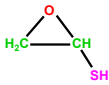
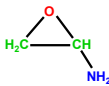

Fragment (id) Dataset	(a) 	(b) 	(c) 	Class
	1	1	0	0
	1	1	1	1
	0	1	1	1

Figure 3.2 Matrix of 1's and 0's for the occurrence or not, respectively, of the closed subgraphs (chemical fragments) in each molecule in the dataset, as well as the class label for the molecule.

In **Figure 3.2**, fragments are assigned id's (a, b, and c) for referral purposes in this example. So, when applying the CBA method to the dataset represented in the figure, the following CARs can be generated:

1. $\{a\} \rightarrow \{1\}$; with 50% confidence, and 33.3% support.
2. $\{b\} \rightarrow \{1\}$; with 66.7% confidence, and 66.7% support.
3. $\{c\} \rightarrow \{1\}$; with 100% confidence, and 66.7% support.
4. $\{a, b\} \rightarrow \{1\}$; with 50% confidence, and 33.3% support.
5. $\{a, c\} \rightarrow \{1\}$; with 100% confidence, and 33.3% support.
6. $\{b, c\} \rightarrow \{1\}$; with 100% confidence, and 66.7% support.
7. $\{a, b, c\} \rightarrow \{1\}$, with 100% confidence, and 33.3% support.
8. $\{a\} \rightarrow \{0\}$; with 50% confidence, and 33.3% support.
9. $\{b\} \rightarrow \{0\}$; with 33.3% confidence, and 33.3% support.
10. $\{a, b\} \rightarrow \{0\}$; with 50% confidence, and 33.3% support.

After that, in building the classifier, the following steps are used. Rules are sorted by their confidence first, then by their support. If two rules have same confidence and support, the one that is generated earlier comes first. Then, for each rule in the sorted sequence, if the rule correctly classifies at least one case, it is marked as a potential rule in the final classifier. Those cases covered by that rule are identified and removed. The total error is computed each time a rule is added, with the default class being the majority class in the data. The process continues until there is no rule or no cases left. Finally, the first rule at which there is the least number of errors recorded is identified as the cutoff rule after which all rules are discarded since they only produce more errors. The undiscarded rules and the default class form the classifier.

In another approach, after building the classifier, a further step is added to enrich the items (chemical fragments) in each rule with other items that are completely correlated with them in the dataset. This provides what is called the closed rules. The reason for doing this

approach is that is to provide the rules with some background items (chemical fragments) that are important for a certain class. These items were not included when building the classifier in CBA because simpler rules come first when rules with equal confidence and support are found. The results of this approach will be compared to those using CBA alone.

Therefore, based on the threshold one uses for the minimum confidence and the minimum support, different classifiers will be built. Then one can decide which one can be accepted as a way to do the external classification for validation of the method. So when classifying a molecule from an external dataset, we look at all the accepted rules in the order they are sorted in the classifier, and see which rule comes first that is applicable to the molecule, and the molecule is classified as having this particular class of that rule, such as mutagenic or non-mutagenic.

Experimental datasets

Three datasets were used in this study, which are the same as the ones used in Chapter 2. The first one included 1217 drug-like molecules with the MRTD (Maximum Recommended Therapeutic Dose) as their target property. This dataset was recently analyzed by the FDA modeling group (Contrera et al., 2004). Following the approach described in the original publication all molecules were divided into two classes based on the MRTD cut off value. This results in having 576 molecules with toxicological effect (adverse or undesirable pharmacological effect), and 641 molecules without toxicological effect. The second dataset is composed of 3434 drug-like molecules with the Salmonella mutagenic activity score as the target property. The score ranged from 10 to 80; molecules with no mutagenic activity have a score of 10, and the most mutagenic molecules have a score of 80. A cut off value is used to divide molecule into 2 classes: mutagenic and non-mutagenic, and thus resulting in 1615 mutagenic molecule versus 1819 non-mutagenic

molecules. This dataset was described in a paper by Votano et al., 2004. The third dataset included 195 molecules shown to be substrates (108 molecules) or non-substrates (87 molecules) of the P-Glycoprotein Protein (PGP). This dataset was analyzed previously in our group using several modeling techniques and descriptor sets and its molecules were taken from a paper by Penzotti et al., 2002.

Method validation

Dataset division into training and external validation sets. As explained earlier in Chapter 2, it is commonly accepted to confirm the validity of the modeling method by dividing the dataset into training and external validation sets (Benigni et al., 2000; Oloff et al., 2006; Trohalaki, Gifford, and Pachter, 2000; Zhang, Golbraikh, and Tropsha, 2006; Zhang et al., 2006). Figure 3.3 explains the work flow for the method development, division of datasets, and validation of the method.

Identifying chemical fragments using FFSM. Using support values in the range 5-10%, closed frequent subgraphs were identified for each dataset and were used as our binary chemical fragments descriptors. Notice that we only look at the presence or absence of a fragment in each molecule in the dataset. This way we can build classification models without having to discretize the values of the descriptors, and then being able to compare it later on with the binary fingerprints descriptors.

Generating rules and building classifiers. Once the chemical fragments are identified for the internal training set, we can generate class association rules (CARs) for the dataset. Multiple values of the minSupport ranging from 0.1-10%, crossed with multiple values of minConfidence range from 50-100% were used to generate the class association rules (CARs) followed by building the classifier. Therefore several classifiers

were built and the ones with the best accuracies were selected for the external validation. The accuracy of prediction for the external dataset will be used to validate the chemical fragments. Accuracies will also be compared to those using the closed-rule approach described earlier.

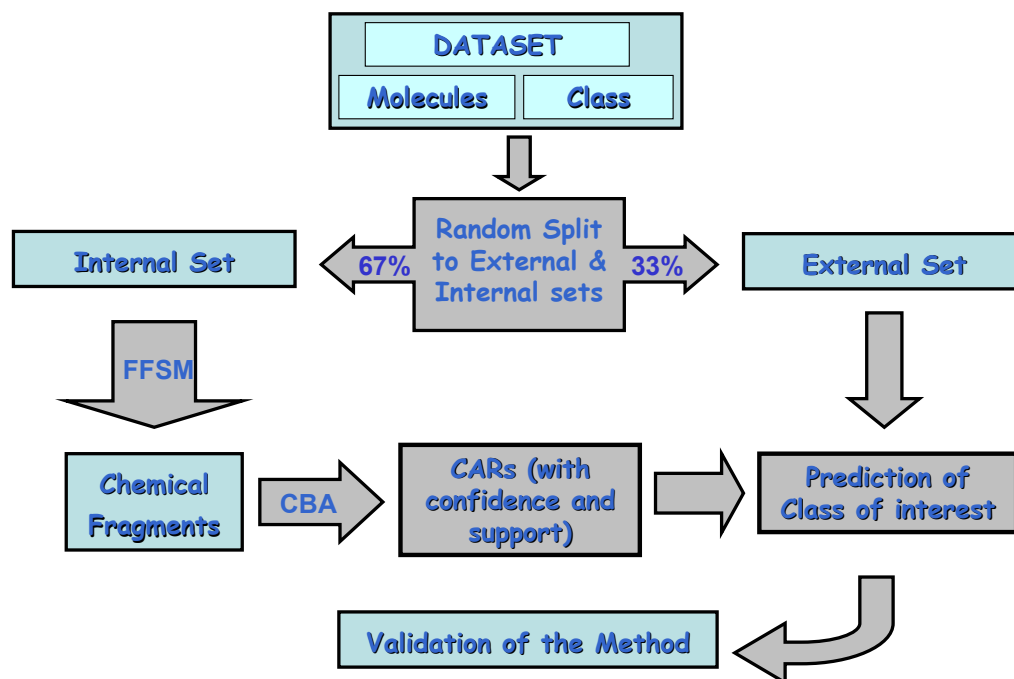


Figure 3.3 Work flow for the division of the datasets; identifying chemical fragments; generation of class association rules; and external validation.

Comparison with other molecular descriptors. To illustrate the usefulness of the descriptors generated, results obtained using CBA will be compared to those using the commonly used fingerprints descriptors. The fingerprints used are the MACCS keys fingerprints generated by MOE (Chemical Computing Group Inc.). The number of fingerprints provided is 166 feature keys.

RESULTS AND DISCUSSION

For the generation of chemical fragments using FFSM, the support values used were in the range 5 to 10% of each of the dataset. Only molecules in the internal dataset that have the property of interest (e.g., active or toxic molecules) were used in deriving the chemical fragments. Then, redundant fragments were eliminated leaving only closed ones. Fingerprints (MACCS keys) were generated for the internal dataset as well. Then, using the methodology described earlier, classifiers were built for the internal dataset. Several classifiers were built (using various confidence and support values) and the ones with the highest accuracies (lowest error) were used to predict the class for the molecules in the external dataset.

Results for the Salmonella mutagenicity. Starting with the fingerprints descriptors, best CBA classifier was obtained with *minSupport* in the range 0.05-0.1%, and *minConfidence* of 50-60%. **Table 3.1** shows the confusion matrix for this classifier which gave a total error of 19.6%. When validating this classifier by predicting the external dataset, the total prediction error jumped to 28.2% as shown in **Table 3.2**.

<div> <div>Predicted</div> <div>Actual</div> </div>	Mutagenic	Non mutagenic
Mutagenic	955	135
Non mutagenic	316	895

Table 3.1 Internal dataset for Salmonella, has a prediction a total error of 19.6% using the fingerprints descriptors.

<div> <div>Predicted</div> <div>Actual</div> </div>	Mutagenic	Non mutagenic
Mutagenic	414	111
Non mutagenic	207	399

Table 3.2 External validation for Salmonella, has a total error of 28.2% using fingerprints descriptors.

When using the chemical fragments derived using FFSM (with an absolute support value of 10, and maximum size of fragments limited to 10 atoms), the number of closed subgraphs representing the chemical fragments was 23,657. When using these descriptors, the best CBA classifier was obtained with *minSupport* of 0.1%, and *minConfidence* of 50-60%. **Table 3.3** shows the confusion matrix using this classifier for the internal set with a total error of 14.6%. When validating this classifier by predicting the external dataset, the total prediction error jumped to 22.0% as shown in **Table 3.4**.

Notice that the classifier generated using the fragment-based chemical descriptors gave less total error by 5% for the internal and 6% for the external prediction.

Predicted Actual	Mutagenic	Non mutagenic
Mutagenic	890	200
Non mutagenic	136	1074

Table 3.3 Internal dataset for Salmonella, has a prediction a total error of 14.6% using the fragment-based chemical descriptors.

Predicted Actual	Mutagenic	Non mutagenic
Mutagenic	374	150
Non mutagenic	99	509

Table 3.4 External validation for Salmonella, has a total error of 22.0% using fragment-based chemical descriptors.

Examples of associated fragments for the Ames Mutagenicity dataset

To demonstrate how this study can be used for interpreting the results, we choose the Salmonella Mutagenicity dataset as an example, and we generated class association rules (CBA) classifier for the entire dataset of 3,434 molecules, instead of just the internal dataset. Using FFSM with an absolute support value of 34, the number of closed subgraphs (chemical fragments) derived was 9,061 fragments. Then, CBA was used to build the classifier with a *minConfidence* of 50% and a *minSupport* 0.5%. **Table 3.5** shows an example selected rules (CARs) with at least a confidence of 90%. Each row represents a rule, where the fragments are found associated and responsible for the mutagenicity (Class T) or non-mutagenicity (Class F) of a number of molecules in the dataset represented by the confidence and support values. These are typically used to classify an unknown molecule.

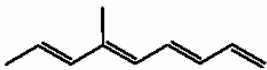
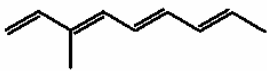
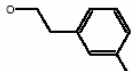
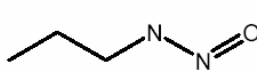
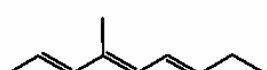
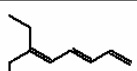
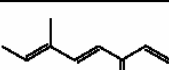
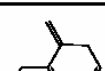
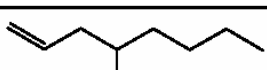
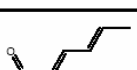



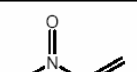

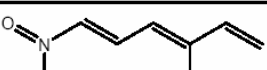

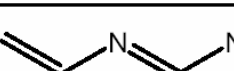



	1	2	Confidence%	Support%	Class
1			100.000%	3.526%	T
2			100.000%	3.263%	T
3			100.000%	1.836%	T
4			100.000%	1.544%	T
5			100.000%	1.224%	T
6			100.000%	1.224%	T
7			100.000%	1.195%	T
8			100.000%	1.195%	T
9			100.000%	1.136%	F
10			100.000%	1.049%	T
11			100.000%	1.049%	T
12			100.000%	0.670%	T
13			100.000%	0.641%	T
14			100.000%	0.554%	T
15			100.000%	0.554%	F
16			100.000%	0.524%	T

Table 3.5 Example of rules used in the classifier built by CBA.

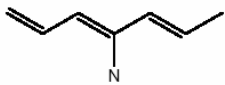
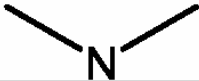
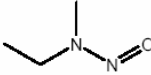
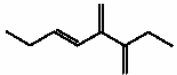
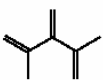
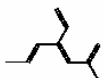

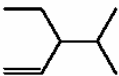
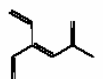
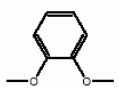
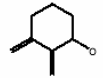
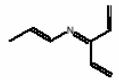

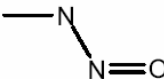

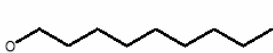
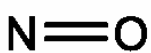

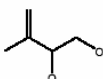
	1	2	Confidence%	Support%	Class
17			100.000%	0.524%	T
18			98.462%	1.894%	T
19			98.000%	1.457%	T
20			97.959%	4.283%	T
21			97.826%	1.340%	T
22			97.619%	1.224%	F
23			97.561%	1.195%	F
24			97.396%	5.594%	T
25			97.297%	1.078%	F
26			97.297%	1.078%	T
27			97.297%	1.078%	T
28			97.143%	1.020%	T
29			96.386%	2.418%	T
30			96.000%	1.457%	F
31			96.000%	0.728%	T
32			95.833%	1.399%	T

Table 3.5 Example of rules used in the classifier built by CBA.

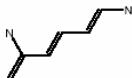

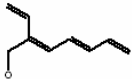
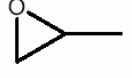
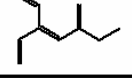
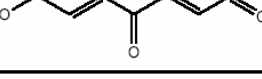
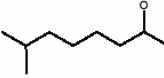
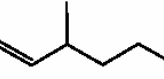

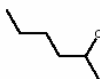
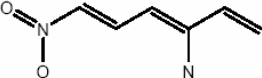
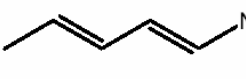


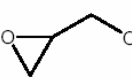
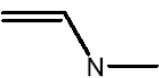
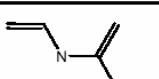

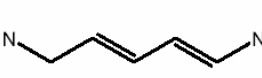
	1	2	Confidence%	Support%	Class
33			95.833%	0.699%	T
34			95.652%	2.010%	T
35			95.652%	1.340%	T
36			95.455%	2.564%	T
37			95.238%	1.224%	T
38			95.122%	1.195%	F
39			95.122%	1.195%	F
40			95.000%	4.079%	T
41			95.000%	1.748%	F
42			95.000%	1.166%	T
43			95.000%	1.166%	T
44			95.000%	0.583%	F
45			94.828%	1.690%	T
46			94.872%	1.136%	F
47			94.872%	1.136%	T
48			94.595%	1.078%	T

Table 3.5 Example of rules used in the classifier built by CBA.

