

**NOVEL ALGORITHMS AND TOOLS FOR LIGAND-BASED DRUG DESIGN**  
**A MACHINE LEARNING APPROACH FOR LIGAND PROFILING**

By

**Chao Ma**

B.S. in Science, Zhejiang University, China 2007

Submitted to the Graduate Faculty of  
School of Medicine in partial fulfillment  
of the requirements for the degree of  
PhD in Computational Biology

University of Pittsburgh

2012

UNIVERSITY OF PITTSBURGH

SCHOOL OF MEDICINE

This dissertation was presented

by

Chao Ma

It was defended on

Aug 22, 2012

and approved by

Billy W. Day, Ph.D., Professor, Department of Pharmaceutical Sciences

Ivet Bahar, Ph.D., Professor, Department of Computational & Systems Biology

Kathryn Roeder, Ph.D., Professor, Department of Statistics, Carnegie Mellon University

Panayiotis Benos, Ph.D., Associate Professor, Department of Computational & Systems Biology

Dissertation Advisor: Xiang-Qun Xie, Ph.D, Professor, Department of Pharmaceutical Sciences

# Novel Algorithms and Tools for Ligand-based Drug Design

Chao Ma

University of Pittsburgh, 2012

Computer-aided drug design (CADD) has become an indispensable component in modern drug discovery projects. The prediction of physicochemical properties and pharmacological properties of candidate compounds effectively increases the probability for drug candidates to pass latter phases of clinic trials. Ligand-based virtual screening exhibits advantages over structure-based drug design, in terms of its wide applicability and high computational efficiency. The established chemical repositories and reported bioassays form a gigantic knowledgebase to derive quantitative structure-activity relationship (QSAR) and structure-property relationship (QSPR). In addition, the rapid advance of machine learning techniques suggests new solutions for data-mining huge compound databases. In this thesis, a novel ligand classification algorithm, Ligand Classifier of Adaptively Boosting Ensemble Decision Stumps (LiCABEDS), was reported for the prediction of diverse categorical pharmacological properties. LiCABEDS was successfully applied to model 5-HT<sub>1A</sub> ligand functionality, ligand selectivity of cannabinoid receptor subtypes, and blood-brain-barrier (BBB) passage. LiCABEDS was implemented and integrated with graphical user interface, data import/export, automated model training/ prediction, and project management. Besides, a non-linear ligand classifier was proposed, using a novel Topomer kernel function in support vector machine. With the emphasis on green high-performance computing, graphics processing units are alternative platforms for computationally expensive tasks. A novel GPU algorithm was designed and implemented in order to accelerate the calculation of chemical similarities with dense-format molecular fingerprints. Finally, a compound acquisition algorithm was reported to construct structurally diverse screening library in order to enhance hit rates in high-throughput screening.

# TABLE OF CONTENTS

1	PREFACE.....	xiii
2	INTRODUCTION .....	1
2.1	OVERVIEW OF QUANTITATIVE STRUCTURE-ACTIVITY RELATIONSHIP (QSAR) 1	
2.1.1	The Basic Concept of QSAR .....	1
2.1.2	Molecular Descriptor Generation and Variable Selection .....	3
2.1.3	Mathematical Optimization.....	4
2.1.4	QSAR Validation .....	7
2.1.5	Limitation of QSAR.....	14
2.2	REVIEW OF MOLECULAR DESCRIPTORS.....	15
2.2.1	Constitutional Descriptor .....	16
2.2.2	Geometric Descriptor .....	19
2.2.3	Topological Descriptor .....	21
2.2.4	Electrostatic/Quantum-Chemical Descriptor .....	24
2.2.5	Hybridized Descriptors and Molecular Fingerprint .....	24
2.3	STATISTICAL MODELING AND MACHINE LEARNING .....	28
2.3.1	Linear Regression .....	29
2.3.2	Logistic Regression.....	30
2.3.3	Partial Least Square .....	31
2.3.4	Naive Bayes Classifier .....	33
2.3.5	Artificial Neural Network .....	35
2.3.6	Classification and Regression Tree.....	36
2.3.7	Support Vector Machine .....	38
2.3.8	Ensemble Methods.....	40
2.3.9	Miscellaneous.....	40
3	AIMS OF THE STUDY .....	42
4	LICABEDS.....	43
4.1	LICABEDS AND MODELING LIGAND FUNCTIONALITY .....	44

4.1.1	Introduction.....	44
4.1.2	Methods, Materials and Calculations.....	46
4.1.2.1	Ligand Classifier of Adaptively Boosting Ensemble Decision Stumps .....	46
4.1.2.2	Classification Tree.....	49
4.1.2.3	Naive Bayes Classifier .....	50
4.1.2.4	Dataset Preparation, Molecular Fingerprint and Computation Protocol .....	51
4.1.2.5	LiCABEDS Software Package .....	52
4.1.3	Results and Discussion.....	53
4.1.3.1	Accuracy of LiCABEDS, Tree, and Naive Bayes Classifier.....	53
4.1.3.2	Initialization Condition.....	55
4.1.3.3	Training Parameter .....	57
4.1.3.4	Reject Option.....	59
4.1.3.5	Across-target Ligand Functionality Prediction.....	60
4.1.3.6	Model Interpretation.....	61
4.1.3.7	Model Robustness .....	65
4.1.4	Conclusion .....	66
4.2	SOFTWARE MANUAL FOR LICABEDS .....	66
4.3	LICABEDS AND MODELING LIGAND SELECTIVITY .....	85
4.3.1	Introduction.....	85
4.3.2	Methods, Materials and Calculation .....	87
4.3.2.1	LiCABEDS.....	87
4.3.2.2	Support Vector Machine.....	88
4.3.2.3	Experimental Section.....	90
4.3.3	Results and Discussion.....	92
4.3.3.1	LiCABEDS and SVM in Default Settings .....	92
4.3.3.2	LiCABEDS and SVM with Cross-validation .....	95
4.3.3.3	Training Iterations of LiCABEDS.....	97
4.3.3.4	ROC Analysis of LiCABEDS models.....	102

4.3.3.5	Prediction of Selectivity of Novel Compounds.....	104
4.3.4	Conclusion .....	105
4.4	LICABEDS AND MODELING BBB PASSAGE .....	106
5	SUPPORT VECTOR MACHINE FOR LIGAND CLASSIFICATION.....	108
5.1	INTRODUCTION .....	109
5.2	METHODS, MATERIALS AND CALCULATION .....	111
5.2.1	Support Vector Machine .....	111
5.2.2	Materials and Calculations.....	113
5.3	RESULTS AND DISCUSSION .....	113
5.3.1	Molecular Properties of Agonists and Antagonists.....	114
5.3.2	Classification Performance Using Fingerprints .....	116
5.3.3	Topomer Distance Kernel .....	118
5.3.4	Model Interpretation .....	120
5.4	CONCLUSION.....	123
6	GPU-ACCELERATED COMPOUND LIBRARY COMPARISON.....	124
6.1	INTRODUCTION .....	125
6.2	METHODS AND CALCULATIONS .....	126
6.2.1	Overview of Compound Library Comparison .....	126
6.2.2	Integer Fingerprint Algorithm on GPUs .....	128
6.2.3	Sparse Vector Algorithm on GPUs.....	130
6.2.4	Find Maximum Tanimoto and Create Histogram on GPUs.....	131
6.2.5	Computation Protocols.....	132
6.3	RESULTS AND DISCUSSION.....	133
6.3.1	GPU-based Sparse Vector Algorithm Compared with Published Results.....	134
6.3.2	Integer Fingerprint Algorithm versus Sparse Vector Algorithm .....	136
6.3.3	Performance of GPU- and CPU-based Programs for Compound Library Comparison	137
6.3.4	Validation on PubChem Database .....	138
6.4	CONCLUSION.....	139
7	COMPOUND ACQUISITION ALGORITHM.....	141
7.1	INTRODUCTION .....	142

7.2	ALGORITHM DESIGN AND EXPERIMENTAL PROTOCOLS.....	143
7.2.1	BCUT Chemistry Space and Compound Acquisition Protocol .....	143
7.2.2	Distance Threshold .....	145
7.2.3	Molecular Diversity Analyses.....	145
7.3	RESULTS AND DISCUSSION .....	147
7.4	CONCLUSION.....	157
7.5	CASE STUDY .....	158
8	CONCLUSIONS AND FUTURE DIRECTIONS.....	163
9	BIBLIOGRAPHY .....	167

# LIST OF TABLES

Table 2-1: Imaginary outcome of classifier A.....	12
Table 2-2: Imaginary outcome of classifier B.....	12
Table 2-3: List of representative molecular descriptors .....	16
Table 2-4: List of representative constitutional descriptors .....	17
Table 2-5: List of constitutional descriptors and their applications .....	18
Table 2-6: List of representative geometric descriptors .....	19
Table 2-7: List of geometric descriptors and their applications .....	20
Table 2-8: List of representative topological descriptors .....	21
Table 2-9: List of topological descriptors and their applications .....	23
Table 2-10: Example applications of molecular fingerprints .....	28
Table 2-11: Machine learning algorithms in major drug discovery platforms .....	41
Table 4-1: Molecular properties of agonists and antagonists .....	51
Table 4-2: Sample mean and standard deviation of prediction accuracy .....	54
Table 4-3: Initialization condition for LiCABEDS training.....	56
Table 4-4: Across-target ligand functionality prediction .....	61
Table 4-5: List of important Molprint features regarding ligand functionality .....	63
Table 4-6: Model performance without cross-validation .....	94
Table 4-7: Model performance with cross-validation .....	97
Table 4-8: Blood-Brain-Barrier passage prediction for BBB+ and BBB- ligands.....	106
Table 5-1: Average prediction accuracy of fingerprint/kernel combinations.....	117
Table 5-2: List of structural patterns emphasized by a linear SVM classifier.....	121
Table 6-1: The number of compounds and coverage statistics for three testing compound libraries. .....	132
Table 6-2: The computation performance using integer fingerprint algorithm on machine 2.....	134
Table 6-3: The computation performance using integer fingerprint algorithm on machine 3.....	135



Table 6-4: The computation performance using sparse vector algorithm on machine 2.....	135
Table 6-5: The computation performance using sparse vector algorithm on machine 3.....	135
Table 6-6: The computation performance of Sybyl Database Comparison Program.....	137
Table 7-1: The specifications of BCUT descriptors for constructing four-dimensional chemistry space .....	147
Table 7-2: Five pairs of compounds illustrate some outliers in Figure 7-4A.....	151
Table 7-3: The average and standard deviation of Tc for different NDL and APL compound subsets, compared to the PMLSC screening collection.....	152

# LIST OF FIGURES

Figure 1-1: The diagram of drug discovery pipeline and associated cost. ....	xiv
Figure 1-2: Performance comparison of CPU and GPU .....	xvii
Figure 2-1: The illustration of knowledge-based inference.....	2
Figure 2-2 : The general steps of modeling molecular properties .....	3
Figure 2-3: Evolution of two classifiers as the amount of training data increases. ....	6
Figure 2-4: Specific model with incorrect assumption versus general purpose model .....	7
Figure 2-5: Popular cross-validation schemes.....	9
Figure 2-6: Demonstration of the goodness-of-fit.....	10
Figure 2-7: Sample ROC curve. ....	13
Figure 2-8: A “toy” mathematical model illustrating model applicability .....	15
Figure 2-9: Illustration of BCUT connectivity matrix (Tripos).....	22
Figure 2-10: Illustration of the coding scheme of Unity fingerprint by Tripos.....	26
Figure 2-11: A graphical representation of a structure pattern in Molprint 2D fingerprint.....	26
Figure 2-12: Sample sigmoid curve produced from logistic regression.....	30
Figure 2-13: A graphical representation of molecular field component in CoMFA studies. ....	32
Figure 2-14: The paradigm of artificial neural network by Sharma. ....	36
Figure 2-15: A sample classification tree for the prediction of the risk of cardiovascular disease. ....	37
Figure 2-16: SVM, maximum margin classifier, CMU 10-701 2008 spring. ....	39
Figure 4-1: Graphical illustration of LiCABEDS.....	47
Figure 4-2: The distribution of prediction accuracy from ten rounds of calculation.....	53
Figure 4-3: Prediction accuracy with different initialization conditions.....	56
Figure 4-4: The boxplot showing the effect of number of boosting rounds.....	58
Figure 4-5: “Reject option” with LiCABEDS .....	60
Figure 4-6: Four sample Molprint 2D features in graphic representation .....	63
Figure 4-7: Compounds used to exemplify four features .....	63

Figure 4-8: 3-D alignment of three agonists.....	64
Figure 4-9: 3-D alignment of four antagonists .....	65
Figure 4-10: The occurrence of the six major LiCABEDS components in ten 5-HT <sub>1A</sub> models.....	65
Figure 4-11: Graphical illustration of the constitution of a LiCABEDS classifier .....	88
Figure 4-12: Illustration of scaffold generation.....	90
Figure 4-13: The performance of selectivity prediction without cross-validation .....	93
Figure 4-14: The performance of selectivity prediction with cross-validation .....	96
Figure 4-15: Training iteration and CB2 selectivity prediction .....	98
Figure 4-16: Training iteration and CB1 selectivity prediction .....	100
Figure 4-17: Principal component analysis of fragments of CB ligands.....	100
Figure 4-18: The top five scaffolds in CB selective compounds.....	101
Figure 4-19: ROC curves of LiCABEDS models for the prediction of CB1 selectivity.....	102
Figure 4-20: ROC curves of LiCABEDS models for the prediction of CB2 selectivity.....	104
Figure 4-21: The structures of newly synthesized CB2 selective ligands.....	104
Figure 5-1: The structures of some representative 5-HT <sub>1A</sub> ligands .....	110
Figure 5-2: The distribution of molecular properties 5-HT <sub>1A</sub> agonists and antagonists.....	114
Figure 5-3: Intra-class and inter-class similarity of 5-HT <sub>1A</sub> agonists and antagonists. ....	115
Figure 5-4: Principal component analysis of five molecular descriptors. ....	115
Figure 5-5: The performance of kernel and fingerprint.....	116
Figure 5-6: The comparison of Topomer kernel and RBF kernel .....	118
Figure 5-7: Two stereoisomers that have distinct biological functionality .....	119
Figure 5-8: Histogram displaying the distribution of elements in vector w.....	120
Figure 5-9: The distribution of support vector coefficient .....	122
Figure 6-1: Flowchart of similarity calculation on GPU.....	127
Figure 6-2: Similarity calculation based on dense-format fingerprint.....	129
Figure 6-3: Illustration of data structure of sparse vectors.....	130
Figure 6-4: Tanimoto coefficient and sparse vector fingerprint.....	131
Figure 6-5: The plot of elapsed time versus processed PubChem compounds. ....	139
Figure 7-1: Motivation of compound acquisition protocol .....	144
Figure 7-2: Two-dimensional chemistry space and filled/void cells.....	146
Figure 7-3: The distribution of four chemistry-space descriptors for the PMLSC screening set.....	148
Figure 7-4: Correlation between Tanimoto Coefficient and Euclidean distance in BCUT chemistry space.....	149
Figure 7-5: Histogram and probability density.....	150