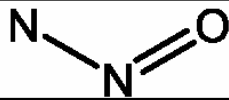
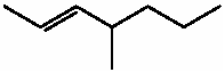
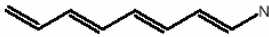

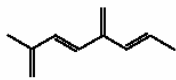

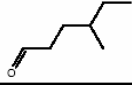
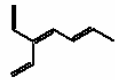

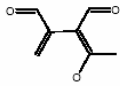
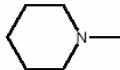
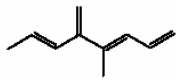
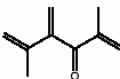
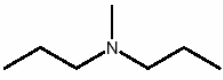
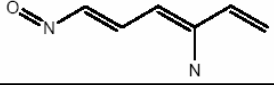
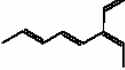
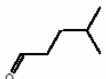
	1	2	Confidence%	Support%	Class
49			94.444%	2.098%	T
50			94.286%	1.020%	F
51			94.118%	0.991%	T
52			94.030%	3.904%	T
53			93.750%	1.399%	F
54			93.617%	1.369%	F
55			93.571%	4.079%	T
56			93.506%	2.244%	T
57			93.478%	1.340%	T
58			93.478%	1.340%	F
59			93.388%	3.526%	T
60			93.333%	1.748%	T
61			93.333%	1.311%	F
62			93.182%	1.282%	T
63			92.982%	1.661%	T
64			92.857%	1.632%	F

Table 3.5 Example of rules used in the classifier built by CBA.

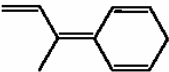
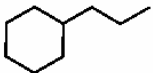
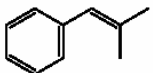
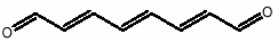
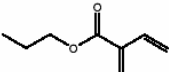
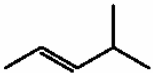
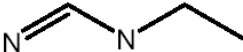
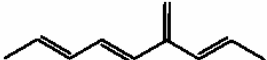
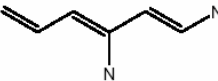
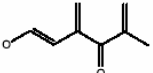
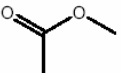

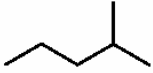
	1	2	Confidence%	Support%	Class
65			92.623%	3.555%	T
66			92.683%	1.195%	F
67			92.437%	3.467%	T
68		—N	92.308%	0.758%	T
69			92.105%	1.107%	F
70			92.105%	1.107%	F
71		—N	91.667%	0.699%	F
72			91.447%	4.429%	T
73			91.489%	1.369%	T
74			91.304%	1.340%	T
75			90.541%	2.156%	F
76		—N	90.323%	0.903%	T
77			90.244%	1.195%	F

Table 3.5 Example of rules used in the classifier built by CBA.

Results for the MRTD dataset. For the fingerprints descriptors, best CBA classifier was obtained with *minSupport* of 1% and *minConfidence* of 50-60%. **Table 3.6** shows the confusion matrix for this classifier which gave a total error of 11.7%. When validating this classifier by predicting the external dataset, the total prediction error jumped to 28.1% as shown in **Table 3.7**.

When using the chemical fragments derived using FFSM (with an absolute support value of 5, and maximum size of fragments limited to 10 atoms), the number of closed subgraphs representing the chemical fragments was 51,048. When using these descriptors, the best CBA classifier was obtained with *minSupport* of 0.2%, and *minConfidence* of 50-60%. **Table 3.8** shows the confusion matrix using this classifier for the internal set with a total error of 8.2%. When validating this classifier by predicting the external dataset, the total prediction error jumped to 26.2% as shown in **Table 3.9**.

Notice that the classifier generated using the fragment-based chemical descriptors gave slightly less total error by 2% for the internal and 3.5% for the external prediction.

Predicted Actual	Toxic	Non toxic
Toxic	356	24
Non toxic	70	354

Table 3.6 Internal dataset for MRTD, has a prediction a total error of 11.7% using the fingerprints descriptors.

Predicted Actual	Toxic	Non toxic
Toxic	148	48
Non toxic	68	149

Table 3.7 External validation for MRTD, has a total error of 28.1% using fingerprints descriptors.

<div> <div>Predicted</div> <div>Actual</div> </div>	Toxic	Non toxic
Toxic	347	33
Non toxic	33	391

Table 3.8 Internal dataset for MRTD, has a prediction a total error of 8.2% using the fragment-based chemical descriptors.

<div> <div>Predicted</div> <div>Actual</div> </div>	Toxic	Non toxic
Toxic	144	52
Non toxic	56	160

Table 3.9 External validation for MRTD, has a total error of 26.2% using fragment-based chemical descriptors.

Examples of associated fragments for the MRTD dataset

To demonstrate how this study can be used for interpreting the results, we generated class association rules (CBA) classifier for the entire dataset of 1,217 molecules, instead of just the internal dataset. Using FFSM with an absolute support value of 5 with maximum size of subgraphs as 8 nodes, the number of closed subgraphs (chemical fragments) derived was 25,318 fragments. Then, CBA was used to build the classifier with a *minConfidence* of 50% and a *minSupport* 1%. **Table 3.10** shows an example selected rules (CARs) with at least a confidence of 90%. Each row represents a rule, where the fragments are found associated and responsible for the toxicity (Class Toxic) or non-toxicity (Class Non) of a number of molecules in the dataset represented by the confidence and support values. These are typically used to classify an unknown molecule.




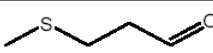
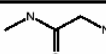
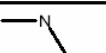


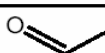
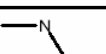
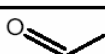


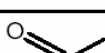

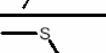
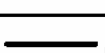

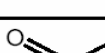




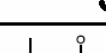
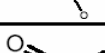


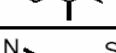
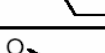

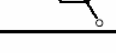

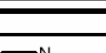

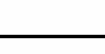
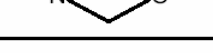
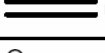
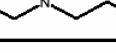
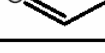
	1	2	Confidence%	Support%	Class
1			100.000%	4.930%	Non
2			100.000%	4.601%	Non
3			100.000%	4.519%	Non
4			100.000%	4.519%	Non
5			100.000%	3.780%	Non
6			100.000%	3.698%	Non
7			100.000%	3.615%	Non
8			100.000%	3.533%	Non
9			100.000%	3.369%	Non
10			100.000%	3.369%	Toxic
11			100.000%	3.287%	Non
12			100.000%	3.122%	Non
13			100.000%	3.122%	Non
14			100.000%	3.122%	Non
15			100.000%	2.136%	Non
16			100.000%	2.054%	Toxic
17			100.000%	2.054%	Toxic
18			100.000%	1.890%	Non
19			100.000%	1.808%	Non
20			100.000%	1.808%	Toxic
21			100.000%	1.726%	Toxic
22			100.000%	1.726%	Non
23			100.000%	1.726%	Toxic
24			100.000%	1.726%	Toxic

Table 3.10 Example of rules used in the classifier built by CBA.

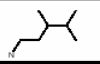
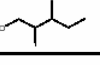



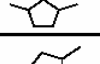
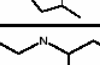
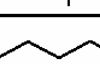
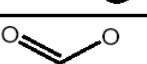
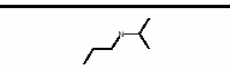

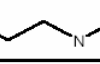
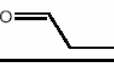
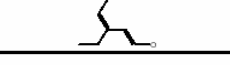

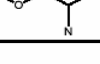
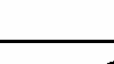
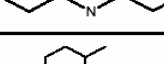
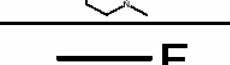

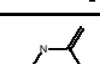
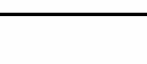
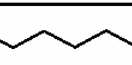
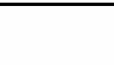
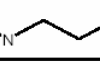
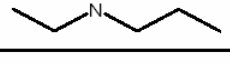
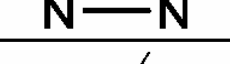
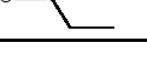
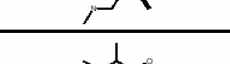
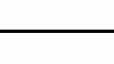
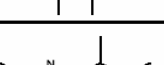
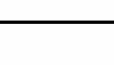
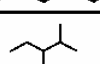




	1	2	Confidence%	Support%	Class
25			100.000%	1.643%	Toxic
26			100.000%	1.643%	Toxic
27			100.000%	1.643%	Non
28			100.000%	1.561%	Toxic
29			100.000%	1.561%	Toxic
30			100.000%	1.561%	Toxic
31			100.000%	1.561%	Toxic
32			100.000%	1.561%	Toxic
33			100.000%	1.561%	Toxic
34			100.000%	1.561%	Toxic
35			100.000%	1.479%	Non
36			100.000%	1.479%	Toxic
37			100.000%	1.479%	Toxic
38			100.000%	1.479%	Toxic
39			100.000%	1.397%	Toxic
40			100.000%	1.397%	Toxic
41			100.000%	1.397%	Toxic
42			100.000%	1.397%	Toxic
43			100.000%	1.397%	Non
44			100.000%	1.315%	Toxic
45			100.000%	1.315%	Toxic
46			100.000%	1.315%	Toxic
47			100.000%	1.315%	Toxic
48			100.000%	1.315%	Toxic

Table 3.10 Example of rules used in the classifier built by CBA.

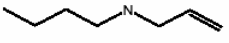



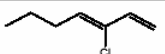


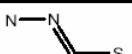
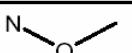
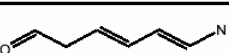

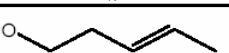
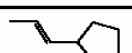




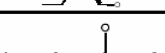

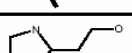
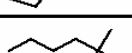



	1	2	Confidence%	Support%	Class
49			100.000%	1.315%	Toxic
50			100.000%	1.315%	Non
51		$\equiv O$	100.000%	1.315%	Non
52		$\equiv O$	100.000%	1.315%	Toxic
53			100.000%	1.233%	Toxic
54			100.000%	1.233%	Toxic
55			100.000%	1.233%	Toxic
56			100.000%	1.233%	Non
57			100.000%	1.233%	Non
58			100.000%	1.233%	Non
59		$O=O$	100.000%	1.233%	Toxic
60		$O=O$	100.000%	1.233%	Toxic
61			100.000%	1.150%	Toxic
62			100.000%	1.150%	Toxic
63			100.000%	1.150%	Toxic
64			100.000%	1.150%	Non
65		$\equiv O$	100.000%	1.150%	Toxic
66		$O=O$	100.000%	1.150%	Toxic
67			100.000%	1.068%	Toxic
68			100.000%	1.068%	Toxic
69			100.000%	1.068%	Toxic
70			100.000%	1.068%	Non
71		$\equiv O$	100.000%	1.068%	Toxic
72		$\equiv O$	100.000%	1.068%	Toxic

Table 3.10 Example of rules used in the classifier built by CBA.




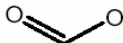
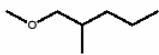
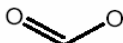


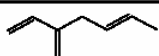
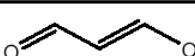
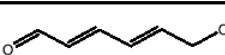





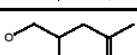
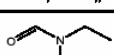




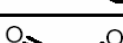


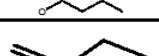


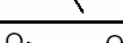
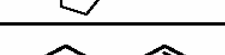



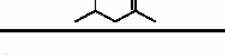

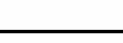




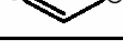
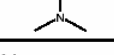
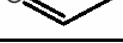
	1	2	Confidence%	Support%	Class
73			100.000%	1.068%	Non
74			100.000%	1.068%	Toxic
75			100.000%	1.068%	Non
76			100.000%	1.068%	Toxic
77			100.000%	1.068%	Non
78			100.000%	1.068%	Toxic
79			98.246%	4.684%	Non
80			98.077%	4.273%	Toxic
81			97.778%	3.698%	Toxic
82			97.778%	3.698%	Non
83			97.674%	3.533%	Non
84			97.619%	3.451%	Toxic
85			97.222%	2.958%	Toxic
86			97.222%	2.958%	Toxic
87			97.059%	2.794%	Toxic
88			96.923%	5.341%	Non
89			96.970%	2.712%	Toxic
90			96.774%	2.547%	Toxic
91			96.774%	2.547%	Toxic
92			96.610%	4.848%	Non
93			96.429%	2.301%	Toxic
94			96.429%	2.301%	Toxic
95			96.364%	4.519%	Non
96			96.226%	4.355%	Non

Table 3.10 Example of rules used in the classifier built by CBA.

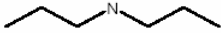

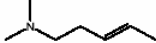







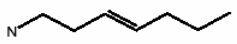


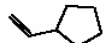
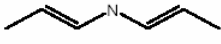
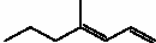




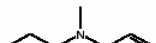
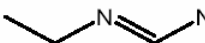




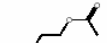





	1	2	Confidence%	Support%	Class
97			96.154%	2.136%	Toxic
98			96.000%	2.054%	Toxic
99			96.000%	2.054%	Toxic
100			95.833%	1.972%	Toxic
101			95.775%	5.834%	Non
102			95.652%	1.890%	Toxic
103			95.652%	1.890%	Toxic
104			95.455%	1.808%	Toxic
105			95.455%	1.808%	Toxic
106			95.238%	1.726%	Toxic
107			95.238%	1.726%	Toxic
108			95.238%	1.726%	Toxic
109			95.238%	1.726%	Toxic
110			95.000%	1.643%	Toxic
111			94.737%	1.561%	Toxic
112			94.737%	1.561%	Non
113			94.737%	1.561%	Non
114			94.444%	2.958%	Toxic
115			94.444%	1.479%	Toxic
116			94.444%	1.479%	Toxic
117			94.444%	1.479%	Toxic
118			94.231%	4.273%	Toxic
119			94.118%	2.794%	Toxic
120			94.118%	2.794%	Toxic

Table 3.10 Example of rules used in the classifier built by CBA.






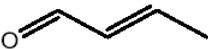



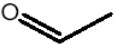





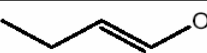
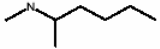

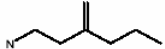

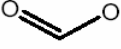



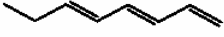



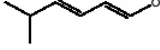

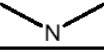


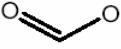
	1	2	Confidence%	Support%	Class
121			94.118%	1.397%	Non
122			94.118%	1.397%	Toxic
123			94.118%	1.397%	Toxic
124			94.118%	1.397%	Non
125			93.939%	2.712%	Toxic
126			93.939%	2.712%	Non
127			93.878%	4.026%	Toxic
128			93.750%	2.629%	Toxic
129			93.750%	1.315%	Toxic
130			93.750%	1.315%	Toxic
131			93.750%	1.315%	Non
132			93.750%	1.315%	Toxic
133			93.548%	2.547%	Toxic
134			93.333%	2.465%	Toxic
135			93.333%	1.233%	Non
136			93.333%	1.233%	Non
137			93.023%	3.533%	Toxic
138			92.857%	1.150%	Toxic
139			92.857%	1.150%	Toxic
140			92.857%	1.150%	Non
141			92.857%	1.150%	Toxic
142			92.857%	1.150%	Toxic
143			92.857%	1.150%	Toxic
144			92.500%	3.287%	Toxic

Table 3.10 Example of rules used in the classifier built by CBA.



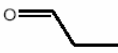



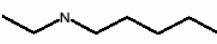


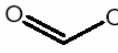


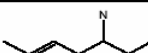

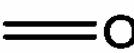




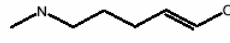


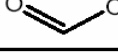

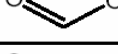
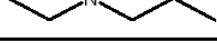
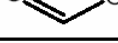

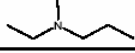
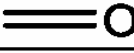
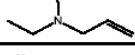


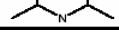
	1	2	Confidence%	Support%	Class
145			92.308%	2.136%	Toxic
146			92.308%	2.136%	Toxic
147			92.105%	6.245%	Non
148			92.000%	2.054%	Toxic
149			92.000%	2.054%	Toxic
150			91.667%	5.916%	Toxic
151			91.667%	3.944%	Toxic
152			91.667%	2.958%	Toxic
153			91.667%	1.972%	Toxic
154			91.667%	1.972%	Toxic
155			91.525%	4.848%	Toxic
156			91.304%	1.890%	Toxic
157			91.304%	1.890%	Toxic
158			91.176%	2.794%	Toxic
159			90.909%	4.519%	Non
160			90.909%	3.615%	Toxic
161			90.909%	2.712%	Toxic
162			90.741%	4.437%	Toxic
163			90.476%	1.726%	Toxic
164			90.476%	1.726%	Toxic
165			90.323%	2.547%	Toxic
166			90.000%	1.643%	Toxic
167			90.000%	1.643%	Toxic
168			90.000%	1.643%	Toxic

Table 3.10 Example of rules used in the classifier built by CBA.

Results for the PGP dataset. The PGP dataset gave a different pattern of results unlike those seen with the Salmonella and MRTD datasets. Using the fingerprints descriptors, best CBA classifier was obtained with *minSupport* of 0.1-3% and *minConfidence* of 50-80%. **Table 3.11** shows the confusion matrix for this classifier which gave a total error of 1.5%. When validating this classifier by predicting the external dataset, the total prediction error jumped drastically to 30.2% as shown in **Table 3.12**.

When using the chemical fragments derived using FFSM (with an absolute support value of 3, and maximum size of fragments limited to 6 atoms), the number of closed subgraphs representing the chemical fragments was 2,491. When using these descriptors, the best CBA classifier was obtained with *minSupport* of 2%, and *minConfidence* of 55%. **Table 3.13** shows the confusion matrix using this classifier for the internal set with a total error of 3.8%. When validating this classifier by predicting the external dataset, the total prediction error jumped to 23.8% as shown in **Table 3.14**.

Predicted Actual	Active	Inactive
Active	71	2
Inactive	0	59

Table 3.11 Internal dataset for PGP, has a prediction a total error of 1.5% using the fingerprints descriptors.

Predicted Actual	Active	Inactive
Active	28	7
Inactive	12	16

Table 3.12 External validation for PGP, has a total error of 30.2% using fingerprints descriptors.

Predicted Actual	Active	Inactive
Active	64	4
Inactive	1	58

Table 3.13 Internal dataset for PGP, has a prediction a total error of 3.8% using the fragment-based chemical descriptors.

Predicted Actual	Active	Inactive
Active	29	6
Inactive	9	19

Table 3.14 External validation for PGP, has a total error of 23.8% using fragment-based chemical descriptors.

In the case of the PGP, results were more interesting and are helping us to understand the fragment based descriptors better. First thing to notice when comparing the fingerprints with the fragment-based descriptors is that the error for the fingerprints was lower than that of the fragment-based descriptors (1.5% vs. 3.8%). However, for the external set, the predictions using the fragment-based were more accurate; the total error using the fingerprints was 30.2% compare to 23.8% for the fragment based descriptors. This means that the classifier built for the fingerprints was overfit for the internal dataset and failed to predict the external set. However, even for the fragment-based descriptors, the change of the total error from 3.8% to 23.8% is a sign of overfit too, but not as bad as that of the fingerprints descriptors.

To further investigate the reason behind this overfitting for the fragment-based descriptors, another set of fragment-based descriptors was generated, but this time not using only the internal dataset, but the whole dataset (internal and external compined together). The absolute support value of FFSM used in this case was 20, resulting in 1082 closed subgraphs. At the same time, the classifier was built for only the internal (training) set and then validated using the external set. Using a minSupport of 1% and a minConfidence of 60%, a classifier with total error for the internal dataset of 3.0% was obtained, and the total error for the external validation dataset was 17.7%, see **Table 3.15** and **Table 3.16**. Obviously, prediction accuracy for the internal dataset is slightly better this time (total error of 3.8% compared to 3%), and the prediction accuracy for the external dataset is much better (total error of 17.7% compared to 23.8%).

Predicted Actual	Active	Inactive
Active	73	3
Inactive	1	57

Table 3.15 Internal dataset for PGP, has a prediction a total error of 3.0% using the fragment-based chemical descriptors derived from the whole PGP dataset.

Predicted Actual	Active	Inactive
Active	23	10
Inactive	1	28

Table 3.16 External validation for PGP, has a total error of 17.7% using fragment-based chemical descriptors derived from the whole dataset.

Examples of associated fragments for the PGP dataset

To demonstrate how this study can be used for interpreting the results, we generated class association rules (CBA) classifier for the entire dataset of 195 molecules, instead of just the internal dataset. Using FFSM with an absolute support value of 20, the number of closed subgraphs (chemical fragments) derived from the whole dataset (not only the internal training dataset) was 1,082 fragments. Then, CBA was used to build the classifier with a *minConfidence* of 60% and a *minSupport* 1%. **Table 3.17** shows an example selected rules (CARs) with at least a confidence of 90%. Each row represents a rule, where the fragments are found associated and responsible for the activity (Class Active) or inactivity (Class Non) of a number of molecules in the dataset represented by the confidence and support values. These are typically used to classify an unknown molecule.

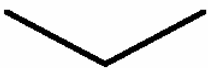

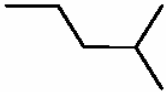
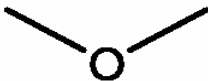


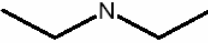
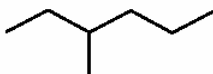
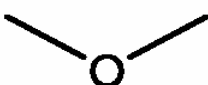


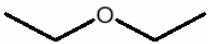

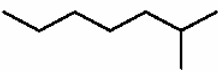
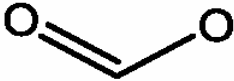
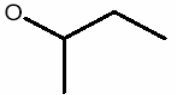
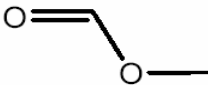

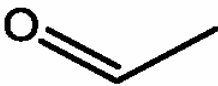
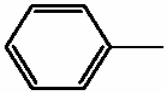
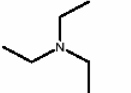
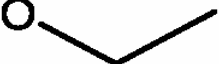

	1	2	Confidence%	Support%	Class
1			100.000%	12.308%	Active
2			100.000%	10.769%	Active
3			100.000%	10.256%	Active
4			100.000%	10.256%	Active
5			100.000%	10.256%	Active
6			100.000%	9.744%	Active
7			100.000%	9.231%	Non
8			100.000%	8.718%	Active
9			100.000%	8.718%	Active
10			100.000%	8.205%	Active
11			100.000%	7.692%	Active
12			100.000%	7.692%	Active

Table 3.17 Example of rules used in the classifier built by CBA.

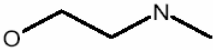

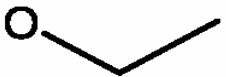


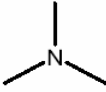
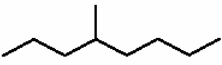



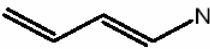
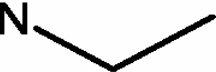

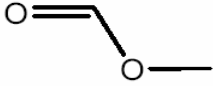
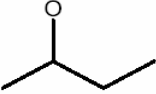

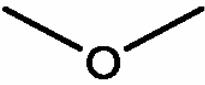
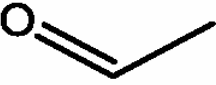
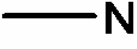
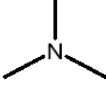
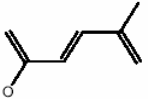
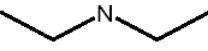

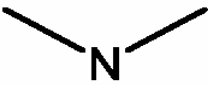
	1	2	Confidence%	Support%	Class
13			100.000%	7.179%	Active
14			100.000%	7.179%	Active
15			100.000%	7.179%	Active
16			100.000%	6.667%	Active
17			100.000%	6.667%	Active
18			100.000%	6.667%	Non
19			100.000%	6.667%	Active
20			100.000%	6.667%	Active
21			100.000%	6.154%	Active
22			100.000%	6.154%	Active
23			100.000%	5.641%	Active
24			100.000%	5.641%	Active

Table 3.17 Example of rules used in the classifier built by CBA.


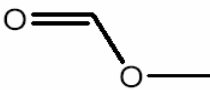



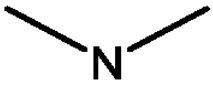
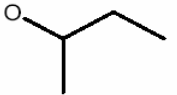

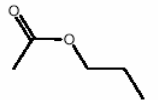


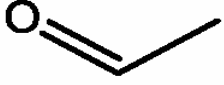
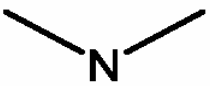
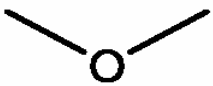
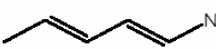




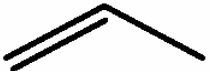



	1	2	Confidence%	Support%	Class
25			100.000%	5.641%	Active
26			100.000%	5.641%	Active
27			100.000%	5.128%	Non
28			100.000%	4.615%	Active
29			100.000%	3.590%	Non
30			100.000%	3.077%	Active
31			100.000%	2.564%	Non
32			100.000%	2.564%	Non
33			100.000%	1.026%	Non
34			95.652%	11.795%	Active
35			92.593%	13.846%	Active
36			90.000%	5.128%	Non

Table 3.17 Example of rules used in the classifier built by CBA.

In conclusion, what these last results telling us is: the chemical-fragments derived are highly dependent on the internal training set. In other words, unless a chemical fragment occurs frequently enough in the internal training set, we will not be able to find the active molecules that contain this fragment. That's why including the external dataset in deriving the fragments gave a better prediction for the external dataset. This is a problem that appears particularly in datasets such as the PGP where fragment-based descriptors are intended to be used as a way to define the pharmacophores. This problem is a short coming of these descriptors as we will discuss shortly, and we will also discuss the solution for that problem in summary and future directions in Chapter 5.

Results of replacing rules in the CBA classifier with the closed rules. Often, we have two rules such that one of them has all its items present in the other rule, and both rules are completely correlated in their appearance in the dataset, and therefore have the same confidence and support value. The rule with more items in this case is called the closed rules (since it contains the closed frequent patterns), and the other rule would be the simple rule, and usually is generated prior the closed rule. Therefore, when building the classifier, the simple rule is selected instead of the closed rule. To answer the question whether selecting the simplest rule is better than selecting the closed one, each rule that was selected by the classifier was replaced by its closed one, and the prediction accuracy was calculated for the external dataset. Ofcourse in this case, the accuracy of the internal dataset will stay the same since the two rules are completely correlated in the internal training set to begin with.

The accuracies for the external dataset stayed the same in almost all cases for the Salmonella and MRTD datasets. But, for the PGP dataset, the accuracy improved by 1.5-

3.0% for some cases, and stayed the same in the rest of the cases, but never got worse in any case. An example of a case where results improved using the closed rules is shown in the tables below. In this example, an absolute support value of 5 was used to find frequent subgraphs with size no larger than 6 atoms (nodes). The number of closed subgraphs was 18,907 constituting the fragment-based descriptors. A *minConfidence* of 66% and a *minSupport* of 7% were used to build the classifier. **Table 3.18** shows the prediction accuracy for the internal set, and **Table 3.19** and **Table 3.20** show the prediction accuracy for the external dataset using the simple rules and the closed rules, respectively.

Predicted Actual	Active	Inactive
Active	64	9
Inactive	9	50

Table 3.18 Internal dataset for PGP, has a total error of 13.6% using the fragment-based chemical descriptors.

Predicted Actual	Active	Inactive
Active	28	7
Inactive	10	18

Table 3.19 External validation for PGP, has a total error of 27.0% using fragment-based chemical descriptors and simple rules built by CBA.

Predicted Actual	Active	Inactive
Active	28	7
Inactive	8	20

Table 3.20 External validation for PGP, has a total error of 23.8% using fragment-based chemical descriptors and closed rules in place of the simple rules built by CBA.

Further more, to simulate the use of these classifiers in database screening, the external dataset for PGP was dissolved in the Maybridge database of 57,626 molecules presumed inactives, and the classifier (for same example in **Tables 3.18-3.20**) was used to screen the database seeded with the external dataset. **Table 3.21** shows the accuracy results using the simple rules, and **Table 3.22** shows the accuracy results using the closed rules. Using the closed rules slightly reduced the total error from 14.3% to 12.5%. Another way to represent the results of the hit list obtained is by using the Hit Rate (number of actives recovered / total hits recovered). Using simple rules built by CBA gave a hit rate of 0.34%, while using the closed rules gave a slightly better hit rate of 0.39%.

These results imply that for datasets such as PGP where the fragment-based descriptors are intended to be used for identifying the pharmacophores, using closed rules will always improve the prediction accuracy. This is simply due to the inclusion of some background fragments that can aid in avoiding the false prediction of inactive molecules as active. This can be clearly seen in **Tables 3.19** and **3.20**, as well as **Tables 3.21** and **3.22**, where the improvement in accuracy came from the reduction of the number of inactive compounds that were predicted as active. On the other hand, for datasets such as Salmonella and MRTD, fragments responsible for the target property are independent and do not rely on some background structure that might be required for activity as was in the PGP case.

Predicted Actual	Active	Inactive
Active	28	7
Inactive	8211	49415

Table 3.21 Screening Maybridge database seeded with the external dataset of PGP gave a total error of 14.3% using fragment-based chemical descriptors and simple rules built by CBA.

Predicted Actual	Active	Inactive
Active	28	7
Inactive	7212	50414

Table 3.22 Screening Maybridge database seeded with the external dataset of PGP gave a total error of 12.5% using fragment-based chemical descriptors and closed rules in place of the simple rules built by CBA.

Weaknesses and strengths of the descriptors and methodology

The strength of the descriptors comes from the fact that it can not miss an important chemical fragment in a dataset. Since defining all possible combination of atoms will give an exponentially large number of chemical fragments, these descriptors can identify the fragments related only to that particular dataset of interest. The methodology used to handle these descriptors (generating class association rules) provides a high chance of predicting external sets with easily interpretable rules to the medicinal chemist. What we see as the weakness of this method is the fact that fragments that are interchangeable (i.e., have the same physicochemical or pharmacophoric characteristic) will not be treated equally, and therefore, unless each of these interchangeable fragments (also known as bioisosters) occur frequently enough in the dataset to be used as descriptors, it won't be taken into account when doing the predictions. Also, if you were to explore an external database of compounds looking for potential leads, unless all important fragments are already discovered in your internal training dataset, you will not be able to come up with a lead with fragments different from what you already have in your dataset. This can be solved in the future by providing a database of bioisosters for the fragments defined, and therefore allowing us to expand the applicability domain of these descriptors to identify new leads, as we shall explain summary and future directions in Chapter 5.

CONCLUSIONS

As the results show, the fragment-based descriptors can perform at least as good as the finger prints, and in some cases better than the fingerprints descriptors. The descriptors are further utilized by a methodology that takes care of the combined effect of these fragments in predicting the target property of interest, such as activity to a certain receptor,

or toxicity or mutagenicity. Medicinal chemists can use these descriptors along with the methodology to identify important fragments for future predictions, especially since that these descriptors are easily interpretable and understood by any medicinal chemist.