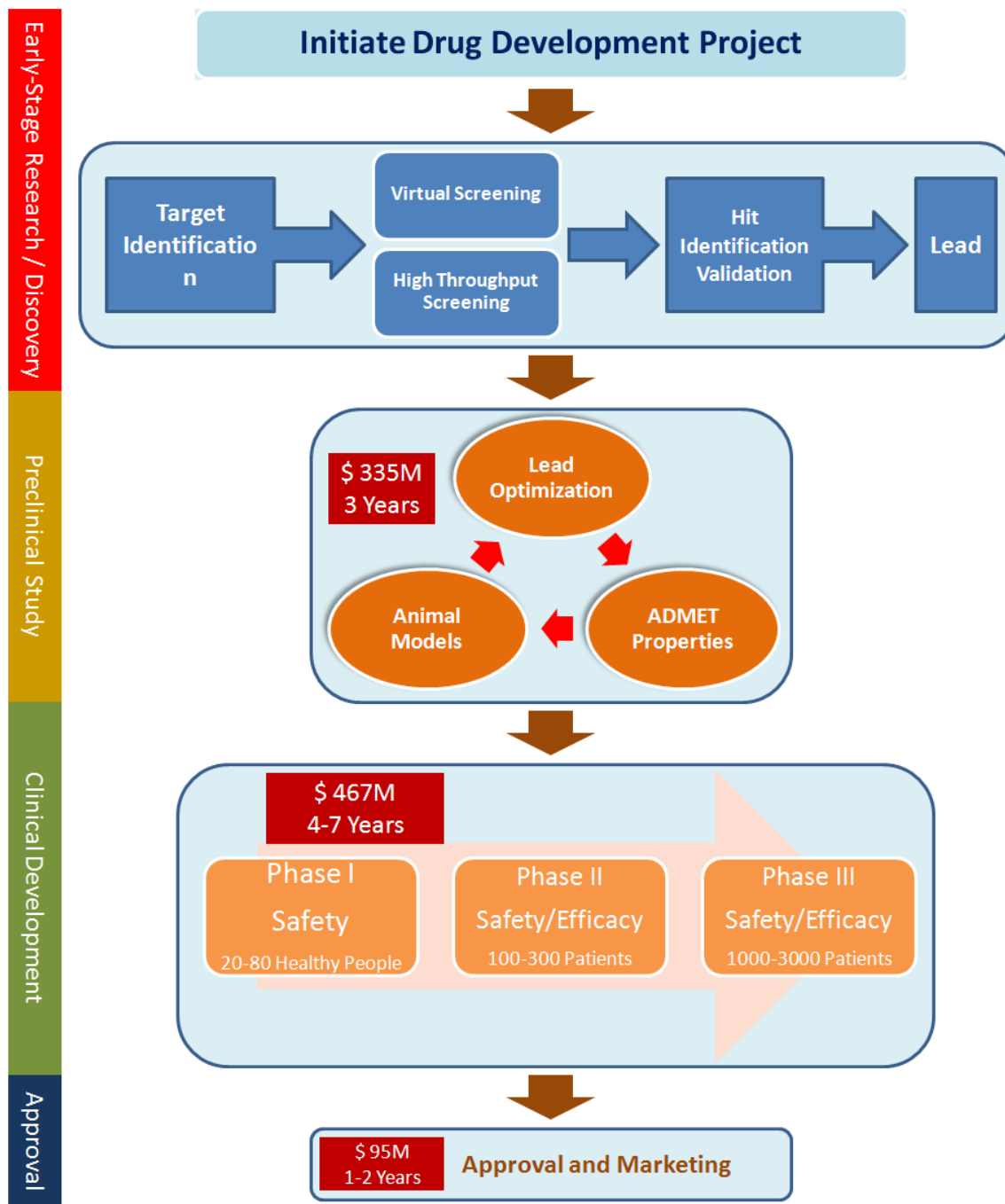
Figure 7-6: Distribution of Tanimoto Coefficients from Database Comparison.....	153
Figure 7-7: The number of filled void cell versus the number of acquired compounds.....	156
Figure 7-8: 3D chemistry-space plots.....	160
Figure 7-9: A database similarity comparison of 43 compounds with SMR library.....	161

# 1 PREFACE

Modern drug discovery and development involve a set of organized processes, including drug target identification and validation, lead compound discovery, early ADMET-oriented optimization, clinic phases, etc. Studies have shown that the cost of the whole drug discovery and development process varies from 500 million to 2 billion dollars<sup>1-3</sup>. The development of a new drug typically takes 10 to 15 years. Among all, clinic testing is the most extensive and expensive phase. The recent trend shows that the failure rate of clinic Phase II and Phase III has been rising<sup>4-5</sup>. The failure of passing clinic experiments denies drug candidate compounds and turns the significant amount of investment fruitless.

The investment on drug discovery is highly risky but highly rewarded. Hence, substantial efforts have been devoted to the investigation of novel and economical approaches with the intention to boost the productivity of pharmaceutical industry and deliver more products to the market. State-of-the-art high-throughput screening (HTS) technology points out a potential solution for lead compound identification. Modern HTS technology is capable of screening 100,000 compounds per day<sup>6</sup>. Furthermore, HTS facilities that were exclusively owned by drug discovery enterprise are becoming readily available to research institutions. Nevertheless, screening the vast number of compounds is still challenging in absence of sufficient funds.

Nowadays, virtual screening has been integrated into the pipeline of drug discovery process (Figure 1-1). The whole discovery process approximately costs 800 million dollars and 7 to 12 years, depending on the types of disease<sup>7</sup>. Virtual screening is a generic computational technique that explores large virtual chemical libraries in order to assess the interaction between ligands and a hypothetical protein receptor or enzyme. Broadly speaking, computer-assisted ADMET property prediction and lead optimization are also part of virtual screening. Virtual screening exhibits two obvious advantages over traditional high-throughput screening<sup>8</sup>. First, virtual screening software is able to evaluate compound collections that do not physically exist. Second, the predictions from virtual screening programs refine the scope of to-be-explored chemistry space and drastically reduce the cost of bioassays.



**Figure 1-1: The diagram of drug discovery pipeline and associated cost.** The whole drug discovery project typically includes early-stage research phase, preclinical study phase and clinical phase. It takes approximately 10 ~ 15 years and 500 million ~ 2 billion dollar to develop a marketable drug.

Currently, there are two categories of approaches in virtual screening: ligand-based virtual screening and structure-based virtual screening. Ligand-based approaches rely on the assumption of “similarity principal”, which states that structurally similar compounds usually exhibit similar biological activity. Nevertheless, the definition of molecular similarity is still ambiguous up to now. A specific pair of compounds may have different similarity score according to various criteria, for example, 3D similarity, maximum common structure, fingerprint-based Tanimoto coefficient, etc. On the other hand, structure-based approaches imitate ligand-receptor interaction, in which ligands are “docked” into a hypothetical receptor, and scores are calculated to estimate binding affinity. Thus, the scoring function or ligand ranking mechanism is an important component of this methodology. Unfortunately, the performance of scoring functions in most “docking” software is far from perfect. Furthermore, the availability of high-quality protein structures also restricts the application scope of structure-based virtual screening. Despite certain limitations, both categories of approaches have been successfully applied to solve practical problems in computer-aided drug design (CADD). In this thesis, the reported algorithms and software are ligand-based.

Regardless of ligand-based or structure-based drug design, virtual screening emphasizes on the discovery of lead compounds that can bind to a protein receptor or enzyme. Besides bioactivity, lack of desired pharmacological and physicochemical properties is another important factor that rejects drug candidates from further development. As a complement to modern high-throughput screening, one of the primary goals of computer-aided drug design is to explore the enormous chemical and biological properties in a time-efficient manner as well as to reduce the cost of experimental screening<sup>9-11</sup>. Particularly, great emphasis is placed on the “drugability” or “drug-likeness” of compounds using cheminformatics tools in the early stages of drug development, with the hope of increasing the probability of “lead” compounds, or their derivatives, to pass through the later phases of drug clinical trials<sup>12</sup>. Thus, generic and robust tools are demanded for the prediction of various ligand properties.

Modern information technology and software engineering methodology produce many high-performance programs for cheminformatics and computer-aided drug design. Due to growth of computer hardware industry, it is widely assumed that *in silico* virtual screening and CADD programs are fast and high-throughput. On the contrary, the speed of these programs is sometimes achieved at the sacrifice of model quality and prediction accuracy. Molecular docking programs may take several seconds or tens of minutes to fit a ligand into a binding pocket, depending on searching algorithms and searching parameters. Thorough examination of numerous possible conformations of both ligand and binding pocket tends to yield convincing binding hypotheses, which has exponential time complexity in the degree of freedom. Casual docking strategies, such as rigid-body docking, hardly reveal reliable ligand-receptor binding

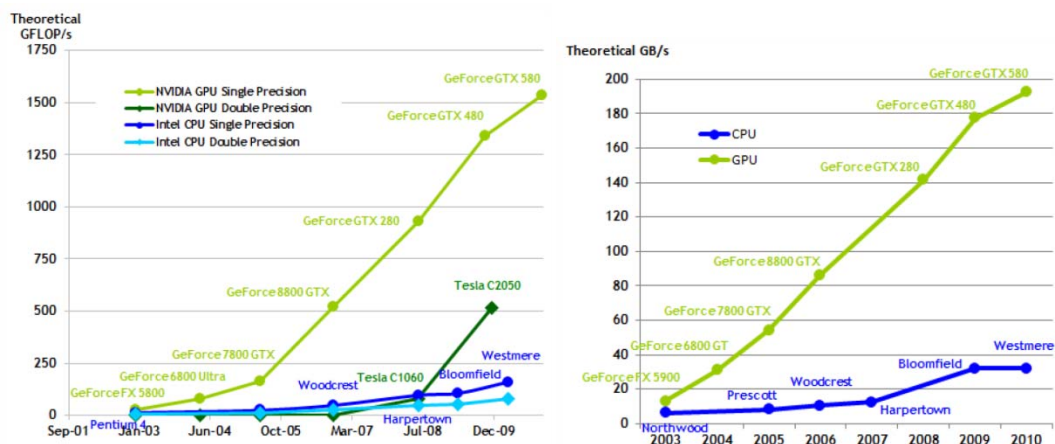
mode. Moreover, some algorithms require large amount of subjective intervention, resulting in relatively low level of automation and unsatisfactory efficiency. The famous field-based pharmacophore algorithm, Comparative Molecular Field Analysis (CoMFA)<sup>13</sup>, is a typical example. CoMFA predicts ligand binding affinities or other molecular properties according to the goodness-of-fit between the ligands and predefined CoMFA model, which specifies favored and disfavored pharmacophore groups in a three-dimensional grid. One of the prerequisites of CoMFA is manually searching for the bioactive conformations that can be fitted into a developed CoMFA model. Given the same ligand, different CoMFA predictions are assigned to different conformations, and the conformation search is usually subjective. Therefore, 3D bioactive conformation search still remains as one of its limitations<sup>14</sup> and as a barrier for high-throughput virtual screening.

Despite the development of cheminformatics methodology, it is a challenge to develop reliable, interpretable and high-throughput computational algorithms for the automation of modeling assorted molecular properties. The success of developing high-throughput algorithms will effectively accelerate the process of computer-aided drug design, and produce consistent and objective predictions. Most of the developed algorithms in CADD are based on physical models and empirical rules. The solutions are generally found by exhaustive search, approximation, or simulation. Physical models are straightforward and effective to some extent. Meanwhile, machine learning methodology is gaining popularity in the community of computer-aided drug design. In the current decade, machine learning has been successfully applied for image analysis, natural language processing, speech and handwriting recognition, etc. Informally, machine learning is a branch of artificial intelligence, aiming at identifying complex patterns and making intelligent decisions according to presented observations<sup>15</sup>. The advantage of machine learning approaches is the capability to discover certain correlations, the rule of which remains unknown. The methods and algorithms reported in this thesis are the instances of machine learning algorithms in cheminformatics and computer-aided drug design.

Recently, the computation power of single-core CPUs is reaching a bottleneck. Going parallel is the theme nowadays. Supercomputers and clusters, formed by a set of integrated processing units, were not strangers to scientists even 20 years ago. To speed up calculation, scientists usually distribute computationally intensive tasks to many “computer nodes”. It is known that the computation time does not reduce linearly as the number of computer nodes increases. Molecular dynamics (MD) is a common computational technique in structural biology. Performance gain of MD simulation is not obvious by adding more computer nodes, when hundreds of nodes have already been utilized. Meaningful MD of DNA and proteins simulates a process spanning nanoseconds to microseconds. Unfortunately, CPU-days to CPU-years are required to complete the simulation<sup>16</sup>.



With the emphasis on “green high performance computing”, the development of modern graphics processing units (GPU) suggests a substitute strategy for scientific computing. GPUs are specialized microprocessors for graphics rendering. Modern GPUs feature higher memory bandwidth and more floating point operations per second (FLOPS) than CPUs (see Figure 1-2). These days, general-purpose computing on graphics processing units (GPGPU) is an active research field. The computation power of GPUs resides in their highly parallel architecture. As GPUs are mainly built for graphics rendering, developing GPU programs is not as flexible as traditional CPU programs. For example, high memory bandwidth is only achieved when global memory (video memory) access is coalesced. In addition, execution divergence reduces the degree of parallelism and device utilization. Therefore, extra attention should be paid to the design and implementation of GPU algorithms. In this thesis, a novel GPU algorithm and its implementation are introduced for fingerprint-based chemical similarity calculation and compound database comparison.



**Figure 1-2: Performance comparison of CPU and GPU**

This figure compares theoretical computation power and memory bandwidth of CPU and GPU models from 2001 to 2010. Computation power (left figure) is measured in Giga Floating-point Operations per Second (GFLOP/s), and memory bandwidth (right figure) is given in Giga-Byte per Second (GB/s)<sup>17</sup>

## ACKNOWLEDGEMENT

The research work reported in this thesis was carried out in University of Pittsburgh 2007-2012. I wish to thank my supervisor, Prof. Xiang-Qun Xie, for solid funding support and constructive advice on my dissertation.