CHAPTER 4

DEVELOPMENT OF POSE-SCORING FUNCTION FOR PROTEIN-LIGAND BINDING BASED ON FREQUENT PATTERNS OF INTER-ATOMIC INTERACTIONS AT THEIR INTERFACES

INTRODUCTION

Structure based drug design (SBDD) is one of the most popular and powerful modern methods for computer aided drug design (Brooijmans, N., and Kuntz, I., 2004; Kitchen, D., Decornez, H., Furr, J., and Bajorath, J., 2004). Since the first seminal paper on SBDD was published in 1982 by the Kuntz group, this approach has been used successfully in numerous studies resulting in some cases (such as HIV protease inhibitors) in the design of approved drugs (Wlodawer, A., and Vondrasek, J., 1998). Two major components of SBDD include docking and scoring. Docking is the process of finding the correct pose for a small molecule in the binding pocket of the protein receptor and scoring typically implies the prediction of binding affinity for a pose. Docking and scoring are frequently integrated within the same software so the term ‘docking’ is often used in more global sense than merely placing a molecule within the binding site. Most common application of docking is screening a virtual or combinatorial library of small molecules to find those that fit into the binding site and bind tightly to the receptor. Another application is lead optimization, which plays a critical role in the drug discovery process. In lead optimization molecules that are expected to be more potent than known compounds are designed by studying and

analyzing the ligand orientation in the binding site. Therefore, finding the correct (i.e., native) pose of binding is essential in this case and is different from screening (docking) to find ligand that bind to the binding site. Initially, several scoring functions have been developed to do screening (docking), and these scoring functions perform poorly in identifying the correct pose accurately, which is why people converged to docking and then using different functions to score relative poses; using in some cases consensus scoring (Kitchen et al., 2004; Muegge, I., and Martin, Y., 1999; Verdonk et al., 1997; Klebe et al., 2003; Wang, R., Lu, Y., and Wang, S., 2003; Vajda, S., and Guarnieri, F., 2006). Although numerous robust and accurate algorithms are available to fit the molecule into the binding site, there remain significant challenges in developing scoring functions that can find the binding ligands, and more importantly accurately identifying the correct binding pose. It is widely recognized that the development of accurate scoring functions continues to be a major limiting factor in ensuring greater success of SBDD (Kitchen et al., 2004).

Scoring functions can be generally classified into three types. Force-field-based scoring functions rely on explicitly computed electrostatic and van der Waals interaction energies between the ligand and the protein. Empirical scoring functions are defined as the sum of individual uncorrelated energy terms (such as: free energy of hydrogen bonding, ionic interactions, hydrophobic interactions, metallic interactions, rotational entropy, and solvation energy) and the regression analysis is used to optimize the regression coefficients so that the model reproduces experimental data such as binding energies. Knowledge-based scoring functions are designed based on various statistical parameters that could reflect the interaction between ligands and receptors such as statistics of pairwise atomic contacts (Kitchen et al., 2004). They implicitly capture binding effects that are difficult to model

explicitly e.g., hydrophobic interactions. Knowledge-based scoring functions are computationally simple allowing for fast and efficient scoring of large sets of ligand receptor complexes resulting from docking. On the other hand, their derivation is essentially based on information encoded in limited sets of protein-ligand complexes. However, this limitation is diminishing due to the exponential increase in the number of protein-ligand complexes available through X-ray and NMR studies. Therefore, researchers are becoming more interested in exploring these complexes to gather the information needed to improve the accuracy docking and scoring.

Most scoring functions focus on ligand ranking based on their predicted binding affinities rather than based on direct scoring of their binding poses with regard to “native-like” orientation of the docked ligand. Many knowledge based scoring functions are derived in the form of pairwise atom interaction pseudopotentials resulting from the analysis of interacting atoms at the interface of protein-ligand complexes. For instance, PMF (Muegge, I., and Martin, Y., 1999; Muegge, I., 2006), BLEEP (Nobeli, I., Mitchell, J., Alex, A., and Thornton, J., 2001), and SMOG2001 (Ishchenko, A., and Shakhnovich, E., 2002) calculate the potential energy based on the statistical distribution of distances between pairs of pre-defined atom types. Some scoring functions define certain regions of interactions for each protein amino acid residue that might be occupied by the ligand atoms (Moreno, E., and Leon, K., 2002). A different approach designs a library of information for 250 pre-defined chemical groups showing their preferred geometries, the library is called IsoStar (Bruno, I., Cole, J., Lommerse, J., Rowland, R., Taylor, R., and Verdonk, M., 1997). Another approach designs a database called ReLiBase (Hendlich, M., Bergner, A., Gunther, J., and Klebe, G., 2003) for comprehensive analysis of protein-ligand interactions;

the database is also used to develop a scoring function called DrugScore (Gohlke, H., Hendlich, M., and Klebe, G., 2000). Another interesting approach uses both the experimental data to provide preferred geometries as well as analytical functional forms to describe the distribution of the experimental data, therefore providing smooth functions to calculate the Probabilistic Receptor Potentials for 21 protein atom types (Labute, P., 2001).

These approaches have been used to do fragment-based de novo design and to look at the binding pose, even though they have been developed to rank ligands based on their binding affinities, but not to identify accurate binding poses, which makes the results inaccurate, as mentioned earlier. Here in this study, we will focus only on identifying the correct binding pose. We introduce a novel knowledge-based scoring function that can identify efficiently the correct pose among a number of poses (decoys) for a given protein-ligand complex. The scoring function is derived from the frequent patterns of inter-atomic interactions that occur at the interface of crystallographically determined protein-ligand complexes. Frequent patterns and their internal coordinates are considered “classical” that a test pose of the ligand is scored against to evaluate its “nativity”. More specifically, given a number of poses produced by a computational docking program for a ligand in the protein binding site, patterns of interaction are identified at the interface of each pose. These patterns are then analyzed for their geometrical similarity to “classical” templates and the score for each pose is calculated based on the number of native-like patterns as well as the frequency of the corresponding “classical” patterns. Thus, the higher the geometric similarity and frequency the better the score is.

We show that the approach that we introduce in this paper was able to accurately identify the correct native pose among other computationally generated non-natives poses

for 1091 protein-ligand complexes. We also demonstrate that the accuracy of predicting the correct binding pose using our approach was significantly higher than using five commercially available scoring functions (such as Shapegauss, PLP, Chemgauss, Chemscore, and Screenshot) both independently as well as using their consensus scoring. We believe that the approach described herein is different from all scoring methods described in the literature. Specifically, it is not limited to using pre-defined chemical groups; instead, new patterns can always be derived and scored as long as they occur frequently in the experimentally determined protein-ligand complexes. Furthermore, the proposed method goes beyond traditional pairwise scoring of interatomic contacts, i.e., it employs multiatomic interaction patterns and consequently it should take into account inherently the cooperative effect of interaction between proteins and their ligands. We suggest that the scoring method described in this report could be successfully used to refine the lists of poses generated by popular docking programs.

COMPUTATIONAL METHODS

Dataset of Protein-Ligand Complexes

The dataset used in this study is the “refined set” of the PDBbind v.2004 (Wang et al., 2004; Wang et al., 2005), which is composed of 1,091 protein-ligand complexes. Each protein-ligand complex in the “refined set” is characterized by the following parameters:

1. It is a crystal structure with an overall resolution ≤ 2.5 °Å;
2. It is a “clean” binary complex formed between one protein and one ligand;
3. It is a non-covalently bound complex without any severe clash between the protein and the ligand;
4. It has an experimentally determined K_d or K_i value;

5. The ligand consists of only C, N, O, S, P, H, and halogens, and its molecular weight is lower than 1,000; and

6. There are no unnatural amino acid residues in the binding site of the protein.

By design, this set is grouped into clusters based on protein sequence similarity. We have identified 77 clusters such that all pairs of proteins within one cluster shared 90% or greater similarity. In each cluster we selected three representative members: one with the highest binding affinity; one with the lowest binding affinity; and one with the median binding affinity. The resulting 231 representative complexes form the “core set” of the PDBbind database. For our study, these 231 complexes were selected as the external testing set to develop the scoring function, and the remaining part of the refined set (i.e., after excluding the 231 core set complexes) composed of 860 complexes were selected as the internal training set.

Graph Representation of the Protein-Ligand Interface

For each protein-ligand complex in the internal training set, interacting atoms at the interface were identified as those within a cut off distance of 3.5 Å. This specific cut off was chosen because it covers the majority of the highly specific and directional interactions (polar, hydrogen-bond, and charge transfer interactions) as well as non-directional van der Waals interactions (Gohlke, H., Hendlich, M., and Klebe, G., 2000). If a water molecule was found at the interface, protein and ligand atoms within 3.5 Å of the water molecule were also considered interacting. In addition, atoms that are directly bound to these interacting atoms were also included as part of the interface. The atoms and bond types were assigned according to the notation given in Tripos SYBYL Mol2 file format. Connecting the interacting atoms at the protein-ligand interface creates an interaction

network that could be regarded as an undirected labeled graph where interacting atom-vertices are linked by graph edges. Thus, each atom at the interface is represented by a labeled node and each intramolecular bond and a non-bonded interacting pair of atoms is represented by a labeled edge, see **Figure 4.1**.

Each protein-ligand complex will then have at least one connected graph representing the inter-atomic interaction between the protein and the ligand at their interface (it is theoretically feasible that some protein ligand complexes may have a configuration of the interface resulting in two or even more interfacial graphs that would be disconnected from each other; see additional discussion *vide infra*). Therefore, for N protein-ligand complexes, we will have at least N connected graphs. As we discuss below, the representation of protein ligand interfaces as connected graphs affords the application of subgraph mining techniques to extract frequent interaction patterns.

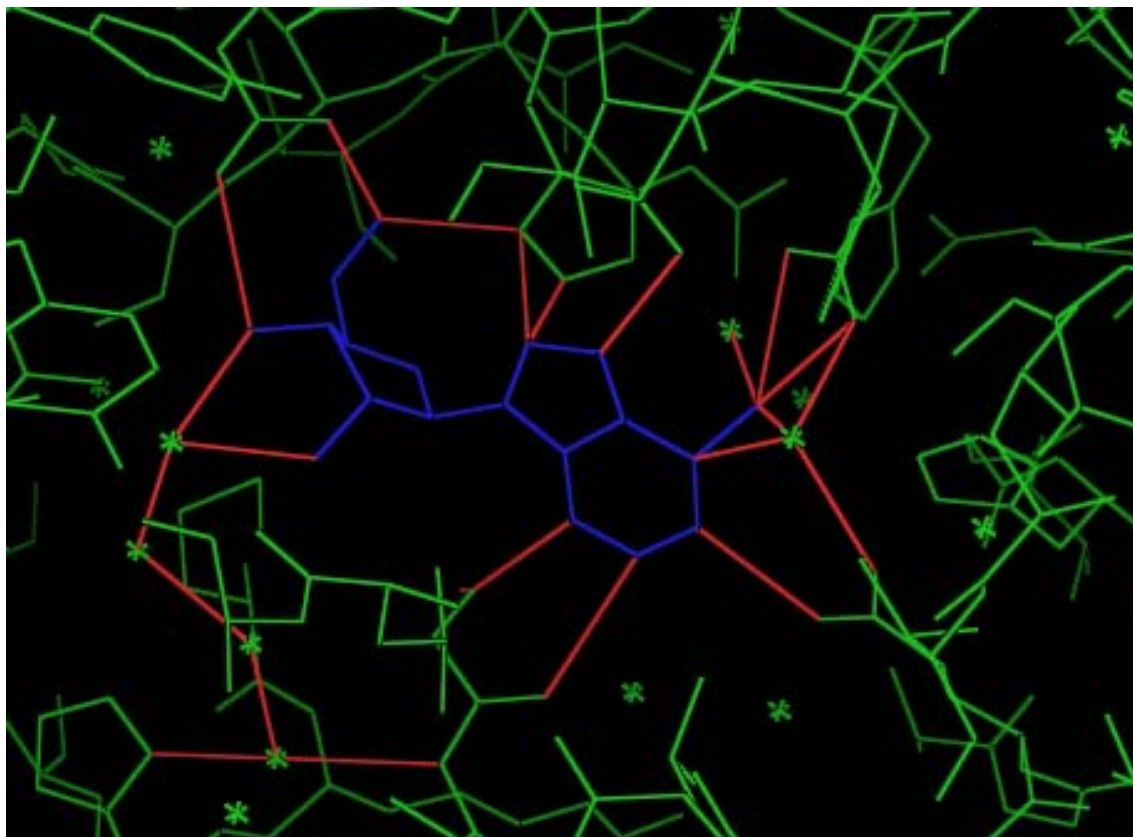


Figure 4.1. Inter-atomic interactions at the protein-ligand interface, within a distance cutoff 3.15 Å. Protein is “adenosine deaminase”, and ligand is “PRH” in the “1a4m” PDB complex.

Application of Frequent Subgraph Mining to Identifying Frequent Atomic Interaction Patterns at the Protein-Ligand Interface.

The interfacial graphs were generated for all 860 protein-ligand complexes in our internal training set. Then, the Fast Frequent Subgraph Mining (FFSM) program developed in one of our laboratories (Huan, J., Wang, W., and Prins, J., 2003) was used to identify the set of subgraphs that occurs in at least a certain fraction (called support value) of these graphs. The FFSM has an advantage over existing similar algorithms of being both fast and robust; this advantage is based on efficient subgraph enumeration operations, in addition to an algebraic graph framework developed to reduce the number of redundant candidates proposed. The details of the FFSM algorithm and its applications to the analysis of small molecules have been described earlier in Chapter 2 of this dissertation in the computational methods section. Another earlier application of the FFSM algorithm to the analysis of protein graph families have also been described elsewhere (Huan, J., Wang, W., Washington, A., Prins, J., Shah, R., and Tropsha, A., 2004).

In this study, FFSM was used with a support value of ~5%. Among these frequent subgraphs, we were interested only in those that contain both the ligand and protein atom-vertices, i.e. frequent subgraphs that are composed of only protein atoms or only ligand atoms were eliminated. Furthermore we were only interested in *closed* subgraphs. In graph mining, a subgraph is considered closed if it has no super-graph (or parent graph) with the same support in the dataset. On the other hand, if a subgraph has the same support as its super-graph (i.e., it occurs in the same place every time its super-graph occurs) then it is not considered closed and consequently eliminated from the consideration. Please refer to the computational methods section in Chapter 2 for more details about identifying closed

subgraphs. The closed subgraphs resulting from this analysis naturally correspond to frequent patterns of inter-atomic interactions. For each of these patterns we have stored both its internal geometric coordinates and its frequency of occurrence in the internal training set of protein ligand complexes (see **Figure 4.2**). These were used in developing the scoring function as described below.

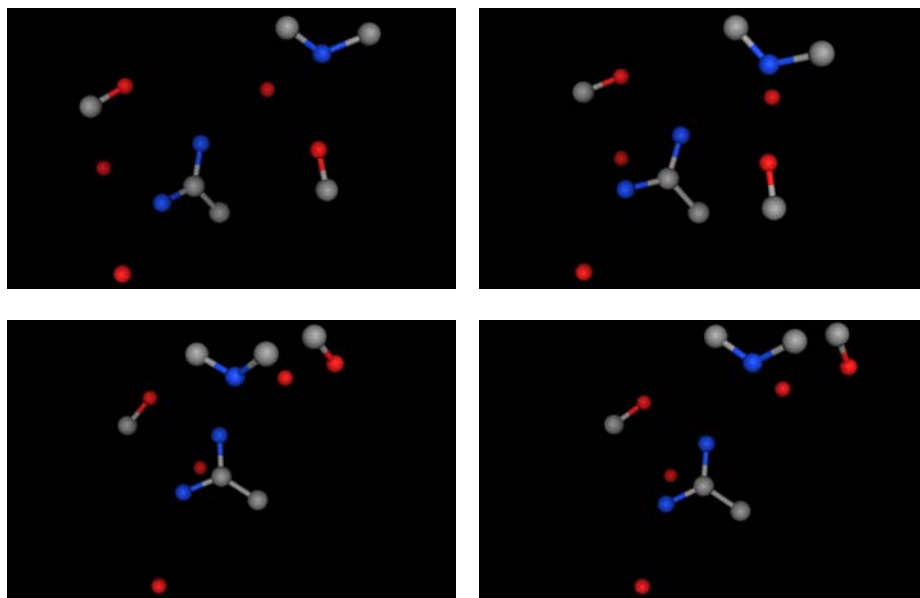


Figure 4.2. Example of 4 different geometries for an interaction pattern between protein and ligand atoms as well as water molecules.

Deriving the Scoring Function Using Frequent Protein-Ligand Interaction Patterns

The first step in calculating the score for a given pose of a protein-ligand complex is identifying the interface in that complex. As discussed above, we define the interface as interactions formed by the protein and ligand atoms that are within 3.5 Å distance cut off; and if a water molecule is found, it is also considered part of the interface in addition to any atom within the same cut off distance of 3.5 Å from the water molecule. Then, we check how many frequent patterns found within the internal training set complexes can be also found at the interface and used in scoring the protein-ligand complex's pose.

We generally assume that the higher the number of frequent “classical” patterns found at the interface of a protein-ligand pose and the more frequent these classical patterns are, the higher this pose should be scored. We also realize that the better the superimposition (i.e. the smaller the Root Mean Square Deviation, RMSD) between the pose pattern and the corresponding “classical” pattern, the higher the score should be as well. Finally, we suppose that the score should be influenced by the size of the frequent pattern identified for the pose, i.e., the score should be higher for bigger patterns. Taking these considerations into account, we derive the following formula to score a protein-ligand complex,

$$\text{Score} = \sum_i^N \sum_j^M |P_i| / (RMSD_{ij} + \epsilon) \quad (1)$$

Where N is the total number of frequent (“classical”) patterns found at the interface, M is the frequency of the pattern i, and therefore is the number of modes of interaction (number of different internal geometric coordinate sets) for that pattern, $|P_i|$ is the size of

the pattern P_i (number of protein and ligand atoms in the pattern), and $RMSD_{ij}$ is calculated for the superimposed pattern P_i with the mode of interaction j at the interface.

The first summation is over all classical patterns that are found at the interface. The second summation reflects the frequency of each pattern and the different modes of interaction for each pattern. An RMSD cut off value of 1.0 °A is used to decide if the pattern should be included in the scoring function or not. This cut off value defines the applicability domain of our knowledge-based scoring function as we will explain later. Also, to avoid dividing by zero, an epsilon (ϵ) value of 1×10^{-60} is added to the RMSD. This value is chosen based on the smallest RMSD value that was found during the study such that it would not affect the final score.

Validation of the Scoring Function

In order to validate the method and test the ability of the scoring function to accurately identify the native pose (as determined by x-ray) among those deviating from the native structure (i.e., generated computationally), a set of experiments was designed. The basic idea behind the experimental design was to simulate the realities of virtual screening when the scoring functions derived from the analysis of known protein-ligand complexes should be used to predict the binding mode ligands in advance of the experimental investigation.

In the first set of experiments, a large database of protein-ligand complexes was divided into two datasets with 1:4 dataset size ratios. The larger dataset (the “internal training dataset”) was used to extract the frequent patterns and their modes of interaction, and the smaller dataset (the “external test set”) had its ligands removed from each protein-ligand complex, and then docked back using available computer programs to generate

various non-native poses. The challenge was to use the knowledge derived from the large dataset (represented by the frequent patterns) to score and accurately identify the correct (native) pose among other non-native poses for the external test set complexes. We were also interested in comparing the performance of our scoring function vs. other commonly used scoring functions. The selection of internal training and external test sets was discussed in Methods.

The non-native (decoy) structures for each protein-ligand complex in the external test set were generated as follows:

1. Each ligand in the protein-ligand complex was processed by Omega (OpenEye Scientific Software, Inc) to produce up to 1,000 conformations.
2. Each conformation was docked into its original protein using FRED (OpenEye Scientific Software, Inc) to produce no more than 1,000 poses. Only one pose with the best score using one of FRED's internal scoring functions was selected.
3. Thus, the number of ligand conformations determined the maximum number of non-native poses that could be generated; that is, no more than 1,000 poses were generated as non-native poses.

The scoring function described in formula (1) as well as six scoring functions provided by FRED were used to score each of the native and non-native poses for each protein-ligand complex. Therefore, for each scoring function, a rank for the native pose was given based on its score relative to the scores of the non-native poses for each protein-ligand complex. Ideally, each scoring function should rank the native pose as number one on top of all non-native structures for each complex. Since the number of non-native poses varies from one protein-ligand complex to another, a percentage value (in addition to the

absolute value) of the rank of native pose was given for each protein-ligand complex. The average of the ranks for all protein-ligand complexes illustrates the ability of each scoring function to identify native among non-native poses. It also affords the comparison between different scoring functions. The following six scoring functions were provided by FRED:

1. Shapegauss. (McGann, M., Almond, H., Nicholls, A., Grant, J., and Brown, F., 2003).
2. PLP. (Verkivker et al., 2000).
3. Chemgauss. (Developed at OpenEye Inc.).
4. Chemscore. (Eldridge et al. 1997).
5. Screenscore. (Stahl, M., and Rarey, M., 2001).
6. Consensus score: the score that results from equal contribution of all the five scores above.

Figure 4.3 summarizes the workflow of the first experiment designed to validate our scoring function.

To verify the robustness of the method and the *pose* scoring function, two more sets of experiments were designed in addition to the first one. The two additional experiments have the same steps but different criteria for splitting the internal training and external test set. In the second experiment, the internal/external datasets switched places, i.e., core set became the internal training set, and remaining part of the refined set became the external test set. In the third experiment, 860 complexes were randomly selected as an internal training set, and the 230 remaining complexes were used as a external test set, such that the internal training set has completely different protein families than those in the external test set. Results of the three experiments were found satisfactory as we shall discuss below.

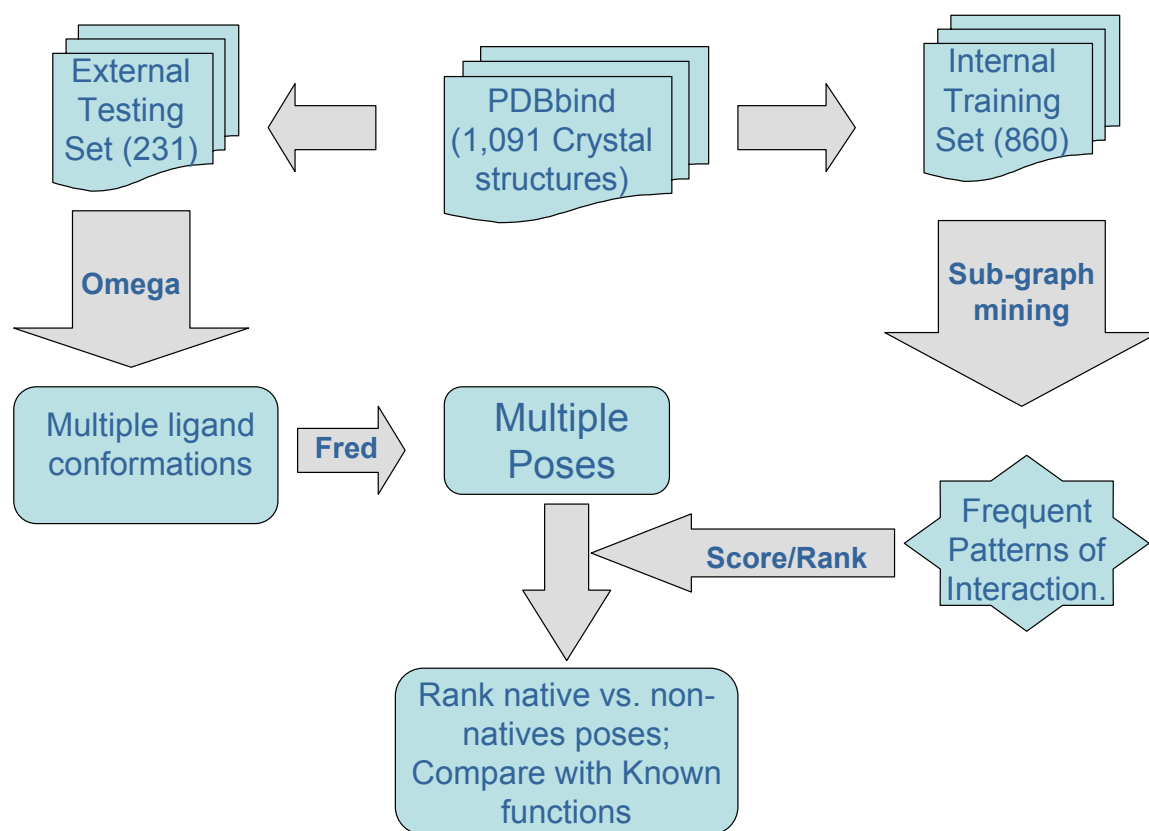


Figure 4.3. Work flow for the validation of the method.

RESULTS AND DISCUSSIONS

Identification of “classical” interaction patterns in the internal training set and external test set scoring.

Graphs for the internal training set of 860 protein ligand complexes were generated as described in Methods. As many as 1732 interfacial graphs have been obtained by applying contact distance threshold of 3.5 Å, which implies that on average we identify nearly two graphs in each complex. The application of FFSM method to identify frequent subgraphs that occur in at least 5% of the graphs (i.e., support value of 5%) resulted in 70,204 frequent subgraphs, among which, 22,584 closed frequent subgraphs were identified. These subgraphs correspond to frequent atomic interaction patterns at the interface of protein ligand complexes in the internal training set. For every pattern we have recorded its frequency (i.e., the number of occurrences in the internal training set) as well as its geometric internal coordinates. The size of the patterns ranged from 4 to 13 atoms (or nodes) with an average of 9 atoms (or nodes).

The external test set used for this experiment included 230 protein-ligand complexes. As explained above, up to 1,000 non-native poses were generated for each protein-ligand complex in the external test set; on average, we have generated 256 poses for each ligand. Each pose including the native one was scored using formula (1) and then, the rank of the native pose among all poses was identified using our scoring function vs. six scoring functions included with the FRED software. Since the total number of poses generated computationally (using Omega and FRED) was different for different protein ligand complexes we expressed the rank order of the native structure as percent value (i.e., for a protein-ligand complex, the rank order of the native pose was divided by the total

number of poses generated for that complex and then multiplied by 100%) instead of expressing the rank order as an absolute value. So, for each protein-ligand complex in the external test set, the percentage rank order of the native pose among all other non-native poses was calculated. Then, the average of this percentage over the entire external test set of protein-ligand complexes was calculated.

Figure 4.4 shows the results of this calculation for each scoring function for comparison purposes. The figure shows two scenarios for scoring using our FP-score scoring function: first scenario (one before last bar), is when all patterns found at the interface are used in scoring. In the second scenario (last bar), only geometrically (not only compositionally) conserved patterns with RMSD value less than 1.0 °Å with respect to “classical” patterns are used in scoring. Obviously, using the geometrical similarity cut off for the patterns used in scoring afforded much better results. In essence the geometrical similarity cut-off imposes a limitation on the applicability of internal training set patterns in scoring the external test set poses and therefore can be regarded as the applicability domain of the scoring function derived from the internal training set: external test set patterns that do not share the geometrical similarity to the internal training set patterns (with similar composition) are excluded from scoring.

Figure 4.4 clearly demonstrates that the scoring function developed in this study (FP-Score) outperforms all six alternative functions as provided by FRED in discriminating the native vs. alternative poses. Thus, the average percentage rank for the native pose using FP-Score is 9.3% compared to 18.6% for the consensus score (which is the best performing scoring function among all those included with FRED).

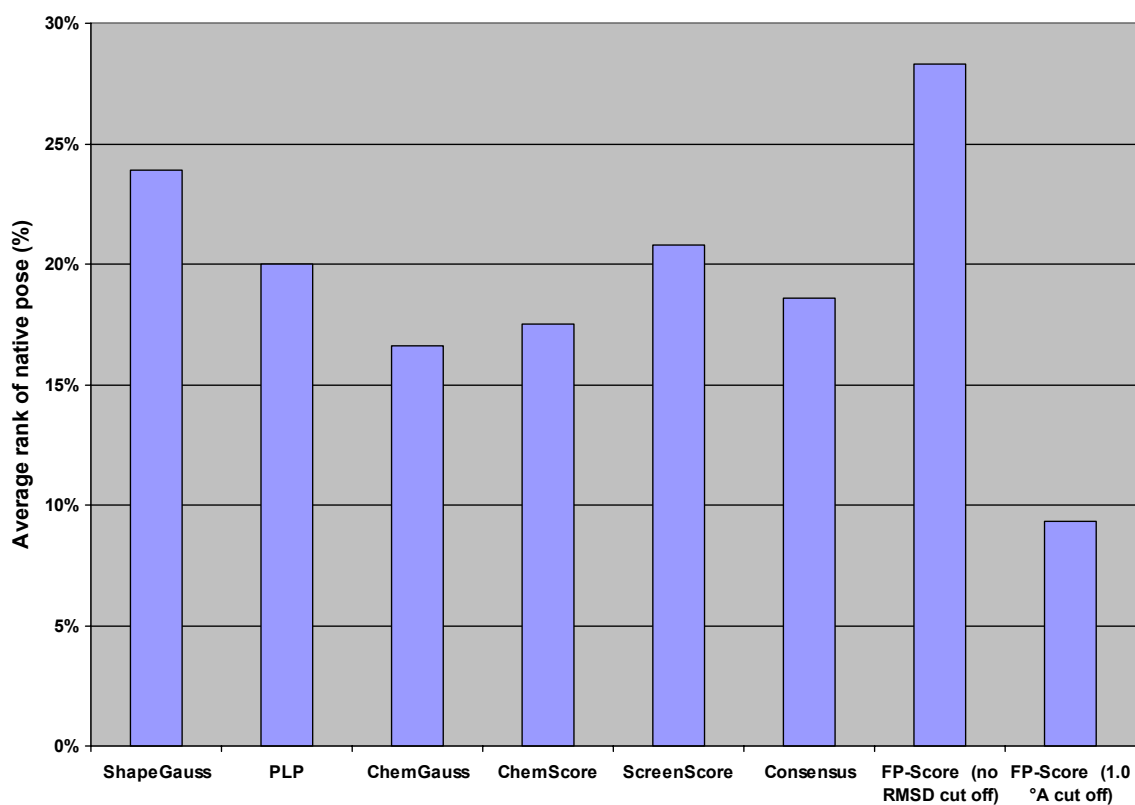


Figure 4.4. Comparison between scoring functions using 231 (core set) as external testing set, and the remaining 860 as internal training set.