I wish to thank my thesis committee members, Dr. Ivet Bahar, Dr. Billy Day, Dr. Takis Benos and Dr. Kathryn Roeder, for their helpful comments on my research projects. I also thank Prof. Kathryn Roeder for her excellent lectures, which guided me into the world of applied statistics. In addition, I would like to thank Dr. Lirong Wang for technical support, Dr. Peng Yang for compound synthesis, and Qin Tong for running bioassays.

## 2 INTRODUCTION

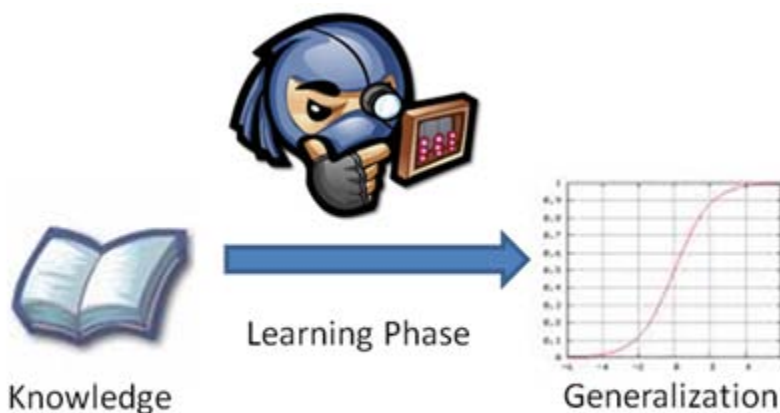
### 2.1 OVERVIEW OF QUANTITATIVE STRUCTURE-ACTIVITY RELATIONSHIP (QSAR)

About 20 years ago, comparative molecular field analysis (CoMFA) was almost equivalent to Quantitative Structure-Activity Relationship (QSAR). Modern machine learning algorithms and cheminformatics techniques brought up various methods to model not only biological affinity but also pharmacological properties. Significant efforts have been devoted to developing reliable, adaptive and robust QSAR algorithms. Although successful QSAR studies have been reported in many fields of computer-aided drug design, their true predictability is still questioned by many practitioners<sup>18</sup>. The development of robust quantitative structure-activity relationship is far beyond automated model training from some commercial software. A robust model requires extensive investigation of molecular descriptor selection, mathematical model training and, most importantly, model validation. Furthermore, scientists should understand the prediction risk when structurally diverse compound collection is involved. In this section, the methodologies involved in QSAR, including descriptor generation, statistical modeling, model validation and the limitation of QSAR, are briefly discussed.

#### 2.1.1 *The Basic Concept of QSAR*

The fundamental goal of computer-aided drug design (CADD) is to predict biological activity and pharmacological properties of drug candidates in order to enhance the productivity and reduce the cost of drug development project. To certain extent, any CADD project can be reduced to a decision-making

process, for example, the prediction of binding affinity, LogP value, toxicity, etc. In general, such decision-making can be carried out through two classes of methods: rule-based method and knowledge-based method. In rule-based method, prediction is made according to widely validated and accepted theories. One example is the estimation of ligand-receptor binding energy using force field. In the current stage, no complete set of theories are established to handle diverse challenges in rational drug design. Then, knowledge-based method becomes a practical approach in these unexplored fields. In one word, knowledge-based method attempts to derive an empirical correlation between the observations and molecular properties of interest, without understanding their underlying mechanisms. Illustration is given in Figure 2-1. In CADD, knowledge is the compound libraries annotated with molecular properties of interest, for example, the results from HTS experiments. The knowledge is then presented to a “learner”. The learner’s job is to detect patterns that are statistically correlated with properties of interest, and solve parameters in mathematical models. Finally, the learner outputs a prediction scheme for future testing samples.

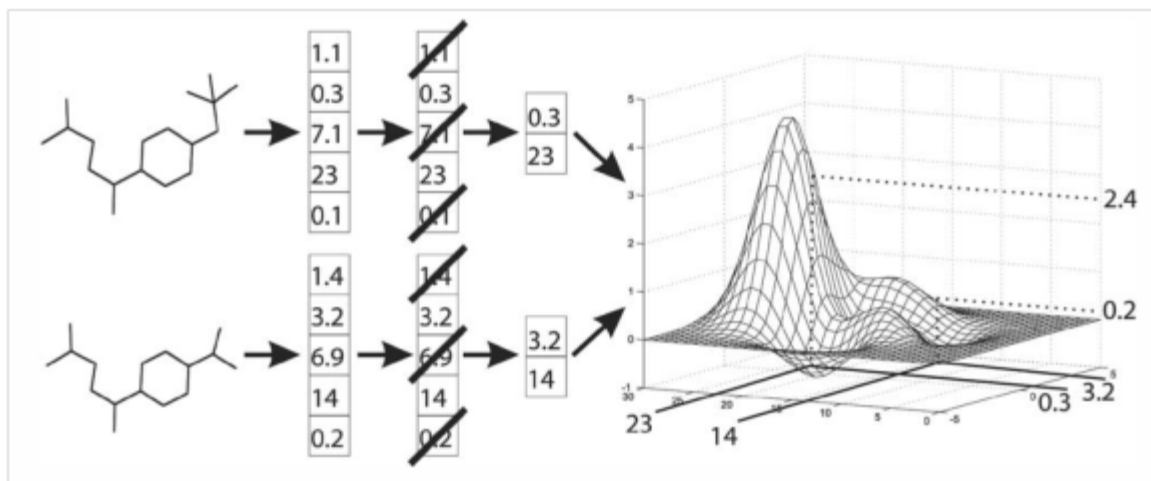


**Figure 2-1: The illustration of knowledge-based inference**

In this diagram, a learner aims at solving a mathematical model (generalization) from training data (knowledge) through numerical optimization (learning phase)

Advanced high-throughput screening (HTS) technologies generate great amounts of bioactivity data, and this data needs to be analyzed and interpreted with intension to understand how these small molecules affect biological systems. As such, there is an increasing demand to develop and adapt cheminformatics algorithms and tools in order to predict molecular and pharmacological properties based on these large datasets. These techniques belong to an active research area, called quantitative structure-activity relationship (QSAR). Figure 2-2 summarizes the major steps of a QSAR study: molecular descriptor

generation, variable selection and numerical optimization. Finally, the QSAR model is validated retrospectively or prospectively before integrating the model into drug development pipeline.



**Figure 2-2 : The general steps of modeling molecular properties**

In ligand-based drug design, compound structures are represented in high-dimensional molecular descriptors. This is followed by a step called variable selection to choose predictive features. The predictive features are used to statistically correlate with molecular properties of interest.<sup>19</sup>

### **2.1.2 Molecular Descriptor Generation and Variable Selection**

In QSAR, compound structures are usually plugged into mathematical models implicitly. Structures can be treated as weighted graphs. Unfortunately, most successful statistical and machine learning algorithms adopt numerical vectors as input. In addition, manipulating numerical vectors and matrices is more efficient compared to graph operations. Thus, mapping chemical structures to high-dimensional descriptor vectors is the recommended strategy for high-throughput virtual screening programs. Another advantage of performing pattern recognition on molecular descriptors instead of chemical structures is the incorporation of hypotheses and assumptions into the mathematical model. In this case, data mining can be restricted in relevant hypothesis space, instead of exploring combinations of noisy structural information. Descriptor generation is the foundation for any QSAR studies. A comprehensive review on prevailing molecular descriptors will be given later.

Variable selection targets at removing redundant and irrelevant molecular features, but preserving predictive and informative variables. A large number of descriptors may increase model complexity, but developed models may be vulnerable to over-fitting. In statistics, parameterization may become intractable as the dimensionality of feature space increases, because enormous data are required to depict the probability density of combinations of input variables. This phenomenon is called “curse-of-

dimension”. Modern machine learning algorithms are not as sensitive as traditional statistical models, regarding high-dimensional feature space. Some robust pattern recognition methods possess built-in mechanism to attenuate the negative effect of irrelevant features. Margin maximization and parameter regularization are two common approaches. Therefore, variable selection is not strictly demanded in QSAR anymore, but it is still recommended in many cases.

### 2.1.3 *Mathematical Optimization*

Solving parameters and variables in a quantitative model is the follow-up step of a QSAR study. Regardless of mathematical models, the parameterization procedure is equivalent to the optimization of an object function. The object function penalizes prediction errors and rewards predictions that are consistent with observations. The parameters that optimize the object function become the solution. This procedure is also referred as “model training”, and the collection of annotated compounds that the solution is derived from is usually called “training set”. In other words, model training is conducted to search for an optimal mapping function,  $f, f \in \mathcal{F}: \mathbf{x} \rightarrow y$ , where  $\mathbf{x}$  is the descriptor vector and  $y$  is a response value. If  $y$  is a discrete scalar, representing compound labels, the inference belongs to classification, for example, predicting whether a compound is active or not. If  $y$  is a continuous scalar, the inference belongs to regression, for example, predicting the binding affinity of a compound. Broadly speaking, there exist three schemes for modeling the correlation between  $\mathbf{x} \rightarrow y$  <sup>20</sup>:

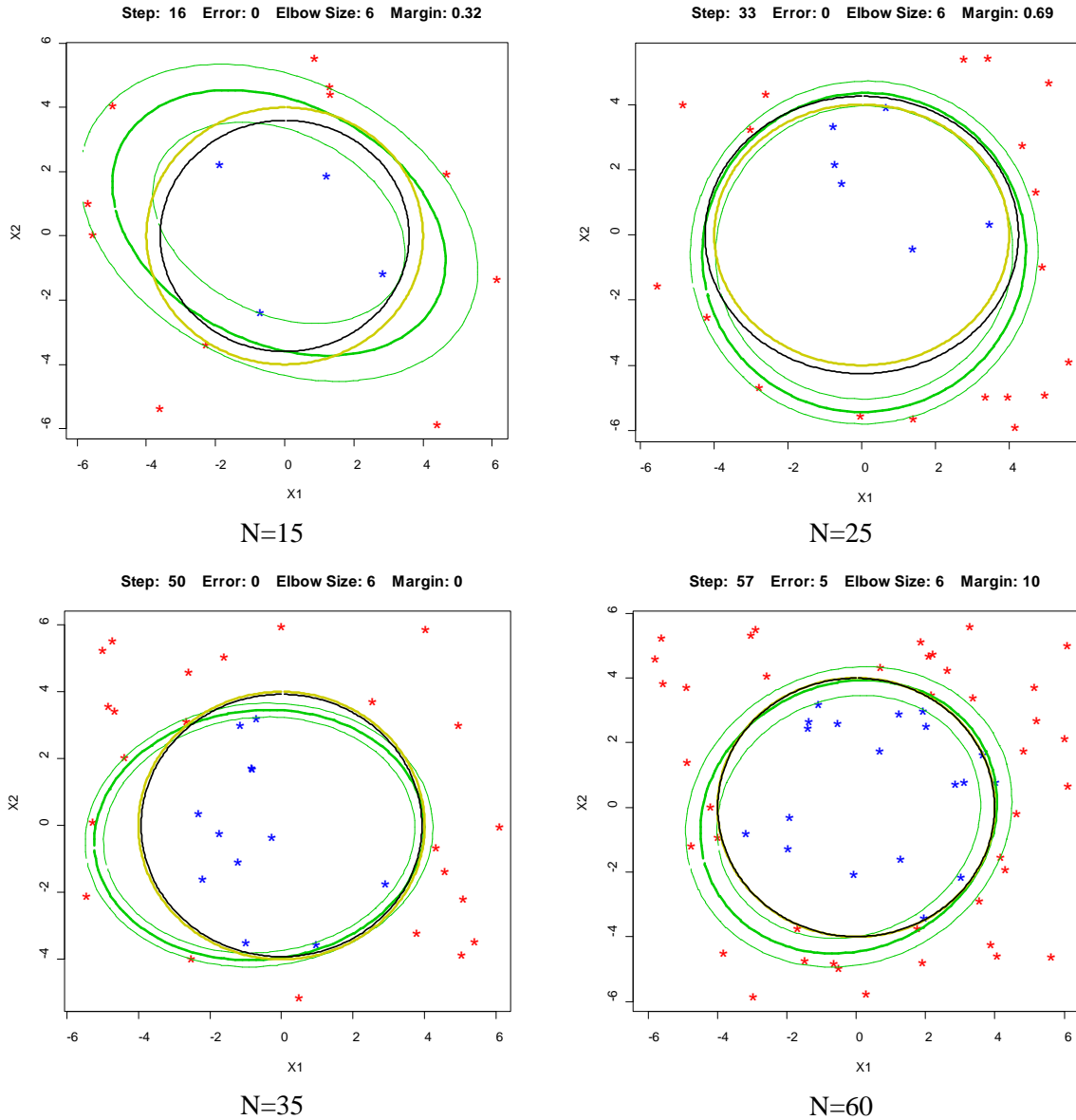
1. Modeling the joint probability density  $p(\mathbf{x}, y)$ . This is the most comprehensive solution of estimating the generalization error and confidence interval, as the function  $p(\mathbf{x}, y)$  reveals the distribution of descriptor vector  $\mathbf{x}$ . Furthermore, the conditional probability,  $p(y|\mathbf{x})$ , can be easily derived to give prediction  $y$  from input  $\mathbf{x}$ . Modeling  $p(\mathbf{x}, y)$  requires vast observations when vector  $\mathbf{x}$  is high dimensional, because of “curse-of-dimension” described above. On the other hand, this scheme is still feasible under certain circumstance. In Naive Bayes classifier, the independence assumption simplifies the estimation of  $p(\mathbf{x}, y)$  by formulating it as a product,  $p(\mathbf{x}, y) = \prod_i p(x_i, y)$ .
2. Modeling the conditional probability  $p(y|\mathbf{x})$ . Knowing the distribution of  $\mathbf{x}$  is sometimes impractical, but the conditional probability  $p(y|\mathbf{x})$  is sufficient to make inference on  $y$  according to descriptor vector  $\mathbf{x}$ . Given any predefined cost function,  $g(\hat{y}, y)$ , the object function is  $E_{y|\mathbf{x}}(g(\hat{y}, y))$  where  $\hat{y} = f(\mathbf{x})$ . This approach neglects the distribution of  $\mathbf{x}$  and only focuses on the distribution of  $y|\mathbf{x}$ , so it is more robust than scheme 1. Yet, the generalization error, i.e.  $E_{\mathbf{x}}(E_{y|\mathbf{x}}(g(\hat{y}, y)))$ , is unsolvable because  $p(\mathbf{x})$  remains unknown. Logistic regression is one example of this scheme.

3. Find function  $\hat{y} = f(\mathbf{x})$  without modeling probability density function. This scheme is adopted by most machine learning algorithms. Since no assumptions are made on the distribution of  $\mathbf{x}$  and  $y$ , this method is the most robust of all. Without probability and distribution, the generalization error is empirically estimated by  $\frac{1}{N} \sum_i g(\hat{y}_i, y_i)$ . Furthermore, no confidence interval can be guaranteed for any prediction,  $\hat{y}_i$ . Famous Support Vector Machine and Artificial Neural Network belong to this scheme.

Correct model assumptions drive parameters to converge faster compared to the case in which little information is known. Figure 2-3 demonstrates this principal through a toy model.  $N$  training observations  $\mathbf{t}_i \in \{(x_1, x_2): x_1, x_2 \in [-6, 6]\}$ . The corresponding class or label,  $y_i$ , for observation,  $i$ , is generated according to the following rule:  $y_i = 1$ , if  $\sqrt{x_1^2 + x_2^2} \geq 4$ ; or  $y_i = -1$  otherwise. To make it more simple, the data points outside the yellow circle in Figure 2-3 belong to one class, while the data points inside the circle belong to another class. In reality, the mechanism of data generation is probably unknown. The goal is to find a mapping function  $y = f(\mathbf{t})$  that emulates the true mechanism according to the observations  $y$  and  $\mathbf{t}$ . Two types of classifiers are trained. One is famous support vector machine (SVM) with polynomial kernel of second degree. Theoretically, the SVM classifier is capable of fitting any quadratic functions, which definitely covers the true “circle” classifier. The other classifier is  $y = 2I(\sqrt{x_1^2 + x_2^2} \geq r) - 1$ .  $I(S)$  is an indicator function, which will be used frequently later.  $I(S) = 1$  if the statement  $S$  is true, or  $I(S) = 0$  otherwise. For the second classifier, it is clear that the label is assigned according to the distance to the origin, but the threshold value,  $r$ , remains unknown. The training procedure aims at making inference on  $r$  according to the training data.

The SVM classifier is solved using SVMPATH package<sup>21</sup>. More details regarding SVM will be given in the following chapters. The radius  $r$  of the “circle” classifier is solved using the equation  $r = \frac{1}{2}(\min_{i, y_i=1} \|\mathbf{t}_i\|_2 + \max_{i, y_i=-1} \|\mathbf{t}_i\|_2)$ .  $\|\mathbf{t}_i\|_2$  is the L2 norm of  $\mathbf{t}_i$ . Its value is equal to  $\sqrt{x_1^2 + x_2^2}$  of the  $i^{\text{th}}$  data point. As shown in Figure 2-3, the “circle” classifier colored in black converges fast to the yellow circle, according to which the labels are assigned. When  $N = 35$ , the black circle starts to overlap with the yellow one. The SVM decision boundary is represented by the thick green curve. The thin green curve represents the soft margin in SVM. As  $N$  increases, the SVM decision boundary gets closer and closer to the yellow circle, but noticeable discrepancy between the two exists. Even if SVM is a robust and flexible classifier, its convergence speed is much slower. In practice, the correct assumption helps us to develop a high-quality QSAR model with only limited training data, however most topics in CADD are not well explored, and the assumptions might be wrong. Figure 2-4 gives one example resulted from wrong model assumption. The mechanism of data generation is the same except that the yellow circle is centered at

point (1, 1). Both classifiers are trained using the same algorithm. The “circle” classifier in this case is inaccurate as it is always centered at the origin. In Figure 2-4, SVM classifier is undoubtedly superior to the “circle” classifier. Thus, a general-purpose classifier is a good choice if the assumptions are uncertain or little is known about the solution.

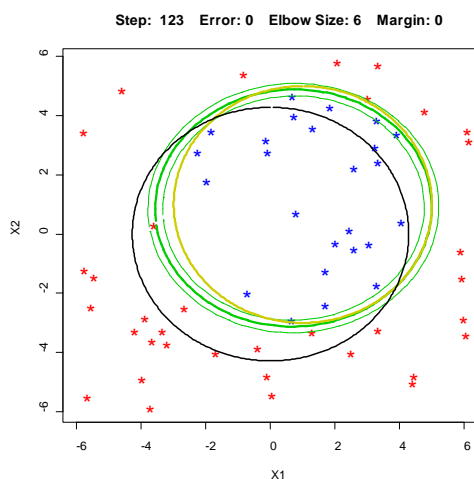


**Figure 2-3: Evolution of two classifiers as the amount of training data increases.**

Each figure shows the developed classification boundaries with different training data sizes (N). Yellow circle is the true boundary that is used to generate training data. Green lines represent the decision boundary and soft margin of support vector machine. And the black line represents trivial “circle classifier”



In CADD, the amount of training data is usually limited, and labeling novel compounds is remarkably expensive and time-consuming, but an accurate prediction model is still desired in practice. Prior assumptions restrict the search space of training algorithms, so limited training data may be still sufficient for statistical inference. As illustrated in the toy model, the “circle” classifier only represents a tiny fraction of the complete quadratic SVM solutions. Therefore, the parameter in the “circle” classifier converges faster than SVM because the training algorithm does not necessarily explore all the quadratic solutions. Conversely with incorrect assumptions (Figure 2-4), the true solution actually lies outside of the search space, and finally parameters converge in a wrong direction. To conclude, the balanced tradeoff between a general-purpose model and a problem-specific model should be well maintained for a QSAR study.



**Figure 2-4: Specific model with incorrect assumption versus general purpose model**

The yellow circle, centered at (1, 1), represents the true decision boundary. Support vector machine converges reasonably well to the yellow circle, and outperforms the trivial “circle classifier” that is centered at origin.

#### 2.1.4 QSAR Validation

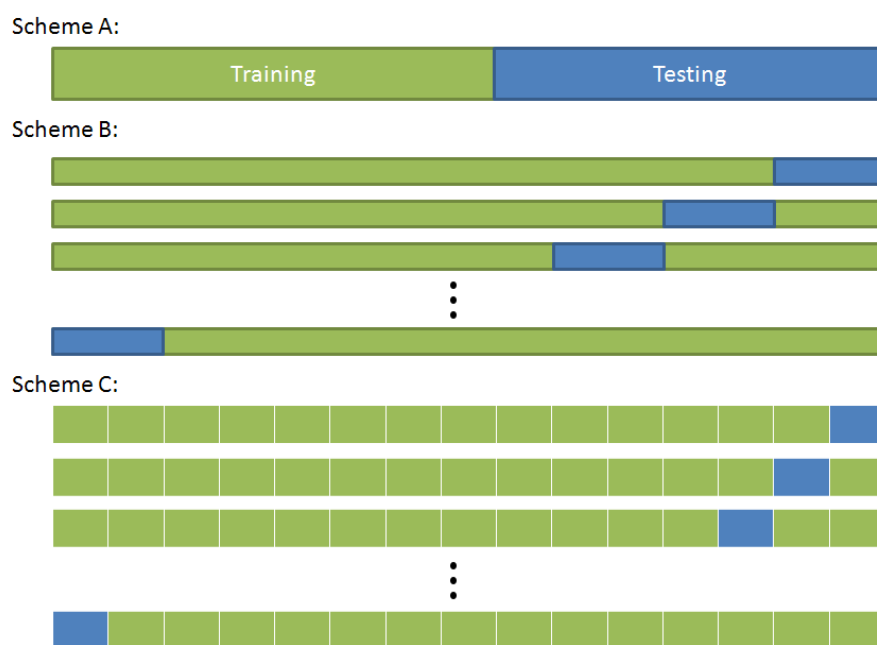
The validation of a quantitative structure-activity relationship is probably the most important step of all. The validation estimates the reliability and accuracy of predictions before the model is put into practice. Poor predictions misguide the direction of drug development and turn downstream efforts meaningless. Validation can be performed either retrospectively or prospectively<sup>22</sup>. In retrospective validation, a

QSAR model is developed without being exposed to some annotated compounds (testing set). To verify model quality, predictions are made on the testing set in order to check the agreement between the theoretical values and experimental values. In prospective validation, the model is applied on an external unlabeled compound set. Then, physical experiments are carried out to validate the predictions. Prospective validation imitates the scenario of actual drug development, which is more trust-worthy. Nowadays, prospective validation is required in major cheminformatics journals for QSAR publications reporting existing methods. Nevertheless, it is necessary to assay a large number of testing compounds in order to draw statistically convincing conclusion, but prospective validation, as a costly and time-consuming approach, is impractical in many cases.

Cross-Validation is probably the most popular technique for estimating generalization error of QSAR models. Generalization error of QSAR models measures their average performance when the models are generalized to the entire possible chemistry space. Thus, generalization error is an important indicator showing the risk of utilizing the models in drug development process. Define an error function  $f_M, f_M \in \mathcal{F}: \mathbf{x} \rightarrow \mathbb{R}$ , where  $\mathbf{x}$  represents a compound.  $f_M(\mathbf{x})$  returns the prediction error for compound  $\mathbf{x}$  with model  $M$ . Precisely, the generalization error of a specific model  $M$  is equal to  $E_{\mathbf{x}}(f_M(\mathbf{x}))$ . In almost all the cases, the analytical solution to  $E_{\mathbf{x}}(f_M(\mathbf{x}))$  does not exist due to the astronomical number of possible compound structures for  $\mathbf{x}$ . In statistics, the law of large numbers (LLN) is a theorem, stating that  $\bar{X}_n \rightarrow \mu$  for  $n \rightarrow \infty$  where  $\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i$  and  $\mu = E(X_i)$ ,  $i = 1, 2, \dots, n$ .  $X_1, X_2, X_3, \dots$  are i.i.d. random variables. By applying LLN, generalization error  $E_{\mathbf{x}}(f_M(\mathbf{x}))$  can be approximated by  $E_{\mathbf{x}}(f_M(\mathbf{x})) \approx \frac{1}{n} \sum_{i=1}^n f_M(\mathbf{x}_i)$ .  $\mathbf{x}_i$  are independent testing samples. There are four basic requirements for  $\mathbf{x}_i$ : (1) the experimental property value of  $\mathbf{x}_i$  is attainable in order to calculate the cost function  $f$ ; (2)  $\mathbf{x}_i$  are independent training samples, upon which QSAR models are developed; (3)  $\mathbf{x}_i$  are true random samples without oversampling a specific region of chemistry space. (4)  $n$  is large and there exist ample testing samples.

In reality, these requirements are hardly achieved due to the high expense in acquiring experimental values of many testing compounds. Thus, a fraction of annotated compounds is “left-out” from training data set for estimating the generalization error, which is called cross-validation. Figure 2-5 illustrates some popular cross-validation techniques. In Scheme A, certain percentage of annotated compounds (training set) is used to develop QSAR models. Then, the models are evaluated on the remaining compounds (testing set). Usually, a set of model candidates are evaluated on the testing set and the one with lowest prediction error is selected as the optimal model. This scheme is straightforward but not flawless, because the testing set might be overfit. A model overfits a data set when it captures random noise instead of the underlying relationship. An overfit model usually shows poor prediction performance. In Scheme B, the annotated compound data set is split into  $k$  subsample sets and the testing calculation is

repeated for  $k$  times. Each time, one of the  $k$  subsample sets is left out as testing set and the remaining  $k-1$  subsample sets are used as the training set. The generalization error is simply approximated by the average prediction error on the  $k$  testing sets. Although the possibility of overfitting still exists, the chance is lower compared to Scheme A, because more testing samples are involved in the approximation of generalization error. This technique is called  $k$ -fold cross-validation. Scheme C, which is also known as leave-one-out cross-validation (LOOCV), is a special case of  $k$ -fold cross-validation. As suggested by the name, each time one annotated compound is kept from the remaining training data as testing sample. The procedure is repeated until every compound is used once as testing data. For some statistical models, such as linear regression, analytical solution exists for estimating LOOCV error. Generally, LOOCV error is calculated by repeating model training and testing for  $N$  times ( $N$  is the total number of annotated compounds), which is computationally expensive.



**Figure 2-5: Popular cross-validation schemes**

In scheme A, certain proportion of annotated data (training set) is used to develop a predictive model, and the model is validated on the remaining data (testing set). In scheme B ( $k$ -fold cross-validation),  $1/k$  of annotated data is reserved as testing set, and the remaining is left for model training. This procedure is repeated  $k$  times. In scheme C (leave-one-out cross-validation), every data entry has chance to serve as testing data, leaving the remaining data as training set.

Besides cross-validation techniques, error function should be rationally designed for risk estimation and model selection, depending on different CADD applications. Regression and classification are two major methodologies involved in QSAR. Regression analysis aims at predicting real values, such as dissociation constant, boiling point, LogP, etc. The goal of classification is to predicting the categories of compounds, for example, active or inactive, selective or unselective, toxic or nontoxic, etc.

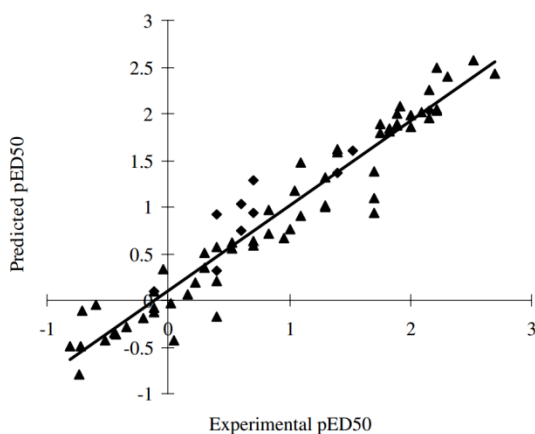
In regression analysis,  $R^2$  or correlation coefficient ( $r$ ) is used to measure the goodness of fit. Let  $y_i$  denote the experimental value for compound  $i$  and  $\hat{y}_i$  denote the corresponding predicted value from regression model. The total variance can be decomposed as <sup>23</sup>:

$$\sum_{i=1}^n (y_i - \bar{y})^2 = \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \sum_{i=1}^n (\hat{y}_i - \bar{y})^2.$$

$$SS_{\text{total}} = \text{RSS} + SS_{\text{reg}}$$

$SS_{\text{total}}$ : total sum of squares;  $\text{RSS}$ : residual sum of squares;  $SS_{\text{reg}}$ : sum of squares for regression.  $y_i - \hat{y}_i$  is the difference between observed value and predicted value, which is also called residue.