To further demonstrate the FR-Score's ability to rank the native pose, as well as the pose with the smallest RMSD (i.e., the one that is closest to the native pose), **Figure 4.5** shows the number of protein complexes in the external test set (in percentage) that has its native pose ranking as top 1, 2, 3, 4, 5, and 6 or more. A comparison with the consensus score shows that the FP-score ranks the native as number one on top of all poses in 50% of the cases, compared to 32% using the consensus score. The figure also shows that using the FP-score, native pose ranked in the top 3 in 97% of the complexes in the external test set, compared to 75% using the consensus score.

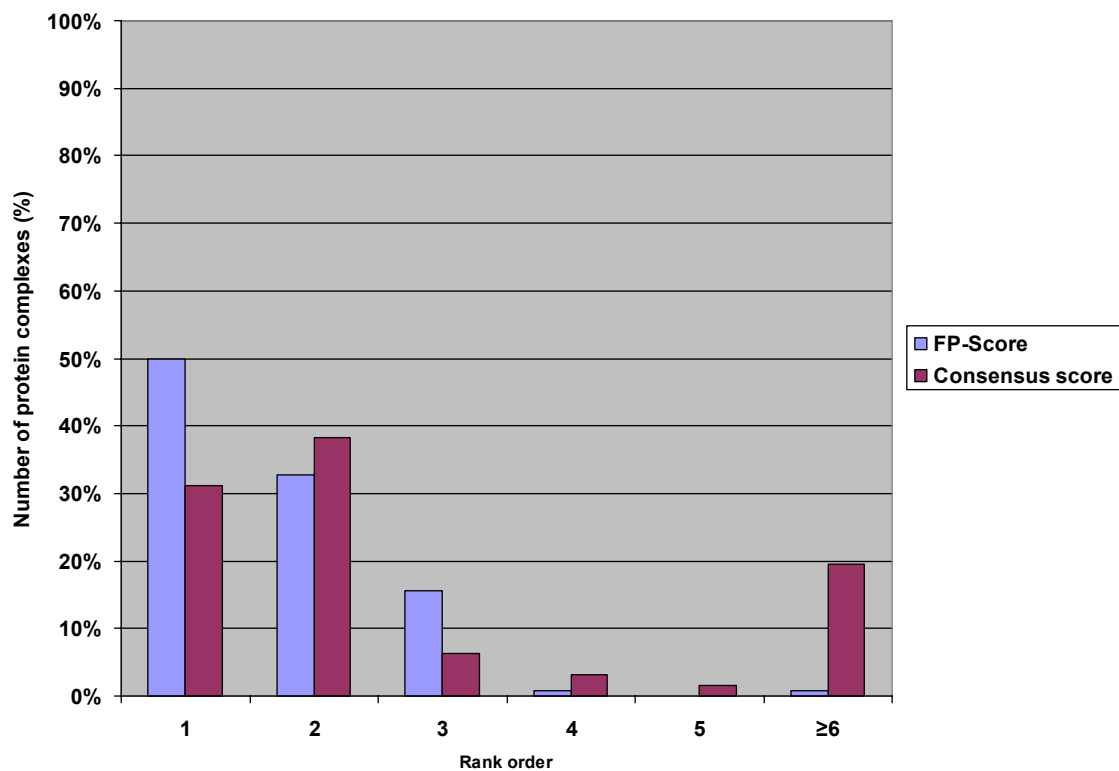


Figure 4.5. The number of protein complexes in the external test set as a function of the rank order of the native pose for these complexes.

In **Figure 4.6**, the same analogy was applied for the pose with the smallest RMSD, i.e., the pose that is the closest to the native structure. Notice that even though the consensus score out performed the FP-score in ranking the pose with the smallest RMSD on top of all poses (38% for the consensus compared to 20% for the FP-score). However, overall, the pose with the smallest RMSD ranked in the top 4 in 95% of the cases using the FP-score compare to 60% using the consensus.

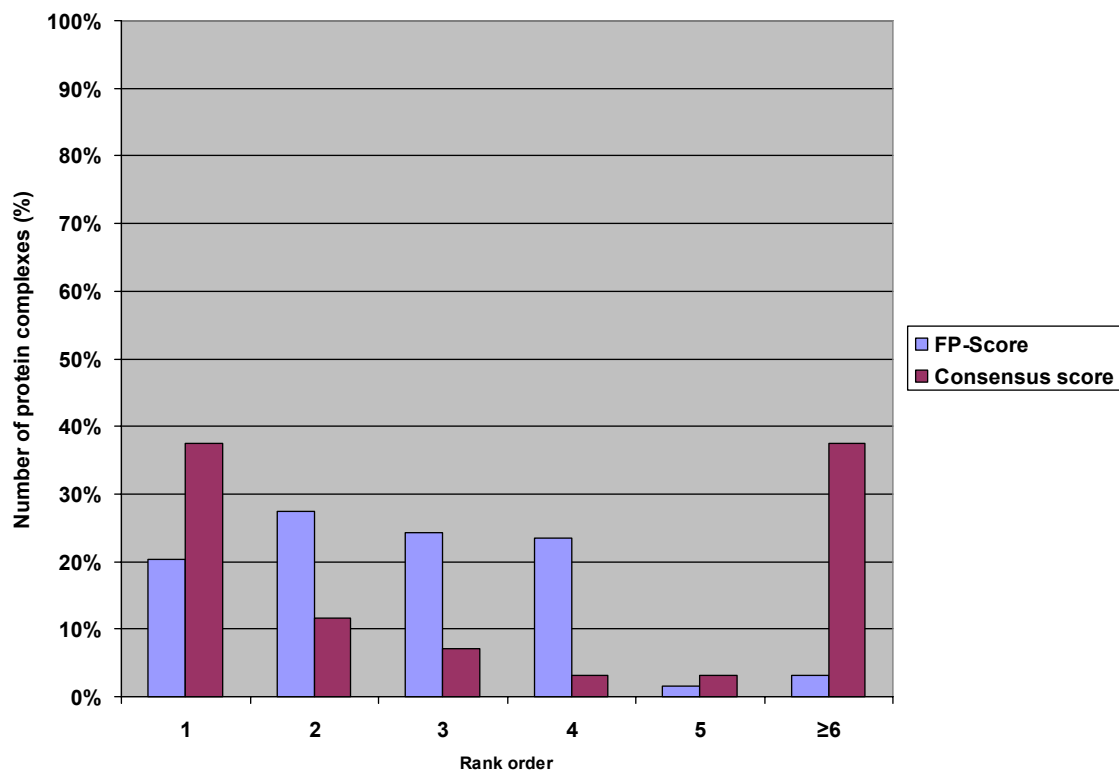


Figure 4.6. The number of protein complexes in the external test set as a function of the rank order of the pose with the smallest RMSD for these complexes.

Figure 4.7 provides an example for one of the protein-ligand complexes in the internal training set, the “1nc3”. As the figure shows, the native pose ranked number one on top of all other poses using the FP-score. This was observed in 50% of the protein-ligand complexes in the external test set (as also shown in **Figure 4.5**). The figure also shows that the pose with the closest RMSD value (0.34 °A in this particular case) ranked as second after the native structure. Finally, since most of the remaining poses were out of the applicability domain, they ranked as third and fourth after the first two poses.

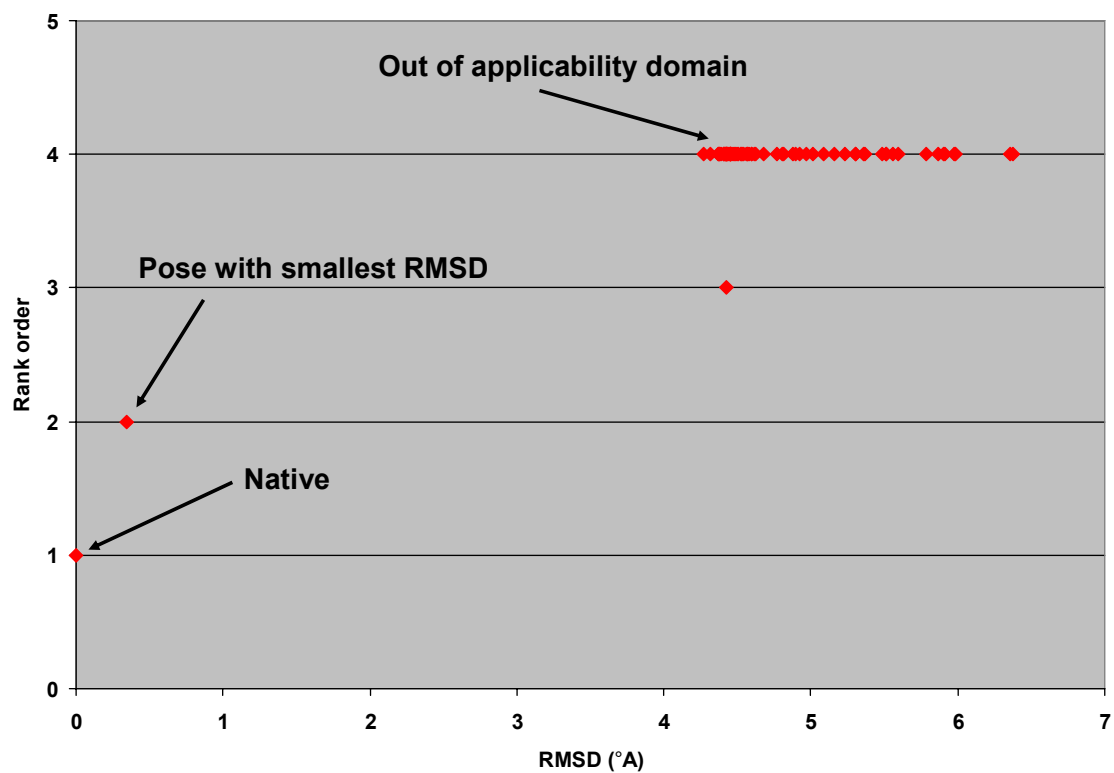


Figure 4.7. Rank order as function of RMSD for the protein-ligand complex "1nc3".

Applying more stringent external test: switching the internal training and external test sets.

To subject our scoring function to a more challenging examination we have inverted the definition of the internal and external sets. Thus, a much smaller group of 231 protein-ligand complexes was now used to derive “classical” patterns using practically the same criteria for frequent subgraph mining as in the previous exercise; the only difference was that we used a support value of 4% instead of 5% as in the previous case. Our analysis identified as many as 422 interfacial graphs, i.e., again almost two such graphs per protein ligand complex, on average. The total number of closed frequent subgraphs was 25,057, and the size of the patterns ranged from 4 to 13 and averaged on 8 atoms (or nodes). In this case we generated up to 500 non-native poses for the external test set of 860 complexes with the average of 174 poses for every external test protein-ligand complex. The poses were scored as before using the same set of scoring functions; based on our previous experience we have used the pattern geometrical similarity cut-off of 1.0 Å RMSD.

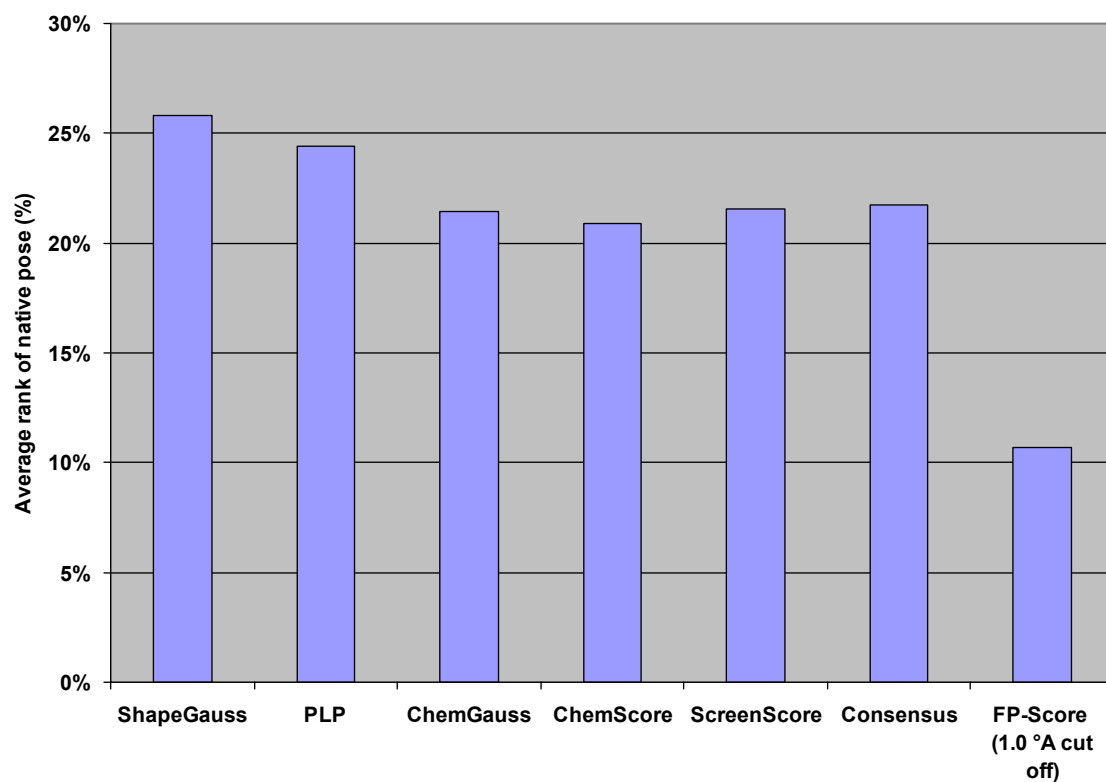


Figure 4.8. Comparison between scoring functions using 231 (core set) as internal training set, and the remaining 860 as external test set.

The results of our experiments are shown in **Figure 4.8**. Somewhat surprisingly, the average rank of the native structure using FP-Score remained practically the same as in the previous experiment, i.e., 10% in spite of using a much smaller internal training set. The consensus FRED score ranked the native structure at 21.7%. This experiment shows that even a small subset of the entire original dataset of 1,091 complexes provides sufficiently representative set of interacting atomic patterns at the protein-ligand interface to allow for accurate scoring of the native ligand pose for a much larger protein ligand external test set.

In the third experiment, we placed several protein families in the external test set that were completely different from those in the internal training set. 860 protein-ligand complexes were selected for the internal training set leaving 231 complexes for the external test set. A total of 1712 interfacial graphs were found, and when using a support value of 6%, 4,809 closed subgraphs were found frequent and used as classical patterns of interaction. For the 231 complexes in the external test set, no more than 500 non-native poses were generated; the average number of poses was 120. The size of these patterns ranged from 4 to 13 with an average of 8 atoms (nodes). **Figure 4.9** shows the results of this experiment.

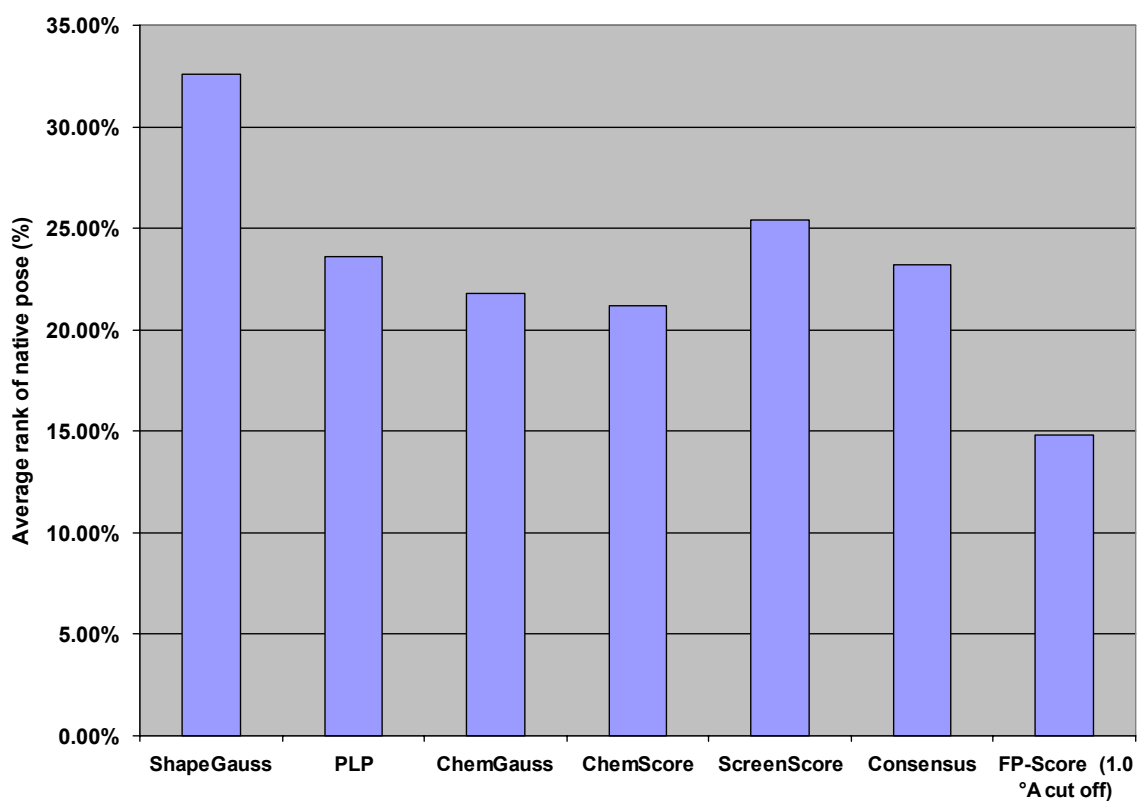


Figure 4.9. Comparison between scoring functions using 860 complexes as internal training set, and the remaining 231 as external test set, where sets have different families.

The same RMSD cut off value of 1.0 °Å was used. FP-Score ranked the native pose as the top 14.8% on average compared to 23.2% using the consensus score. The reason behind this experiment was to see if patterns derived for the internal training set were dependent on the protein families used in the derivation process. Ideally, the “classical” patterns of interaction should be dependent only on the atom types and the contact geometry regardless of the protein family. The results of this experiment certainly agree with this expectation. Nevertheless, when representatives from the same protein families were included in both internal training and external test sets the results were slightly better (cf. **Figures 4.4** and **4.8**), which is apparently due to the fact that the internal training and external test sets complexes had generally more frequent patterns in common..

CONCLUSIONS

In this paper we described a novel approach to scoring ligand poses that are generated in multiple docking experiments. The method is based on a simple principle of comparing the atomic interaction patterns identified at the interface of a external test protein-ligand complex with those found frequent (“classical” patterns) in x-ray characterized protein-ligand complexes of the internal training set. In order to identify the “classical” patterns we have used advanced methods of frequent subgraph mining applied to the unique representation of protein ligand interfaces as chemical unidirectional graphs with the nodes corresponding to individual atoms and edges defined by physical proximity of atom-nodes. Several validation experiments described in this paper have confirmed that the frequent “classical” geometric and chemical patterns of interaction are robust, i.e., they can be identified as universal even within a relatively small set of reasonably diverse protein-ligand complexes. We have demonstrated that a simple scoring function based on

chemical and geometrical similarity between the external test complex-specific interaction patterns and the “classical” patterns could fairly accurately distinguish between native and decoy poses. Furthermore, the additional tests indicated that our FP scoring function identifies the most geometrically native-like pose within top four best scoring poses in 95% of external test protein-ligand complexes. We have shown that the FP scoring function demonstrated higher accuracy in both distinguishing the native pose vs. decoys and in identifying most native like poses than several alternative scoring functions available commercially. Unlike the competing functions, the FP scoring function is very simple; it does not consider any solvation or entropy effects, or the active site (or the ligand) protonation state since only heavy atoms are included in the study and hydrogen atoms are disregarded. What makes this function particularly unique is that our fragments are not limited to some number of chemical groups defined *a priori*; instead, new patterns can always be derived and analyzed as long as they occur frequently in the dataset.

As with any empirical scoring function, it was important to define the applicability domain (using RMSD cut off values) for the interaction patterns. The applicability domain is naturally dependent on the dataset used to derive the “classical” patterns. It restricts the conformational flexibility of fragments that can be considered similar to the “classical” ones. Our studies have demonstrated that the use of the applicability domain has significantly improved the accuracy of scoring.

This FP scoring function has provided efficient way to identify the correct binding modes for protein-ligand complexes. We expect that it will be widely used to improve the accuracy of modern docking approaches. We plan to expand upon this pilot study by implementing most efficient frequent subgraph mining approaches as well as looking into

different ways of defining atom types. We shall expect further improvement in the accuracy of the FP scoring function with the continuing growth in the number of x-ray characterized protein ligand complexes stored in such databases as PDDBind (Wang et al., 2005) and MoAD (Carlson et al., 2005). Finally, we plan to extend the use of frequent protein-ligand interaction patterns these patterns towards other structure-based design approaches such as de novo design, receptor-based pharmacophore modeling, and bioisosteric replacements.

ACKNOWLEDGMENTS

We thank Prof. Jun (Luke) Huan (Department of Electrical Engineering and Computer Science, University of Kansas) for his pioneering and continuing development of the FFSM method and software. We are also grateful to OpenEye for making the Omega and FRED software available and to Prof. S. Wang and his group at the University of Michigan for the development of the PDDBind database and making it available. And we would like to acknowledge the NIH for grant funding this study.

CHAPTER 5

SUMMARY AND FUTURE DIRECTIONS

Discovering new drugs is a long and expensive process. The challenge is to reduce both the cost and the time without compromising the efficacy of designed drugs. Computer-Assisted Drug Design (CADD) approaches help medicinal chemists prioritize synthesis and testing of compounds that are likely to be active. CADD techniques have been used successfully to improve the efficiency of the drug discovery process. The combination of computational chemistry concepts, robust software, and high-end computer hardware are used to assist the medicinal chemists identifying or designing ligands that are more likely to interact with the receptor of interest. The main objective of this research is to develop novel effective CADD approaches.

CADD methods can be categorized based on the availability of the three-dimensional (3D) structure of the target protein. Ligand-based drug design methods are used if the structure of the target protein is not known. A commonly used method is the Quantitative Structure-Activity/Property Relationship (QSAR/QSPR) approach. It generates molecular descriptors for all ligands with known target property (i.e. biological activity, toxicity) and uses them in combination with multivariate statistical modeling techniques to arrive at predictive activity or property models. The success of this approach relies on the robustness of the molecular descriptors used, as well as the strength of the statistical technique used to build the predictive models. Most currently available molecular descriptors either lack the mechanistic interpretability or are limited by the pre-defined set

of chemical fragments that are used in chemotyping of any dataset of interest. The current limitations of molecular descriptors used in modern QSAR and cheminformatics research underline the significance of this research that is focused on developing dataset-specific descriptors based on the frequent sub-structures in the dataset. These frequent sub-structures will be identified using the graph representation of molecules and the sub-graph mining approach, as we shall explain later. The medicinal chemist can easily interpret these descriptors. In addition, new important fragments that might have not been defined a priori can be discovered. The research question that needs to be answered in the course of this project is whether these descriptors can indeed give a better predictive QSAR model as compared to those generated with current descriptors.

Another ligand-based drug design method is the Active Analog Approach. It is used to explore active compounds that bind to same target protein in order to identify “pharmacophoric” groups responsible for the specific activity; these groups are subsequently used to screen chemical databases for new leads. In this research, we tried to answer the question whether the frequent sub-structures can be used as novel means to identify the pharmacophoric groups and examine their ability to identify new leads in the context of the Active Analog Approach. The significance of this particular study rests on the fast identification of the pharmacophoric groups for database mining. The advantage of our proposed approach is that it does not rely on 3D conformational search of the structures and therefore it is highly efficient computationally. In addition to identifying the pharmacophoric groups, toxicophores and fragments responsible for mutagenicity have also been addressed in this research and can be helpful for doing safety predictions on molecules before synthesizing them.

If the 3D structure of the target protein is available then structure-based drug design methods are used. The most common one and a widely used method is the computational “docking”. Here, a database of compounds is screened to identify compounds that can fit into the active site of the target protein. This approach has been widely used in hit identification and lead optimization. However, there remain significant challenges in the application of this approach, in particular in relation to current scoring schemes. With the exponential increase in the number of protein-ligand crystal structures in the protein databank (PDB), researchers are more interested in exploring the information that can be gathered from these structures. In this research, we tried to answer the question whether the frequent chemical subgraphs at the protein-ligand interface can be used in devising novel accurate scoring functions and docking protocols as compared to current schemes. The study can be highly significant and of interest to many researchers in that field.

SUMMARY AND FUTURE DIRECTIONS OF CHAPTER 2

Computational QSAR modeling is fundamentally based on the similarity principle, which states that “similar compounds have similar biological properties”. Consequently one can predict the biological target property of a molecule from that of chemically similar compounds for which the property is known. However, to build valid quantitatively predictive models a similarity metric is required; therefore a unit of measurement such as molecular descriptors needs to be identified. Once the descriptors are defined, QSAR techniques can be used to relate the chemical structure of a molecule to its target property.

In this study, we presented an approach to generate fragment-based molecular descriptors. Unlike molecular descriptors based on physicochemical properties and distances of atoms in the molecule, fragment-based descriptors could potentially provide a

mechanistic explanation of the dependence of the target property on molecular structure. Such explanation especially with respect to the differences between active and inactive molecules could provide useful guidance to medicinal chemists with respect to rational design of new biologically active chemical entities.

A common trait to other fragment-based chemical descriptors is that fragments are identified a priori; thus frequently the total number of such descriptors generated for a molecular dataset is exceedingly large (e.g., hundreds or thousands fingerprints are generated typically) and/or fragment descriptors are generic. This makes it difficult to build robust and statistically predictive QSAR models that uniquely describe the relationship between structure and activity of specific datasets such that the derived QSAR models could successfully identify novel unique computational hits.

In our approach, we use a labeled chemical graph representation of molecules and employ Fast Frequent Subgraph Mining (FFSM) method developed in our group. Our fragment-based descriptors are derived based on frequent common substructures that are found in at least a subset of molecules (this fraction is defined as a *support value*) in the dataset. This is followed by removing the smaller substructures correlated with their parents leaving only what is called the closed substructures. Once these frequent closed substructures are identified, the count of each substructure in each molecule in the dataset is calculated; thus each frequent common substructure serves as a chemical descriptor type and the frequency becomes a descriptor value. This representation affords the application of conventional QSAR modeling techniques to any chemical dataset with measured biological activity leading to a novel fragment descriptor based QSAR modeling approach.

The counts of each frequent fragment have been used as descriptors in variable selection k Nearest Neighbor (kNN) QSAR modeling. Highly predictive models have been generated for the datasets used in this study, and were comparable to MolConnZ descriptors, which is one of the commonly used molecular descriptors. Frequent subgraphs implicated in validated models can afford mechanistic interpretation of the results that are easily understood by medicinal chemists in terms of essential pharmacophoric or toxicophoric elements responsible for the molecule activity, as we demonstrated in Chapter 3 of this dissertation using another classification method that can provide a better way of interpreting the selected descriptors than kNN does. Also, since these fragment-based descriptors are dataset-derived and not predefined, this will open the door to finding new sub-structures that are not defined *a priori*. In addition, they are dataset-specific, and therefore provide a better definition of the model applicability domain than *a priori* defined fragments.

In the future, we would like to improve the way we select the frequent substructures specially that the number of these substructures increases quickly with the reduction in the support value. In addition, we will look for a better QSAR modeling techniques that can best utilize these fragment-based descriptors and their counts to optimize the prediction ability of the whole process.

SUMMARY AND FUTURE DIRECTIONS OF CHAPTER 3

Having the ideal descriptors by itself is not enough to do QSAR predictions. The descriptors should be combined with the appropriate modeling technique to provide the best prediction. Based on the nature of the molecular descriptors, one modeling technique might perform better than another. In this study we will describe a unique methodology that

is used with the fragment-based descriptors we identify. This methodology should provide a better interpretation to the models generated than kNN does.

As explained earlier in Chapter 2, our fragment-based descriptors are derived based on frequent common substructures that are found in at least a subset of molecules (this fraction is defined as a *support value*) in the dataset. Once these frequent substructures are identified, the occurrence of each substructure in each molecule in the dataset is calculated; thus each frequent common substructure serves as a chemical descriptor type and the occurrence becomes a binary descriptor value. In addition, a modeling methodology is developed based on identifying frequently associated chemical fragments responsible for producing the desired class (activity or toxicity) of the molecules studied. These associated fragments are used as rules (Class Association Rules, or simply CARs) that are characterized by confidence and support values. These CARs can then be used to build a classifier for predicting an external dataset of molecules.

As the results show, the fragment-based descriptors can perform at least as good as the fingerprint descriptors, and in some cases performed better. The descriptors are further utilized by a methodology that takes care of the combined effect of these fragments in predicting the target property of interest, such as activity to a certain receptor, toxicity or mutagenicity. Medicinal chemists can use these descriptors along with the methodology to identify important fragments for future predictions, especially since that these descriptors are easily interpretable and understood by any medicinal chemist.

The strength of the descriptors comes from the fact that it can not miss an important chemical fragment in a dataset. Since defining all possible combination of atoms will give an exponentially large number of chemical fragments, these descriptors can identify the

fragments related only to that particular dataset of interest. The methodology used to handle these descriptors (generating class association rules) provides a high chance of predicting external sets with easily interpretable rules to the medicinal chemist. What we see as the weakness of this method is the fact that fragments that are interchangeable (i.e., have the same physicochemical or pharmacophoric characteristic) will not be treated equally, and therefore, unless each of these interchangeable fragments (also known as bioisosters) occur frequently enough in the dataset to be used as descriptors, it won't be taken into account when doing the predictions. Also, if you were to explore an external database of compounds looking for potential leads, unless all important fragments are already discovered in your internal training dataset, you will not be able to come up with a lead with fragments different from what you already have in your dataset. This was clearly demonstrated in the PGP dataset.

To solve this problem, in the future, we would like to use a database of bioisosters for the fragments defined, and therefore expand the applicability domain of these descriptors and therefore their ability to identify new leads. Implementing the bioisosteric replacement concept can be a potential improvement to the method and will aid in discovering new leads that can potentially be active.

SUMMARY AND FUTURE DIRECTIONS OF CHAPTER 4

Many docking and scoring approaches have been developed over the years in the context of structure based drug design. However, there remain significant challenges in both developing scoring functions that can identify ligands that bind to the active site within a large library of chemical compounds as well as accurately identify the correct binding pose. Many scoring functions have been reported in the scientific literature, and it

has been shown that most scoring functions perform poorly in identifying the correct pose accurately. It has become a common approach to separate the docking and scoring, i.e., generate several alternative binding poses using available docking algorithms and then rank poses using independent scoring functions or consensus scoring. In our studies, we have focused on the problem of identifying the correct binding pose. To this end, we have developed a novel knowledge-based scoring function termed Frequent Patterns-based Score, or simply FP-Score that can identify efficiently the correct (native or geometrically closest to the native) pose among many poses (decoys) for a given protein-ligand complex. The FP scoring function is derived based on frequent geometric and chemical patterns of inter-atomic interactions at the interface of a representative dataset of x-ray characterized protein-ligand complexes.