According to this decomposition,  $R^2 = 1 - \frac{\text{RSS}}{SS_{\text{total}}} = \frac{SS_{\text{reg}}}{SS_{\text{total}}}$ . Thus,  $R^2$  can be understood as the ratio of variance explained by the regression model to the total variance. In other words,  $R^2$  indicates the proportion of total variance, traced by the regression model. In an ideal case, a regression model can predict 100% of uncertainty and variance of  $y$ , then  $R^2 = 1$ . In the worst case, a regression model is uncorrelated with  $y$ , then  $R^2 = 0$ . Sometimes  $R^2$  calculated on the cross-validation set can be negative when  $\text{RSS}$  is greater than  $SS_{\text{total}}$ . Criteria for judging whether an  $R^2$  is satisfactory or not remains flexible. Normally for QSAR studies, an  $R^2$  value above 0.5 indicates good predictability <sup>24</sup>. Another way to illustrate the goodness of fit is by plotting experimental value versus predicted value, as shown in Figure 2-6.



**Figure 2-6: Demonstration of the goodness-of-fit**  
Figure adopted from reference <sup>25</sup>

In compound classification, compound labels or categories, such as “active” or “inactive”, are represented by some integers. One effective measure of prediction quality is the ratio of the number of correct predictions to the total number of testing compounds. This measurement was applied in modeling drug-likeness with binary classifiers<sup>26</sup>. Nevertheless, the percentage of correct prediction is meaningless when the testing set is overwhelmed by one category of observations or the category of interest is minority, for example virtual screening. In virtual screening, a few active compounds are immersed in the sea of inactive ones. A classifier that blindly assigns “inactive” label to any testing compound probably achieves 99% prediction accuracy. The classifiers in sole pursuit of high prediction accuracy probably will never recover any active compounds. Therefore, a few other indices have been proposed for unbalanced data sets. Take virtual screening as an example, in which a classifier attempts to distinguish active compounds from inactive ones. Define the following quantities:

True Positives (TP): the number of active compounds that are also predicted as active

True Negatives (TN): the number of inactive compounds that are also predicted as inactive

False Positives (FP): the number of inactive compounds that are incorrectly predicted as active

False Negatives (FN): the number of active compounds that are incorrectly predicted as inactive

Matthews correlation coefficient (MCC) is frequently adapted as a measure of the quality of binary classifiers for unbalanced data sets. MCC is calculated according to the following equation:

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}$$

The example below demonstrates the difference between MCC and prediction accuracy. Suppose that a collection of 1000 compounds has 10 active compounds and 990 inactive compounds. The outcome of classifier A and classifier B is listed in Table 2-1 and Table 2-2 respectively.

**Table 2-1: Imaginary outcome of classifier A**

Predict \ Experimental	Positive	Negative
Positive	8	22
Negative	2	968

**Table 2-2: Imaginary outcome of classifier B**

Predict \ Experimental	Positive	Negative
Positive	2	16
Negative	8	974

Classifier A and B have identical prediction accuracy as they produce the same number of TP and TN. So the simple classification accuracy is  $976/1000 = 97.6\%$ . Nevertheless, classifier A is far superior to classifier B because A discovers 8 true positives (true active compounds or hits) and B only has 2. Furthermore, classifier A is more precise than B. Among the positives identified by A, 8 out of 30, which is 26.7%, are truly active; while only 11.1% of positives identified by B are truly active. According to Matthews correlation coefficient, the MCC for classifier A is 0.454, while the MCC for classifier B is 0.138. Thus, MCC is more informative than simple classification accuracy for unbalanced data sets.

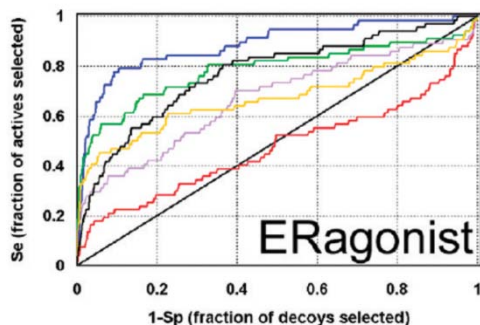
Besides MCC, sensitivity (or recovery rate, recall rate) and precision (purity) are frequently mentioned in scientific articles:

$$\text{Sensitive} = TP / (TP + FN)$$

$$\text{Precision} = TP / (TP + FP)$$

More formally, classifier A is more sensitive than classifier B, based on Table 2-1 and Table 2-2. The classifier that brings more true active compounds or “hits” is always desirable since it is more likely to develop a lead compound out of it. In addition, classifier A is more precise than classifier B. A high-precision classifier brings more true “hits” in experimental validation and saves the cost of bioassay. Sensitive and precision are usually contradictory. Sensitivity can be increased by labeling more testing

compounds as active. Sensitivity can reach 1.0 by label all testing compounds as active, since  $FN = 0$ . On the other hand, enhancing the sensitivity usually sacrifices precision as the classifier brings in more false positives and increases the cost of experimental validation. Thus, a balanced trade-off should be determined between these two quantities. Based on this, it is inappropriate to judge the quality of a classifier by simply investigating either sensitivity or precision.



**Figure 2-7: Sample ROC curve.**

ROC curve plots sensitivity versus 1-specificity by adjusting decision boundaries. This figure illustrates the ROC curves of different virtual screening models.<sup>27</sup>

ROC curve (receiver operating characteristic) is commonly plotted to measure the performance of several models graphically.

Figure 2-7 shows a typical ROC curve for evaluating several virtual screening models. The X-axis of a ROC plot is false positive rate, and the Y-axis is sensitivity. False positive rate (FPR) is equal to  $FP / (FP + TN)$ . ROC curve tells how false positive rate responds to the change in sensitivity. For a specific virtual screening model, the sensitivity can be only enhanced by reducing prediction threshold (such as docking score) and predicting more compounds as active. This approach usually increases FP but reduces TN, resulting in higher FPR. Thus, false positive rate monotonically increases with sensitivity. A random classifier will show a diagonal ROC curve, while an effective classifier can enrich positives and achieve high sensitivity at low false positive rate. A typical characteristic of a ROC curve from an effective screening model is the relative large accumulated area under the curve. Therefore, AUC (area under curve) is a common measure to assess the performance of a screening model.

Many metrics are currently used to assess the performance of ligand classification models, but no one is perfect. For example, ROC curve is a poor indicator for early enrichment of active compounds<sup>28</sup>. The performance metric should be carefully selected in order to distinguish optimal models effectively.

### 2.1.5 Limitation of QSAR

Although quantitative structure-activity relationship (QSAR) has been widely applied in drug discovery, ranging from virtual screening for lead compound identification, to early-stage drug development for the optimization of ADMET properties, the reliability and predictability of QSAR models are still under hot debate<sup>28</sup>. As pointed out by Truchon et al, the major reason why QSAR fails is attributed to the vast number of equivalent models and deficient external validation. In other words, it is because QSAR overfits the training data without detecting the true structure-activity relationship.

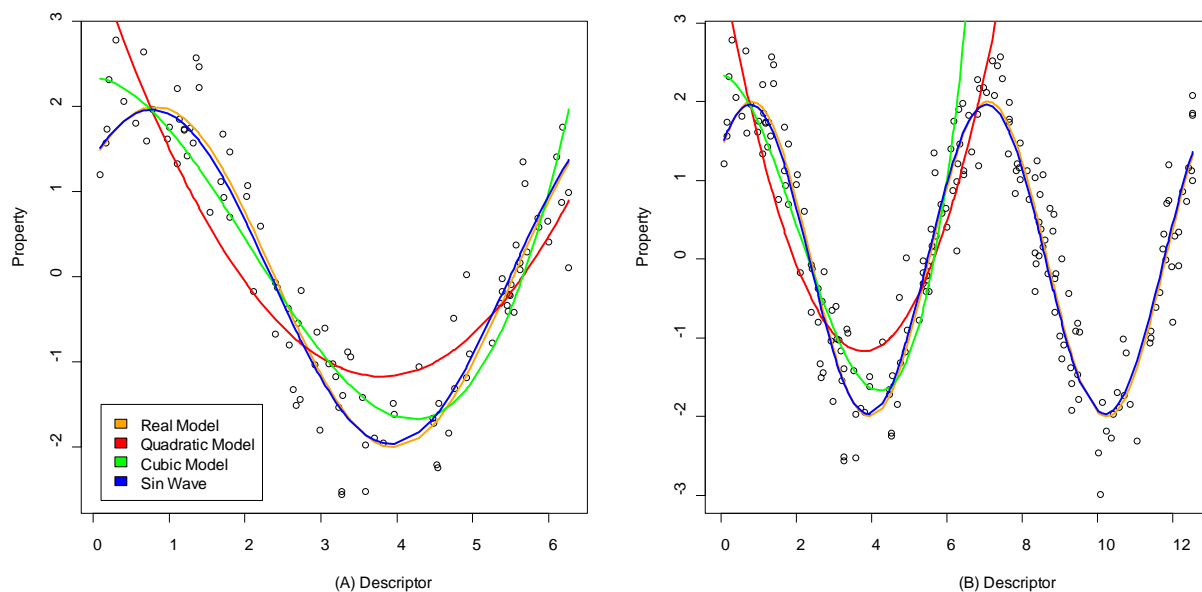
QSAR models retain a limited scope of application. The uncertainty and variance are expected for predictions made beyond the scope. Figure 2-8 illustrates this concept through another “toy” model. In Figure 2-8, X-axis and Y-axis are imaginary descriptor and property of interest, respectively. The data points in Figure 2-8 (A) are generated according to the following equation:  $y = 2 \times \sin(x + 0.75) + \epsilon$ , where  $\epsilon \sim \text{Norm}(0, 0.5)$ . Three candidate functions are developed to fit the data points:

- a) Quadratic function:  $y = ax^2 + bx + c$
- b) Cubic function:  $y = ax^3 + bx^2 + cx + d$
- c) Sine function:  $y = a \sin(x + b)$

Parameters in quadratic and cubic functions are solved using standard linear regression analysis, with fitted  $R^2$  being 0.65, 0.85 respectively. The parameters in the sine function are solved using Newton–Raphson method, with fitted  $R^2$  being 0.89. In Figure 2-8 (A), the data points are well fitted by the cubic function where  $x \in [0, 6]$ , but neither cubic nor quadratic function shows relevant predictions on the testing data set  $(x, y)$  where  $x \in [6, 12]$  in Figure 2-8 (B).

In QSAR, good predictions are expected for testing compounds that are similar to the training set. For example, testing compounds are supposed to be aligned with the training set in CoMFA. In fact, testing compounds usually do not share similar scaffold or substitutes with the training compounds. In this case, not much knowledge is accessible regarding these new structures. Similar to the case described in Figure 2-8, QSAR model may fail when it is applied on the chemistry space beyond its modeling scope.





**Figure 2-8: A “toy” mathematical model illustrating model applicability**

X-axis denotes certain variable to predict the property value along Y-axis. The property value is generated by adding random noise to the orange line (real model). Then, quadratic model, cubic model and sin wave are used to fit the training data in the left figure. Even if cubic model fits training data well, its prediction is irrelevant when the testing data is beyond the range of training data (right figure).

## 2.2 REVIEW OF MOLECULAR DESCRIPTORS

Molecular descriptors transform chemical structures to numerical vectors, upon which QSAR models are developed. Hundreds of molecular descriptors have been proposed for solving physicochemical and biological properties. The descriptors can be grouped according to dimension invariance, ranging from 0D to 4D descriptors. The invariance of molecular descriptors in N-dimension means that any structural changes in higher dimension do not affect the structural representations in N-dimension. For example, conformational isomers have identical 2D representations and descriptors. 0D descriptor is calculated based on molecular formula. To calculate 1D descriptor, a structure is represented by a set of linearly connected structural fragments, e.g. A-B-A-C where A, B, C denote certain fragments. In 2D descriptor generation, a chemical structure is transformed into a connectivity matrix, in which the diagonal elements describe atom types and the off-diagonal elements describe nominal bond types. Then topological or connectivity indices can be calculated conforming to the matrix. 3D descriptors require 3D coordinates of

each atom, so they are sensitive to 3D conformations. 4D descriptors characterize 3D molecular conformations as time evolves. They may be calculated from the trajectories of molecular dynamics simulation or ensemble sampling. The table below lists some representative descriptors for each dimension.

**Table 2-3: List of representative molecular descriptors**

Dimensionality	Example	Reference
<b>0D</b>	Molecular weight, Number of bonds, Atom counts	<sup>29</sup>
<b>1D</b>	The count of functional groups, such as H-bond donors, heterogeneous ring, etc.	<sup>29</sup>
<b>2D</b>	Topological indices, Atom paths, Burden eigenvalues	<sup>30-32</sup>
<b>3D</b>	Polar surface area, Volume, Electrostatic field, 3D-MoRSE	<sup>33</sup>
<b>4D</b>	Volsurf, 4D-fingerprints	<sup>34 35</sup>

Another way of classifying molecular descriptors rests on the depicted structural characteristics, including constitutional descriptor, geometric descriptor, topological descriptor and electrostatic/quantum-chemical descriptors. The following subsections will focus on the theory and application of these descriptors.

### **2.2.1 Constitutional Descriptor**

Constitutional descriptors depict the composition of a compound structure and capture the properties of the elements that constitute the structure. Generally regarded as 0D, constitutional descriptors are independent on the geometry and topology of a structure. They are calculated either based on the occurrence of different types of atoms, bonds, x-member rings, or as a mathematical function of these elements. Table 2-4 lists some popular constitutional descriptors involved in QSAR studies.

**Table 2-4: List of representative constitutional descriptors**

Molecular Weight	Number of X-member Rings
Total Number of Atoms	Number of Aromatic Rings
Numbers of Atoms of Different Identity	Number of H-bond Donors
Number of Bonds	Number of H-bond Acceptors
Number of Rotatable Bonds	Sum of Atomic Polarity
Numbers of Double, Triple or Aromatic Bonds	Sum of Atomic van der Waals Volumes

A significant advantage of constitutional descriptors is the ease of calculation and interpretation. Despite their simplicity, constitutional descriptors exhibit relevance to many pharmacological properties. Particularly, descriptors characterizing oxygen and nitrogen atoms have been shown effective for deriving ADMET models, because these atoms affect hydrogen bonding and molecular polarity that are related to solvation and absorption<sup>36</sup>. Constitutional descriptors are usually insufficient to build any QSAR or QSPR model independently. Instead, they serve as complement to 2D or 3D descriptors in modern data mining algorithms. Table 2-5 lists some constitutional descriptors and their application for the prediction of physiochemical properties, binding affinity and ADMET properties.

**Table 2-5: List of constitutional descriptors and their applications**

<b>Descriptor</b>	<b>Application</b>	<b>Reference</b>
<b>number of carbon atoms</b>	partition coefficients of barbituric acids	37
<b>relative number of nitrogen atoms</b>	pKa for neutral and basic drugs	38
<b>relative number of benzene rings</b> <b>number of carbon atoms</b> <b>relative number of double bonds</b>	biological activity of carbonic anhydrase CA II inhibitors	39
<b>number of sulfur atoms</b> <b>number of H-bond donors</b> <b>number of hydroxyl groups</b>	general solubility	40
<b>number of fluorine atoms</b>	gas chromatographic retention index	41
<b>number of acceptor atoms for H-bonds</b>	Henry's law constant	42
<b>the relative number of single bonds</b>	logk of peptides in HPLC	43
<b>number of Cl atoms</b>	boiling point	44
<b>atom number in the hydrophobic-hydrophilic segment</b>	critical micelle concentration	45
<b>relative number of aromatic bonds</b>	classification of the oxindole-based inhibitors of cyclin-dependent kinases	46
<b>number of all atoms</b> <b>number of all bonds</b> <b>number of aromatic atoms</b> <b>number of hydrophobic atoms</b> <b>fraction of rotatable bonds</b>	Caco-2 permeability	47
<b>47 constitutional descriptors involved in principal component analysis</b>	carcinogenic activity	48
<b>number of carboxylic acids</b>	drug intestinal absorption	49

### 2.2.2 Geometric Descriptor

The relevance between molecular geometric configuration and bioactivity or physiochemical properties is well recognized. For receptor-ligand interaction, the geometric arrangement of ligands influences whether the ligands can fit into the receptor pocket. To capture these molecular features, geometric descriptors are designed to be a set of mathematical functions of molecular size, shape, position and properties in space. The calculation of geometric descriptors is not as straightforward as constitutional descriptors, since it involves mathematical integration and numerical approximation. The interpretation of some descriptors requires thorough understanding of underlying theories. Table 2-6 shows some prevailing geometric descriptors for the investigation of molecular conformation and ligand-receptor interaction. Brief introduction to some of these descriptors is given below.

**Table 2-6: List of representative geometric descriptors**

Molecular Volume	Molecular Surface Area
Charged Partial Surface Area	Solvent-accessible Molecular Surface Area
Gravitational Indices	Electric Quadrupole Moments
Principal Moments of Inertia	Sphericity/Asphericity Index
Shadow Areas of a Molecule	Linearity Index
Distance Between Atom Pairs	WHIM shape index

Molecular volume, as suggested by its name, is the total volume enclosed by van der Waals surface. Molecular volume calculation is not simply the summation of the volume of every atom, since covalently bonded atoms have significant volume overlapping. Thus, the overlapping volume of adjacent atom pairs (atom identity, bond type, conjugate system, etc.) should be considered.

Molecular surface area determines the interactions between the molecule and its surrounding area. Molecular surface area as a geometric descriptor has been found statistically correlated with water solubility, octanol-water partition coefficients, activity coefficients and boiling points<sup>50</sup>. The surface area refers to the area of van der Waals surface. Some derived descriptors show better biological relevance, such as water-accessible surface area, polar surface area, etc. The surface area can be approximated by summing either the areas of triangles of a defined surface or the contributions from all atoms. In addition,

efficient algorithms, developed by Pearlman, Lee and Richards, and Hermann, for the approximation of surface area have caught the most attention.

**Table 2-7: List of geometric descriptors and their applications**

Descriptor	Application	Reference
<b>gravitational index</b>	boiling point	51
<b>molecular surface area</b>	octanol/water partition	52
<b>molecular volume</b>	coefficients	
<b>electric quadrupole moments</b>	alignment-free 3D QSAR	53
<b>principal moments of inertia</b>	modeling inhibition of cytopathic effects of HIV-1	
<b>Linearity index (L/B index)</b>	classification of carbonic anhydrase inhibitors	54
<b>partial positive surface area</b>	biological activity of carbonic	39
<b>partial negative surface area</b>	anhydrase CA II inhibitors	
<b>WHIM descriptors</b>	toxicity of heterogeneous chemicals LogP, Caco-2 permeability	55 56

Sphericity (or asphericity) index is a measurement of the degree in which the shape of a molecule is similar to a sphere. The calculation is based on molecular volume and surface area. The sphericity index of a molecule is defined as the ratio of the surface area of a sphere with the same volume to the surface area of the molecule. As a sphere has the lowest surface-area-to-volume ratio, the more “round” a molecule is, the larger its sphericity index becomes. The maximum of a sphericity index is 1.

Gravitational index is defined as  $G = \sum_{i < j} \frac{m_i m_j}{r_{ij}^2}$ , where  $i$  and  $j$  are the atom indices,  $m_i$  is the mass of atom  $i$ . Gravitational index is bond-restricted when the summation is only calculated on bonded atom pairs<sup>51</sup>. This descriptor characterizes the mass distribution of a molecule. Principal moments of inertia are also mechanical descriptors, describing the resistance to changes in rotation. Similar to dipole moments, electric quadrupole moments describe the distribution of electric charges over a molecule.

Quadrupole moments have higher order than dipole moments. A molecule that has zero dipole moment may still have quadrupole moment, as if a molecule that has zero net charge may still have dipole moment.

Geometric descriptors have been widely used in QSAR and QSPR modeling. Table 2-7 lists some of the representative studies. The major disadvantage of most geometric descriptors is the dependence on three-dimensional conformations. The calculation of geometric descriptors however mainly relies on internal coordinates so that they are independent on molecular position and orientation. This valuable feature enables us to build alignment-free 3D QSAR models.

### 2.2.3 Topological Descriptor

Topological descriptor, also known as connectivity index, emphasizes what atoms are connected by covalent bond and how they are connected. To generate topological descriptors, compound structures are usually treated as molecular graph with suppressed hydrogen atoms. Topological descriptor is usually graph invariant, meaning that the descriptor value remains constant regardless of how the structure is represented. Topological descriptors have been employed in numerous QSAR and QSPR studies, but their interpretation is usually obscure in the context of physicochemical properties. Besides its graph invariance, topological descriptor exhibits another apparent advantage, the simplicity in computation. Table 2-8 lists some popular topological descriptors involved in many studies. Wiener index and BCUT descriptors will serve as examples for descriptor interpretation. And a few representative applications of topological descriptors are listed in Table 2-9.

**Table 2-8: List of representative topological descriptors**

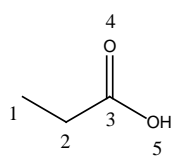
Wiener Index	Hosoya Index
Balaban J Index	Randić Index
Structural Information Content Index	Bonding Information Content Index
Kappa Shape Index	Connectivity Index
BCUT Descriptors	Molecular Path/Walks

Wiener index is a classical quantity to characterize the connectivity of a graph. The formal definition is given by the following equation:

$$W = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n d_{i,j}$$

Where  $d_{i,j}$  is the shortest distance between point  $i$  and point  $j$ , or atom  $i$  and atom  $j$  in a molecular graph. To restate this equation by words, Wiener index is equal to the sum of the nearest distance between any pair of atoms. Wiener index can be also interpreted in terms of each individual covalent bond. In acyclic case, the contribution of each bond is the product of the total number of atoms at each side, and the Wiener index is simply the sum of the contribution of all the bonds. If both sides of a bond show similar number of atoms, the product tends to be large, because the total number of atoms is constant and  $a \times b \leq (\frac{a+b}{2})^2$ . Intuitively, the index is mainly affected by the bond located in the center of a structure. In other words, the topology of side branches weights relatively less in the index calculation. A structure with a large number of atoms tends to have a large Wiener index value. Thus, the index is potentially correlated with van der Waals surface area. Examples and explanations of Wiener index and some other descriptors are given by Randic and Zupan.<sup>57</sup>

BCUT (Burden CAS University of Texas) descriptors<sup>58-61</sup> incorporate comprehensive information regarding molecular structure, atom property and more into decimal numbers. Creating BCUT descriptors is one of the most popular approaches to construct low-dimensional chemistry space and perform diversity analyses. Briefly, BCUT descriptors are defined by combining atomic descriptors for each atom and description of the nominal bond-types for adjacent and nonadjacent atoms into BCUT matrices. The value of each chemistry-space coordinate is specified as the highest or lowest eigen value of BCUT matrix as illustrated in Figure 2-9.



	1	2	3	4	5
1	-0.047	1.1	0.01	0.01	0.01
2	1.1	0.046	1	0.01	0.01
3	0.01	1	0.231	2.1	1.1
4	0.01	0.01	2.1	-0.371	0.01
5	0.01	0.01	1.1	0.01	-0.293

**Figure 2-9: Illustration of BCUT connectivity matrix (Tripos).**

In the matrix, diagonal elements are filled by atom charge, and off-diagonal elements are filled by bond order if any. The remaining elements are filled by a predefined constant (0.01). BCUT descriptor is either the lowest or the highest eigen value of this connectivity matrix.



In Figure 2-9, the diagonal elements of the matrix are filled with estimated partial charge of each atom. The off-diagonal elements can be a small number, e.g. 0.01 in this example, if two atoms are not connected. Otherwise, a nominal value is filled to represent covalent bond type. An increment (0.1) is added to the nominal bond value if the bond connects terminal atoms. Then, the off-diagonal elements may be scaled by a preset factor. Finally, the lowest or the highest eigen value of the connectivity matrix becomes BCUT descriptor of this query structure. BCUT descriptors have many variants, depending on atomic property filling the diagonal elements and the scaling factor. Generally, the optimal subsets of BCUT descriptors capture the maximum variance of a compound library, and show little pair-wise correlation.

**Table 2-9: List of topological descriptors and their applications**

Descriptor	Application	Reference
<b>BCUT</b>	Library Design, QSAR, Diversity Analysis	62-64; 65, 66
<b>connectivity chi and kappa indices</b>	Prediction of HPLC Retention Index	67
<b>Wiener index</b>	Drug-receptor interaction	68; 69
<b>triplet index, Balaban's J index, information contents, Wiener index, etc.</b>	mutagenic potency of aromatic and heteroaromatic amines	70
<b>Edge-Connectivity Index, Hosoya Index, Wiener index, Randic's index, Balaban's Index, etc.</b>	The study on the relation between topological descriptors and physicochemical properties of Octanes, such as boiling point, molecular volume	71

### **2.2.4 Electrostatic/Quantum-Chemical Descriptor**

The prediction of bioactivity and physicochemical properties of drug candidate compounds has become more and more important in modern medicinal chemistry. The correlation between structure and property is established by mapping molecular descriptors to numeric values. Obviously, the quality and accuracy of descriptors are deterministic for QSAR and QSPR studies. The advance of computer hardware and computation theory turns the generation of quantum-chemical descriptor feasible, but practical *ab initio* calculation is based on orbital approximation. Semi-empirical methods may also accelerate the calculation of quantum-chemical descriptors.

Quantum-chemical descriptors encode unique molecular information that is hardly covered by other descriptors, including atomic charges, HOMO and LUMO energies, orbital electron densities, superdelocalizabilities, polarizabilities, dipole moment and total energies<sup>72</sup>. All of these descriptors are constantly employed in modeling bioactivities and various ligand properties. For example, LUMO energy, HOMO energy and dipole moment are involved in the prediction of oral drug absorption<sup>73</sup>. The theoretical atomic charges are involved in modeling congeneric and non-congeneric  $\alpha_1$ -adrenoceptor antagonists<sup>74</sup>. In Tuppurainen et al.'s study, electron density is shown to be predictive for mutagenicity<sup>75</sup>. Atom-atom polarizabilities and molecular polarizabilities contribute to the activity of a diverse set of enzyme activators<sup>76</sup>. These case studies demonstrate the wide applicability of quantum-chemical descriptors in QSAR and QSPR modeling. The descriptors provide additional information at atomic and orbital level, which is relatively more accurate than empirical molecular descriptors. And quantum-chemical descriptors generally have physical meaning. A comprehensive variable selection, or descriptor selection, is mandatory to build and refine a QSAR/QSPR model, especially for some unexplored properties and novel compound scaffolds.

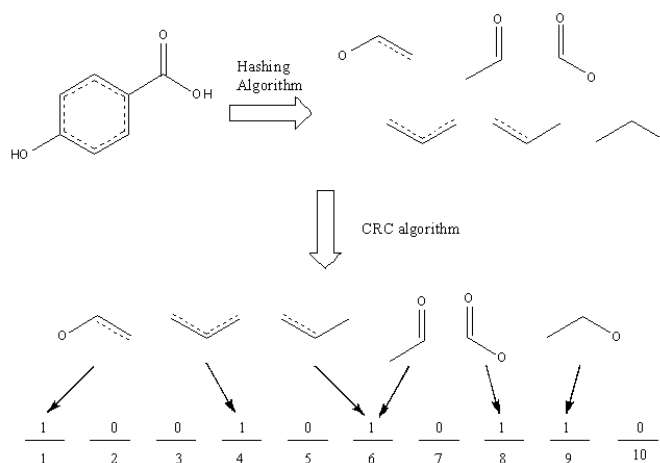
### **2.2.5 Hybridized Descriptors and Molecular Fingerprint**

Molecular fingerprints are probably the most frequently used descriptors in modern QSAR/QSPR modeling. Broadly speaking, molecular fingerprint could be a high dimensional categorical or real-value vector that encodes a diverse set of structural patterns and derived properties. Mapping chemical structures to desired molecular properties is challenging and complicated. One of the difficulties is the selection of appropriate descriptors. Molecular fingerprint, a high dimensional descriptor, is originally designed to discriminate and screen compounds in a compound inventory. As it encodes a variety of structural patterns, fingerprint offers great possibilities to build quantitative prediction models.

Nevertheless, the risk also exists due to potential overfitting. Recent publications report the application of molecular fingerprints for the prediction of ligand bioactivity, ADMET properties, selectivity, physicochemical properties, etc. Fingerprints are also the prevailing descriptor for traditional topics in cheminformatics, such as similarity calculation, virtual screening, library design, diversity analysis, etc. Majority of the chapters covered in this thesis are related to molecular fingerprints.