The approach includes the following steps. First, the protein-ligand interface of each complex in the internal training set is represented by labeled chemical graph where nodes are atoms labeled by atom chemotypes and edges connect protein and ligand atoms within certain distance of each other. Second, frequent common subgraph mining techniques are used to find frequent subgraphs (i.e., interacting atomic patterns) that occur in no less than a certain percentage of the complexes in the internal training set. These frequent subgraphs are considered as “classical” interaction patterns, which are used in scoring each pose in a given protein ligand complex to determine its “native-pose-likeness” as follows. For each pose produced for a ligand in the protein binding site by a computational docking program, patterns of interaction are identified at the protein-ligand interface. These patterns are then analyzed for their both chemical graph and geometrical similarity to “classical” templates. The score for each pose is calculated based on the

number of classical interaction subgraph patterns, their frequency of occurrence in the internal training set, and their similarity to the classical patterns in terms of RMSD. Higher geometric similarity and frequency are associated with a better score.

For our studies we used a set of 1091 protein-ligand complexes in the PDBbind databaset that was divided into internal training (860 complexes) and external test (231) sets. Classical patterns were derived for the internal training set and used to score multiple docking poses that were generated for each protein in the external test set using FRED software from OpenEye. We showed that FP score ranked the native pose as best for 50% of the external test set, and within four top scoring poses for 95% of the external test set proteins. The accuracy of predicting the correct binding pose using FP score was significantly higher than using five commercially available scoring functions (Shapegauss, PLP, Chemgauss, Chemscore, and Screenshot) both independently as well as using their consensus scoring. To the best of our knowledge, the approach described herein is different from all scoring methods described in the literature. The FP function is very simple; it does not consider any solvation or entropic effects and does not take into account the active site (or ligand) protonation state since only heavy atoms are included into the consideration. Furthermore, it is not limited to chemical groups defined a priori; instead, patterns are derived and scored as long as they occur frequently in the experimentally determined dataset of protein-ligand complexes. In addition, FP score goes beyond traditional pairwise scoring of interatomic contacts, i.e., it employs multi-atomic interaction patterns and consequently it takes into account inherently the cooperative effect of interaction between proteins and their ligands.

We suggest that the FP scoring function could be successfully used to refine the lists of poses generated by docking programs. Thus, we expect it to be used widely to improve the accuracy of modern docking/scoring approaches. We plan to expand upon this pilot study by implementing more efficient frequent subgraph mining approaches as well as looking into different ways of defining atom chemotypes. We shall expect further improvement in the accuracy of the FP scoring function with the continuing growth in the number of x-ray characterized protein ligand complexes stored in such databases as PDDBind and MoAD.

The results of this study suggest many uses for frequent chemical and geometric patterns of protein-ligand interaction. Besides the obvious use in scoring the interactions between ligands and proteins, the analysis of frequent interaction patterns could help visualizing the modes of interactions and understanding the mechanisms of interaction. Potentially, knowing the active site atoms and frequent patterns they could participate in one could think of *de novo* design of specific ligands; or at least their fragments that can be then pieced together as is done in several recent approaches (Vajda, S., 2006; Mauser, H., and Stahl, M., 2007)

Another closely related potential use for these classical interaction patterns is the development of the receptor based pharmacophore models, which can be used in traditional pharmacophore based screening. Such structure based pharmacophore generating methods have become popular and successful in recent years (Wolber, G., and Langer, T., 2005). Thus, using nearest neighbor atom patterns in the active site of the protein of interest that participate in frequent “classical” interaction patterns one could deduce the corresponding 3D fragments of the complimentary ligands and use these fragments as pharamacophore

queries against a database of multiple conformations of commercially available chemicals such as ZINC (Irwin, J., and Shoichet, B., 2005).

In addition, these patterns can be used to aid in bioisosteric replacements. This can be done by identifying fragments from the ligand side that share the same type of atoms on the receptor side of the protein. Different fragments which bind to the same region of the protein should in theory be interchangeable. Therefore, several bioisosters can be derived from these patterns and used for lead optimization and drug design purposes.

REFERENCES

- Agrawal, R., and Srikant, R. (1994). Fast algorithms for mining association rules. VLDB-94, 1994.
- Benigni, R. et al. (2000). Quantitative structure-activity relationships of mutagenic and carcinogenic aromatic amines. Chemical Review, 100 (10), 3697-714.
- Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). Classification and regression trees. Belmont: Wadsworth.
- Brooijmans N, Kuntz ID. (2003). Molecular recognition and docking algorithms. Annu Rev Biophys Biomol Struct., 32, 335-73.
- Bruno, I. J.; Cole, J. C.; Lommerse, J. P. M.; Rowland, R. S.; Taylor, R.; and Verdonk, M. L. (1997). IsoStar: A library of information about nonbonded interactions. Journal of Computer-Aided Molecular Design, 11, 525-537.
- Contrera, J., Matthews, E., Kruhlak, N., and Benz, R. (2004). Estimating the safe starting dose in phase I clinical trials and no observed effect level based on QSAR modeling of the human maximum recommended daily dose. Regulatory Toxicology and Pharmacology, 40, 185-205.
- Dehaspe, L., Toivonen, H., and King, R., (1999). Frequent Sub-Structure-Based Approaches for Classifying Chemical Compounds. Proc.of the 8th International Conference on Knowledge Discovery and Data Mining.
- Deshpande, M., Kuramochi, M., Wale, N., and Karypis, J., (2005). Frequent Sub-Structure-Based Approaches for Classifying Chemical Compounds. IEEE Transactions on Knowledge and Data Engineering, 17 (8), 1036-1050.
- FRED. OpenEye Scientific Software, Inc.
- Gohlke, H.; Hendlich, M.; and Klebe, G. (2000). Knowledge-based scoring function to predict protein-ligand interactions. Journal of Molecular Biology, 295, 337-356.
- Golbraikh, A., and Tropsha, A. (2002). Beware of q^2 ! Journal of Molecular Graphics and Modeling, 20 (4), 269-76.
- Golbraikh, A., and Tropsha, A. (2002). Predictive QSAR modeling based on diversity sampling of experimental datasets for the training and test set selection. Journal of Computer Aided Molecular Design, 16(5-6), 357-69.

- Golbraikh, A., Shen, M., and Tropsha, A. (2002). Enrichment: A New Estimator of Classification Accuracy of QSAR Models. Abstracts of papers of the american chemical society 223: 206-COMP Part 1, Apr 7, 2002.
- Han, J., Pei, J., and Yin, Y. (2004). Mining Frequent Patterns without Candidate Generation: A Frequent Pattern-Tree Approach. Data Mining and Knowledge Discovery, 8, 53-87.
- Hendlich, M.; Bergner, A.; Gunther, J.; and Klebe, G. (2003). Relibase: Design and development of a database for comprehensive analysis of protein-ligand interactions. Journal of Molecular Biology, 326, 607-620.
- Holder, L., Cook, D., and Djoko, S. (1994). Substructures discovery in the subdue system. Proc.AAAI'94 Workshop Knowledge Discovery in Databases.
- Huan, J., Prins, J., and Wang, W. (2003). Efficient Mining of Frequent Subgraph in the Presence of Isomorphism. Proc.of the 3rd IEEE International Conference on Data Mining (ICDM), 549-52.
- Huan, J., Wang, W., Washington, A., Prins, J., Shah, R., and Tropsha, A. (2004). Accurate Classification of Protein Structural Families Using Coherent Subgraph Analysis. Pacific Symposium on Biocomputing, 9, 411-422.
- Irwin, J. J. and Shoichet, B. K. (2005). ZINC--a free database of commercially available compounds for virtual screening. J. Chem. Inf. Model., 45, 177-182.
- Ishchenko, A. V. and Shakhnovich, E. I.(2002). Small Molecule Growth 2001 (SMoG2001): an improved knowledge-based scoring function for protein-ligand interactions. J. Med. Chem., 45, 2770-2780.
- Kellogg, G., Kier, L., Gaillard, P., and Hall, L. (1996). The E-State Fields. Application to 3D QSAR. Journal of Computer Aided Molecular Design, 10, 513-520.
- Kitchen, D. B.; Decornez, H.; Furr, J. R.; and Bajorath, J. (2004). Docking and scoring in virtual screening for drug discovery: methods and applications. Nat. Rev. Drug Discov., 3, 935-949.
- Kubinyi, H., Hamprecht, F., and Mietzner, T. (1998). Three-dimensional quantitative similarity-activity relationships (3D QSiAR) from SEAL similarity matrices. Journal of Medicinal Chemistry, 41(14), 2553-64.
- Kuramochi, M. and Karypis, J. (2001). Frequent subgraph discovery. Proc.International Conference on Data Mining'01.
- Labute, P. Probabilistic receptor potential (2001). Chemical Computing Group Journal.<http://www.chemcomp.com/journal/cstat.htm>.

- Liegi Hu, Mark L. Benson, Richard D. Smith, Michael G. Lerner, and Heather A. Carlson (2005). Binding MOAD (Mother of All Databases). Prot. Struct. Func. Bioinformatics, **60**, 333-340.
- Liu, B., Hsu, W., and Ma, Y. (1998). Integrating Classification and Association Rule Mining. Appeared in KDD-98, New York, Aug 27-31, 1998.
- Mark R. McGann, Harold R. Almond, Anthony Nicholls, J. Andrew Grant, and Frank K. Brown (2003). Gaussian Docking Functions, Biopolymers, **68**, 76–90.
- Martin Stahl and Matthias Rarey (2001). "Detailed Analysis of Scoring Functions for Virtual Screening", Journal of Medicinal Chemistry, **44**, 1035-1042.
- Martin, Y. (1981). A Practitioner's perspective of the role of quantitative structure activity analysis in medicinal chemistry. Journal of Medicinal Chemistry, **24**, 229.
- Matthew D. Eldridge, Christopher W. Murray, Timothy R. Auton, Gaia V. Paolini and Roger P. Mee (1997). Empirical scoring functions: I. The development of a fast empirical scoring function to estimate the binding affinity of ligands in receptor complexes, Journal of Computer-Aided Molecular Design, **11**, 425-445.
- Mausser, H., and Stahl, M. (2007). Chemical Fragment Spaces for de novo Design. Journal of Chemical Information and Modeling, ASAP Article.
- Moreno, E. and Leon, K. (2002). Geometric and chemical patterns of interaction in protein-ligand complexes and their application in docking. Proteins-Structure Function and Genetics, **47**, 1-13.
- Muegge, I. and Martin, Y. C. (1999). A general and fast scoring function for protein-ligand interactions: A simplified potential approach. Journal of Medicinal Chemistry, **42**, 791-804.
- Muegge, I. (2006). PMF scoring revisited. Journal of Medicinal Chemistry, **49**, 5895-5902.
- Nobeli, I.; Mitchell, J. B. O.; Alex, A.; and Thornton, J. M. (2001). Evaluation of a knowledge-based potential of mean force for scoring docked protein-ligand complexes. Journal of Computational Chemistry, **22**, 673-688.
- Oloff, S. et al. (2006). Chemometric analysis of ligand receptor complementarity: identifying Complementary Ligands Based on Receptor Information (CoLiBRI). J.Chem.Inf.Model. **46** (2), 844-51.
- Omega. OpenEye Scientific Software, Inc.

- Penzotti, J.; Lamb, M.; Evensen, E.; and Grootenhuis, P. (2002). A computational ensemble pharmacophore model for identifying substrates of P-Glycoprotein. Journal of Medicinal Chemistry, **24**, 1737-1740.
- Quinlan, J. (1992). C4.5: Program for machine learning. Morgan Kaufmann.
- Raymond, J., & Willett, P. (2002). Maximum common subgraph isomorphism algorithms for the matching of chemical structures. Journal of Computer-Aided Molecular Design, **16**, 521-533.
- Renxiao Wang, Yipin Lu, and Shaomeng Wang. Comparative Evaluation of 11 Scoring Functions for Molecular Docking. J. Med. Chem., **46** (12), 2287 -2303, 2003
- Sheridan, R., Rusinko, A., Nilakantan, R., Venkataraghavan, R. (1989). Searching for Pharmacophores in Large Coordinate Data Bases and Its Use in Drug Design. Proceedings of the National Academy of Sciences of the United States of America, **86** (20), 8165-8169.
- Snarey, M. et al. (1997). Comparison of algorithms for dissimilarity-based compound selection. Journal of Molecular Graphics and Modeling, **15** (6), 372-85.
- Trohalaki, S., Gifford, E., and Pachter, R. (2000). Improved QSARs for predictive toxicology of halogenated hydrocarbons. Computational Chemistry, **24** (3-4), 421-27.
- Vajda S, Guarnieri F. (2006). Characterization of protein-ligand interaction sites using experimental and computational methods. Curr Opin Drug Discov Devel, **9**(3):354-62.
- Vajda, S. (2006). Computational Mapping of Proteins for Fragment Based Drug Design. Keystone Symposium on Structure Based Drug Discovery 2006 April, Whistler BC, Canada.
- Valerei, G., Willett, P., & Bradshaw, J. (2003). Similarity searching using reduced graphs. Journal of Chemical Informatics and Computer Science, **43**, 338-345.
- Verkivker, G.; Bouzida, D.; Gehlaar, D.; Rejto, P.; Arthurs, S.; Colson, A.; Freer, S.; Larson, V.; Luty, B.; Marrone, T.; and Rose, P. (2000). Deciphering common failures in molecular docking of ligand-protein complexes. Journal of Computer-Aided Molecular Design, **14**, 731-751.
- Votano, J., Parham, M., Hall, L., Kier, L., Oloff, S., Tropsha, A., Xie, Q., and Tong, W. (2004). Three new consensus QSAR models for the prediction of Ames genotoxicity. Mutagenesis, **19** (5), 365-377.
- Wang, R. X.; Fang, X. L.; Lu, Y. P.; and Wang, S. M. (2004). The PDBbind database: Collection of binding affinities for protein-ligand complexes with known three-dimensional structures. Journal of Medicinal Chemistry, **47**, 2977-2980.

- Wang, R. X.; Fang, X. L.; Lu, Y. P.; Yang, C. Y.; and Wang, S. M. (2005). The PDBbind database: Methodologies and updates. Journal of Medicinal Chemistry, 48, 4111-4119.
- Wolber, G., and Langer, T. (2005). LigandScout: 3-D Pharmacophores Derived from Protein-Bound Ligands and Their Use as Virtual Screening Filters. Journal of Chemical Information and Modeling, 45, 160-169.
- Yan, X., and Han, J. (2002). Graph-based substructure pattern mining. Proc.of the 2nd International conference on Data Mining.
- Yan, X. and Han, J., (2003). CloseGraph: Mining closed frequent graph patterns. Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining.
- Yoshida, K., and Motoda, H. (1995). CLIP: Concept learning from inference patterns. Artificial Intelligence, 75, 63-92.
- Zhang, S. et al. (2006). A Novel Automated Lazy Learning QSAR (ALL-QSAR) Approach: Method Development, Applications, and Virtual Screening of Chemical Databases Using Validated ALL-QSAR Models. Journal of Chemical Information and Modeling, 46 (5), 1984-95.
- Zhang, S., Golbraikh, A., and Tropsha, A. (2006). Development of quantitative structure-binding affinity relationship models based on novel geometrical chemical descriptors of the protein-ligand interfaces. Journal of Medicinal Chemistry, 49 (9), 2713-24.
- Zheng, W., and Tropsha, A. (2000). A Novel Variable Selection QSAR Approach Based on the k-Nearest Neighbor Principle. Journal of Chemical Information and Computer Science, 40, 185-194.