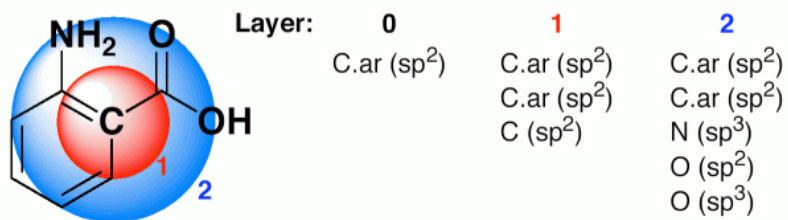
According to representation, molecular fingerprints can be classified into the fixed-length fingerprints and dynamic fingerprints. In the former case, the length of the fingerprint is usually preset, and the pattern that each dimension encodes is predefined. The length of the fingerprints ranges from hundreds to thousands. For example, the classical MACCS key has 166 bits; Unity fingerprint has 992 bits. The length of dynamic fingerprint depends on the presented compound library because the number of possible patterns may be very large or even infinite. These fingerprints are usually represented in sparse arrays, but they are sometimes folded into regular vectors for simplicity. ECFP (extended-connectivity fingerprint) from ChemAxon is an example. Each dimension of molecular fingerprints can be binary (0 or 1) indicating the presence or the absence of a feature, integers indicating the occurrence of a feature, or real-value properties. For example, MACCS key is a binary structural key, and LINGO counts the occurrence of each present feature.

Molecular fingerprints can also be categorized by encoded features, including pre-defined structural keys, atom environment, atom path, and pharmacophore keys. To generate pre-defined structural keys, computer program scans a query compound structure to search for a list of structure patterns. Figure 2-10 illustrates the scheme of Unity fingerprint in Tripos Sybyl software suite. In Figure 2-10, the compound structure is “chopped” into fragments that are defined in a fragment dictionary. These fragments are further mapped to different fingerprint “bits” by cyclic redundancy check algorithm. MACCS key 166 and PubChem fingerprints are also typical examples.



**Figure 2-10: Illustration of the coding scheme of Unity fingerprint by Tripos.**  
 Each dimension of the molecular fingerprint represents the presence or absence of predefined molecular fragments, following hashing algorithm and CRC algorithm.

Atom environment fingerprints are mostly favored in recent publication for its optimal performance. These fingerprints describe atoms together with their surrounding environment in a structure. Atoms may be differentiated by the hybridization, and environment may be defined by connected atoms. Molprint 2D fingerprint<sup>77</sup> and ChemAxon extended-connectivity fingerprint (ECFP) are famous examples. The feature in Figure 2-11 can be translated as a  $sp^2$  carbon atom that is surrounded by another  $sp^2$  carbon atom and two aromatic carbon atoms at one-bond distance away, and that also has two  $sp^2$  carbon atoms, one  $sp^3$  nitrogen atom, one  $sp^2$  oxygen atom and one  $sp^3$  oxygen atom at two-bond distance away. In Molprint 2D, atoms are differentiated by Sybyl atom types. ECFP has a similar concept but discriminates atoms in a more detailed level.



**Figure 2-11: A graphical representation of a structure pattern in Molprint 2D fingerprint.**

Figure reference<sup>77</sup>

FP2 fingerprint in OpenBabel and Daylight fingerprint belong to path-based fingerprints. In these fingerprints, compound structures are treated as graphs. So atoms are vertices and bonds are edges. An “atom walk” is the movement from one atom to another if they are connected by any covalent bond. An “atom path” is a trajectory of several “atom walks” without revisiting any bonds. The types of atoms and bonds visited in an atom path uniquely define a structure pattern. Pharmacophore fingerprints translate atoms in a structure into pharmacophore tags, such as H-bond donor, H-bond acceptor, hydrophobic group, formal positive charge, formal negative charge, and etc. Pharmacophore fingerprints generally encode the pharmacophore groups in a structure and distance between them. PharmPrint<sup>78</sup> and TGT fingerprint belong to this category. The changes in 3D conformation generally affect pharmacophore fingerprints, opposed to other 2D fingerprints.

Many free computer programs are available to generate molecular fingerprints, such as Molprint 2D, FP2, MACCS key, PubChem fingerprints. Molecular Operating Environment (MOE) and Discover Studio are two popular proprietary drug design programs for fingerprint generation. The successful extended-connectivity fingerprint is implemented in JChem module from ChemAxon. Some fingerprints need to be generated by in-house scripts, e.g. PharmPrint. Researchers may also design structure patterns by SLN and carry out substructure search to generate customized fingerprints.

Commonly, any high dimensional descriptor vectors that are sufficient to differentiate compound structures are molecular fingerprints. This coding strategy accelerates similarity calculation and substructure search for large chemical databases. Combined with robust machine learning algorithms, fingerprints have become powerful descriptors applied in almost all the fields in QSAR/QSPR modeling. Table 2-10 lists some of the application of molecular fingerprints that are introduced in this section.

**Table 2-10: Example applications of molecular fingerprints**

Fingerprints	Application	Reference
<b>MACCS</b>	Similarity search, virtual screening on ChemBL	79
	Cytotoxicity Prediction	80
	Drug-likeness	81
	ADME property	82
<b>Daylight</b>	Structure ranking and virtual screening	83
	Drug/compound clustering	84
		85
	Classification of kinase inhibitors	86
	Generation of diverse screening library	87
<b>ECFP</b>	Lipophilicity	88
	Melting Point and Aqueous Solubility	89
	QSAR for kinase inhibition	90
	Prediction of biological target	91
<b>TGT</b>	Structure ranking and virtual screening	83
	Virtual screening and QSAR	92

## 2.3 STATISTICAL MODELING AND MACHINE LEARNING

Statistical modeling and machine learning tools, the bridge that connects molecular descriptors and molecular properties, are an important component in modern QSAR/QSPR studies. Data mining on the vast collection of descriptors determines the optimal descriptor subset that correlates well with properties of interest. In addition, parameters in QSAR/QSPR models are solved according to machine learning algorithms and numerical optimization. Besides quantitative and qualitative prediction, computational learning theory, a subfield in machine learning, guides researchers in model selection and the estimation of generalization error. This section highlights classical machine learning algorithms, model hypotheses, and applicability in cheminformatics.

### 2.3.1 Linear Regression

Linear regression assumes that one or more explanatory variables ( $x$ ) have linearly additive contribution to a response scalar variable ( $y$ ). However, a normally distributed noise in  $y$  cannot be explained by  $x$ . To express this mathematically,

$$y = \beta_0 + \sum_{i=1}^K \beta_i x_i + \varepsilon$$

$\varepsilon$  is an error term, and it follows normal distribution;  $\beta$  is coefficient. The predicted  $y$ , denoted by  $\hat{y}$ , is equal to  $\hat{y} = \beta_0 + \sum_{i=1}^K \beta_i x_i$ . The establishment of a linear model is the parameterization of  $\beta$  based on training data. Once  $\beta$  is solved, the prediction of  $y$ ,  $\hat{y}$ , can be easily obtained from  $x$ . The motivation of solving  $\beta$  is to maximize the likelihood of observations, or to minimize the squared sum of error  $\varepsilon$ . Let matrix  $X$  represent the training set. Each row of  $X$  is a single observation, and each column corresponds to variable  $x_i$ . Thus, a training set that contains  $K$  dimension variables and  $N$  samples can be represented by matrix,  $X$ , which has  $N$  rows and  $K + 1$  columns. The estimation of  $\beta$  is  $\hat{\beta} = (X'X)^{-1}X'y$ , where  $\beta$ ,  $y$  are coefficient vector and response vector. The most basic requirement of solving a linear model is that matrix must have a full rank. In other words, there must be at least  $K + 1$  linearly independent samples to solve  $\beta$ .

Linear regression has been used to model physicochemical properties and ligand bioactivities. For example, the famous Free-Wilson QSAR model<sup>93</sup>:

$$LD_{50} = \mu + a[H] + a [CH_3] + b[N(CH_3)_2] + b[N(C_2H_5)_2]$$

Free and Wilson are the pioneers in QSAR research, and their Free-Wilson model initiated a new branch of drug discovery and suggested a quantitative approach for structure-activity study. In this equation, bioactivity is the sum of a bias value,  $\mu$ , the contribution of H group or  $CH_3$  group at R1 position and the contribution of  $N(CH_3)_2$  or  $N(C_2H_5)_2$  at R2 position. This model can be easily transformed into linear regression analysis, in which a four-element binary vector  $x$  represents the presence and absence of the functional groups at R1 and R2 position, and coefficient  $\beta$  represents the bias value and the contribution of each group.

Linear regression analysis relies on a few assumptions, including linearity, normally distributed residual and constant variance. Even though the criteria do not necessarily hold in all occasions, the numerical optimization still minimizes the discrepancy between observed value and predicted value. Running some diagnosis on the regressor may improve fitting. For cheminformatics applications, variable transformation is a common remedy. For example, transforming bioactivity into logarithm scale is a popular practice in QSAR modeling. Another challenge in linear regression is variable selection. The high

degree of freedom tends to over-fit the training data, but yields poor performance in future prediction. Ridge regression, the Lasso, and stepwise regression are the techniques for bias-variance tradeoff.

### 2.3.2 Logistic Regression

Linear regression introduced in the previous section is a popular technique to predict real-value response. Sometimes, the desired response may be categorical label, such as active or inactive, soluble or insoluble. Besides predicting real-value response, linear regression could also be carried out for classification by assigning symbolic numbers to different categories. This approach however presents a few challenges to the assumptions of linear regression: first, the prediction may be inadmissible due to some extreme variables,  $x$ ; second, the distribution of categorical labels is expressed as a function of explanatory variables ( $x$ ), so the variance is not uniform or follows normal distribution. These impair hypothesis testing of coefficients, construction of confidence interval, variable selection, etc., as all the inferences assume normality.

Logistic regression is regarded as generalized linear regression. It models the conditional probability of observing  $y$ , given its explanatory variables. It expresses the logit of the probability, or the log odds, as a linear combination of  $x$ :  $\log\left(\frac{P(y)}{1-P(y)}\right) = \beta_0 + \sum_{i=1}^K \beta_i x_i$ . Thus,

$$P(Y = y|X) = \frac{\exp(\beta_0 + \sum_{i=1}^K \beta_i x_i)}{1 + \exp(\beta_0 + \sum_{i=1}^K \beta_i x_i)}$$

With one dimension variable  $x$ , logistic regression produces a sigmoid curve, as shown in Figure 2-12. The predicted category is the one that has larger than 0.5 probabilities.

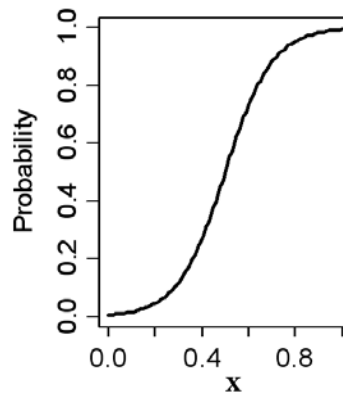


Figure 2-12: Sample sigmoid curve produced from logistic regression



The solution to coefficient vector  $\beta$ , has no analytical form. Instead,  $\beta$  is found numerically by maximizing the likelihood:

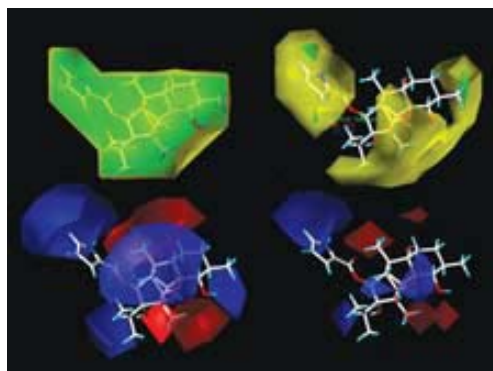
$$\hat{\beta} = \underset{\beta}{\operatorname{argmax}} \prod_{i=1}^N \left( \frac{\exp(\beta_0 + \sum_{i=1}^K \beta_i x_i)}{1 + \exp(\beta_0 + \sum_{i=1}^K \beta_i x_i)} \right)^{y_i} \left( \frac{1}{1 + \exp(\beta_0 + \sum_{i=1}^K \beta_i x_i)} \right)^{1-y_i}$$

A popular algorithm of finding  $\hat{\beta}$  is called reweighted least square, which is an iterative procedure. And the convergence is shown to be fast.

Linear regression can be used to predict the real-value bioactivity, while logistic regression is designed to predict binary outcome. For example, Cronin et al.<sup>94</sup> uses logistic regression to classify compounds according to whether they have antibacterial activity. In their study, multiple physicochemical descriptors are involved in logistic regression model. As discussed in their article, logistic regression develops “interpretable and transparent” models.

### 2.3.3 Partial Least Square

Comparative Molecular Field Analysis (CoMFA), developed by Cramer et al and published in JACS in 1988, is probably the earliest and most influential modeling method in QSAR, especially in 3D QSAR. After a decade, CoMFA and its derivative methods are still frequently seen in major cheminformatics journals. At the beginning stage, Tripos Sybyl implemented the CoMFA methodology and gained significant market share in commercial computer-aided drug design platform. As a renovated QSAR technology, its success is mainly attributed to a novel statistical method: partial least square regression. All the QSAR algorithms correlate activity with structure descriptors, and CoMFA is not an exception. The descriptor in CoMFA is the steric energy and electrostatic energy in an aligned 3D grid. According to a set of training compounds and their labeled activity, CoMFA figures out favored field energy for ligand-receptor interaction. Figure 2-13 shows a graphical illustration of a developed CoMFA model.



**Figure 2-13: A graphical representation of molecular field component in CoMFA studies.**  
A CoMFA model depicts favored and disfavored functional groups in 3-D space.

Linear regression and logistic regression are inappropriate techniques in CoMFA. In CoMFA, hundreds of energy field values are calculated in a 3-dimensional grid, but labeled training compounds are usually less than a hundred. Linear regression cannot solve the coefficients when the degree of freedom is higher than the number of constraint. In other words, the variable matrix,  $X$ , is not full rank. Before the advent of partial least square regression (PLSR), principal component analysis (PCA) was a common dimension-reduction technique. PCA generates a few components that are linear combinations of original descriptors in order to capture the major variance in the distribution. Nevertheless, the major variance does not necessarily correlate with some response of interest. For example, minor modification to functional groups may induce significant change in bioactivity, but ligands in different scaffold may have similar biological profiles. On the contrary, PLSR calculates the principal components that correlate well with response values. Anyhow, PCA is in the domain of unsupervised learning, while PLSR is a supervised learning algorithm.

Partial least square regression (PLSR)<sup>95</sup> approximates the original matrix by the hidden variable  $T$ :

$$X = TP' + E$$

$$Y = TC' + F$$

where  $X$  is  $n \times m$  descriptor matrix,  $T$  is  $n \times l$  score matrix,  $P'$  and  $C'$  are  $l \times m$  and  $l \times 1$  loading matrices,  $E$  and  $F$  are error terms. In this case,  $Y$  is only a vector with selectivity label or activity ratio for each compound.  $T$  is solved through iterative procedure.

For each round  $k$ ,  $k=1,2,\dots,l$

1. Find vector  $w_k$  that maximizes the covariance and score vector  $t$ :

$$w_k = \arg \max_w \text{cov}(Xw, y) \text{ s.t. } w'w = 1, t_k = \frac{Xw_k}{\|Xw_k\|}$$

2. The best approximation of matrices X and Y is  $t_k p_k'$  and  $t_k c_k'$  through

$$p_k = \min_p \|X - t_k p\|; c_k = \min_c \|Y - t_k c\|$$

$p_k$  will be the  $k^{\text{th}}$  column of matrix P,  $c_k$  will be the  $k^{\text{th}}$  column of matrix C and  $t_k$  will be the  $k^{\text{th}}$  column of score matrix T

3. Replace matrix X and Y by their residuals, which are the difference between original X, Y and their approximation by  $p_k, c_k$  and  $t_k$

The final regression equation is  $y = x'b$  where  $b = W(T'XW)^{-1}T'y$ .  $b$  is solved by minimizing the training error  $\min_v \|TP'Wv - TC\|^2$  and  $b = Wv$ . The only variable  $l$ , the number of components, can be calculate through leave-one-out cross validation.

The invention of PLSR is a “landmark” in statistics as it is probably the first algorithm motivated by pharmaceutical science, and it plays a central role in CoMFA. The theory on force field and field energy calculation was well established before CoMFA was developed. It was PLSR that turned modeling bioactivity through field energy practical. From CoMFA, the importance of quantitative modeling techniques can be easily observed.

### 2.3.4 Naive Bayes Classifier

Bayes' theorem states that the posterior probability is proportional to the prior and likelihood:

$$P(Y = y | \mathbf{x} = x_1, x_2, \dots, x_K) = \frac{P(\mathbf{x} = x_1, x_2, \dots, x_K | Y = y)P(Y = y)}{\sum_Y P(\mathbf{x} = x_1, x_2, \dots, x_K | Y)P(Y)}$$

or

$$P(Y = y | \mathbf{x} = x_1, x_2, \dots, x_K) \propto P(\mathbf{x} = x_1, x_2, \dots, x_K | Y = y)P(Y = y)$$

$P(Y)$  is the prior,  $P(\mathbf{x} | Y)$  is likelihood and  $P(Y | \mathbf{x})$  is posterior likelihood.

Bayes' theorem enables us to calculate the probability of Y given its explanatory variables  $\mathbf{x}$ , based on the prior and likelihood. Similar to logistic regression, this theorem is suitable for ligand-based drug design, such as virtual screening or ligand classification. Naive Bayes classifier imposes independence

assumption on each dimension of descriptor vector  $\mathbf{x}$ , which notably simplifies the calculation of likelihood. By applying the independence assumption,

$$P(\mathbf{x} = x_1, x_2, \dots, x_K | Y = y) = \prod_{i=1}^K P(x_i = x_i | Y = y)$$

Where  $x_i$  is the  $i$ th element in vector  $\mathbf{x}$ . and

$$P(x_i = x_i | Y = y) = \frac{\sum_{j=1}^N I(X_{j,i} = x_i, Y_j = y) + a}{\sum_{j=1}^N I(Y_j = y) + a + b}$$

$I$  is an indicator function, and  $a$ ,  $b$  are pseudo-counts to avoid zero probability. Although the independence assumption is not always accurate, the estimation of likelihood, especially for high-dimensional variable  $\mathbf{x}$ , becomes feasible. Otherwise, the sample sizes required for likelihood calculation grows exponentially as the number of elements in  $\mathbf{x}$  increases. This is also called “curse of dimensionality”. Naive Bayes classifier decouples the conditional probability and the distribution of each variable dimension is approached independently. Although the independence assumption impairs the precision of posterior probability, the goal of Naive Bayes classifier is to predict the correct class or label. Correct predictions can be still reached as long as the conditional probability of the correct class is higher than the one of incorrect ones. Therefore, Naive Bayes classifier is a flexible and robust classification tool in practice.

The success of Molprint 2D descriptor, proposed by Bender et al.<sup>77</sup>, is partially due to their software package that implements Tanimoto similarity calculation and Naive Bayes classifier for ligand screening. They systematically evaluated the performance of Molprint 2D fingerprint and Naive Bayes classifier by various classes of ligands<sup>96</sup>. Naive Bayes classifier is an appropriate modeling tool in this application. The independence assumption on Molprint 2D fingerprint is approximately correct as the presence of one feature is relatively irrelevant to another. This does not mean Molprint 2D features in a ligand are absolutely independent. For example, atom B is in the environment of atom A, while atom A is also in the environment of B. Thus, the presence of a feature centered at atom A has positive contribution to the likelihood of the presence of the feature centered at atom B. The molecular weight of small organic ligands is usually limited, especially for drug-like compounds. The number of Molprint 2D features is equal to the number of heavy atoms in a chemical structure. Therefore, the present Molprint 2D features reduce the chance of observing other features. Nevertheless, these influences are only theoretical and ignorable in real-world computation because there are more important factors that we should take care of, such as the pseudo counts.

The number of possible Molprint 2D patterns is almost infinite. A regular virtual screening library may generate a dictionary that defines thousands of three-bond-length Molprint 2D features. In this case, meaningful models cannot be expected from logistic regression or linear regression because of overfitting.

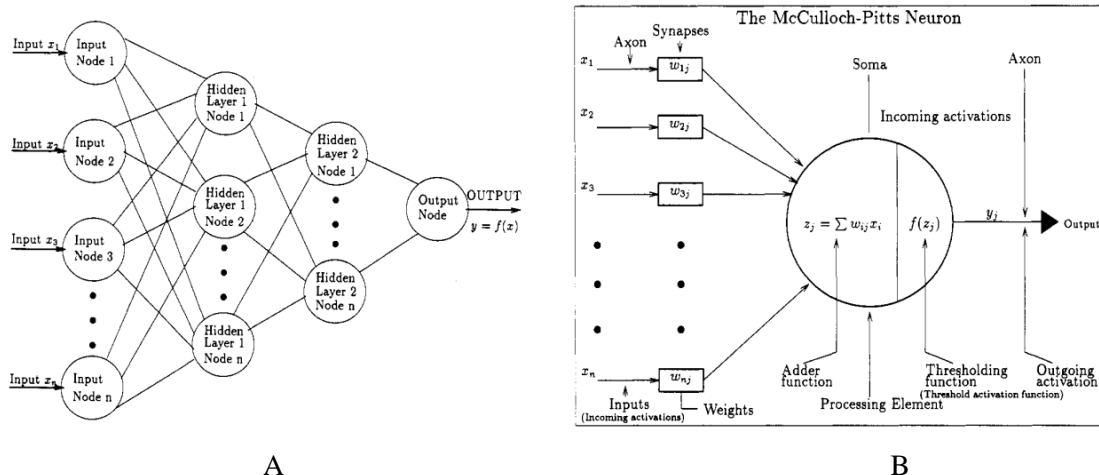
On the contrary, Naive Bayes classifier is a suitable choice as discussed above. Furthermore, it can provide degree of confidence in terms of probability. Last but not least, feature selection scheme is natively incorporated in this algorithm, resulting in transparent and interpretable probabilistic models.

### 2.3.5 Artificial Neural Network

In 1970s, research on artificial intelligence or machine learning was almost equivalent to studying artificial neural network (ANN), just like that CoMFA was almost equivalent to QSAR in 1990s. The establishment of ANN theories initialized another branch of machine learning methodology that used to be dominated by statistical modeling. Statistics means probability and inference. Distribution and probability exist in every corner of statistics. ANN, representing the opinions of computer scientists, maps explanatory variables directly to response variable. Nowadays, ANN is still a popular method in OCR, robotic science, clustering, etc. Figure 2-14 shows the overall paradigm of traditional neural network (NN) models. ANN model consists of input layer, hidden layer(s), and output layer. Each layer has a set of nodes, or neurons, to receive input, process signal and calculate output. In Figure 2-14 (A), explanatory variables  $x$  are passed to the network through the input nodes. Every node in downstream layer receives the input ( $x$ ) from all the nodes from the upstream layer. In Figure 2-14 (B), each of the input value is weighted by value  $w$ , and a linear combination of the input value becomes the incoming activation of the downstream node, i.e. the output of a hidden node  $j$  is

$$\frac{1}{1+e^{-z_j}} \text{ where } z_j = \sum_{i=1}^N w_{ij}x_i + \delta$$

Note that each hidden node possesses an independent set of weights  $w$ . The output of hidden layer nodes becomes the input of downstream nodes. The signal passes through the network structures until the output layer node is reached. The output node finally predicts the response value. It is clear that the parameters in a neural network given its structure are the weights associated with hidden nodes and output node(s). Back-propagation is a common algorithm to minimize the square loss of training data by gradually adjusting the weights.



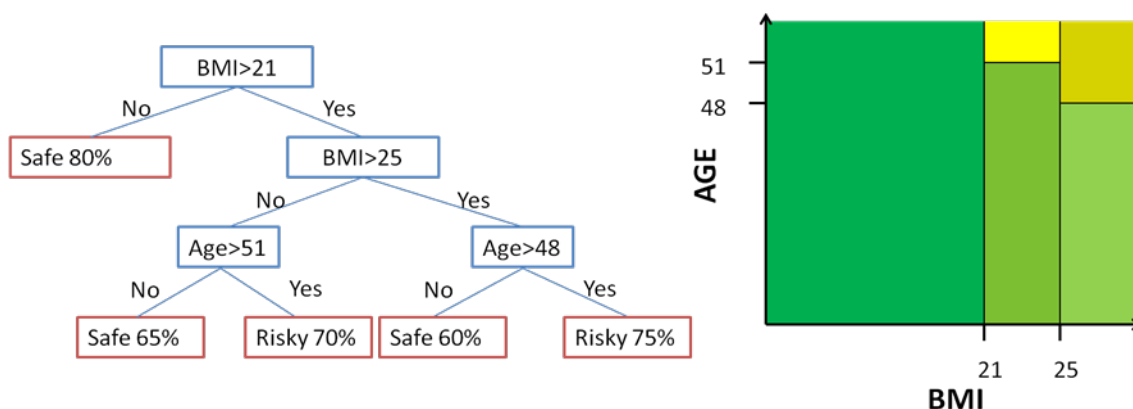
**Figure 2-14: The paradigm of artificial neural network by Sharma.**  
Figure adopted from reference<sup>97</sup>

Aoyoma et al. may be the forerunners who bring ANN in to the QSAR world<sup>98</sup>. On-line training and multi-class classification are major advantages of ANN. Its flexibility and established learning theories achieve its popularity in computer science. Nevertheless, ANN may not be the appropriate tool in drug discovery. First, the determination of network structure is time-consuming and the structure varies from application to application. The high degree-of-freedom in model complexity demands sufficient training data for reliable prediction performance.

### 2.3.6 Classification and Regression Tree

Trees are straightforward and effective data-mining technique for both classification and regression. Figure 2-15 plots a sample classification tree and corresponding 2D classification scheme. Green color indicates safety and yellow color indicates risk. Deeper color means higher prediction confidence in the region. The blue boxes indicate the splitting criteria, and the red boxes represent local classification hypotheses together with confidence level. By traversing queries through the tree, from root to leaf following the splitting rules, predictions are made according to local decision hypotheses. This sample tree utilizes two variables, BMI and age, to predict the risk of cardiovascular disease. In fact, the classification tree partitions the age-BMI space into disjoint regions, as illustrated in Figure 2-15. Separate classification hypotheses are derived for each region in order to achieve better classification

accuracy and higher confidence level. The selection of splitting criteria targets at minimizing the impurity and reducing the response variance of each region. Technical details on how to derive a tree are given in Chapter 4, LICABEDS.



**Figure 2-15: A sample classification tree for the prediction of the risk of cardiovascular disease.** The left figure displays a decision tree classifier, which partitions the AGE-BMI space in the right figure.

In spite of the simple idea, trees are shown to be quite effective in many data-mining applications. They are also widely used in different areas of computer-aided drug design, such as virtual screening<sup>99</sup>, drug-likeness prediction<sup>100</sup> and ligand property prediction<sup>101</sup>. One advantage of tree algorithm is its simplicity and flexibility. Tree algorithm has little mathematical assumptions compared to linear regression, logistic regression, and Naive Bayes classifier. Tree can be designed for both classification and regression purposes. The selection of splitting criteria effectively eliminates irrelevant features, so that we can derive a tree model from limited high-dimensional training data. Tree is a non-linear model. Its flexibility may handle complicated data patterns that present challenges to linear models.

The disadvantage of tree algorithm coexists with its advantage. First, trees are supposed to be “pruned”, otherwise they can “grow” until all the training data is perfectly fit. Tree pruning is a procedure to control overfitting. At the same time, this approach also limits the number of predictive variables in a tree structure. Second, tree structures are derived from training data in a greedy manner because the search for optimal tree structure is proven to be NP-complete problem. “Greedy” trees are vulnerable to

noise in the training data. Little variance in the training data may drastically change the tree structure, especially the root nodes.

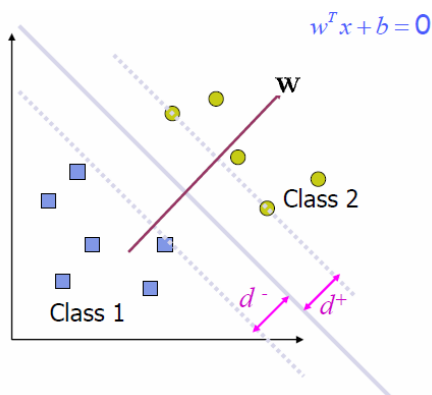
### 2.3.7 Support Vector Machine

Because of its robustness, Support Vector Machine (SVM) is probably the most successful and popular method in binary classification. SVM classifiers can be either linear or non-linear, depending on the choice of kernel function. The application domain of SVM has been further extended to regression, ranking, structured prediction, etc. Its elegant numerical optimization has guided many variant models in digital image processing, probabilistic graphical model, computational genomics, etc. The major components in SVM object function are error term and margin term. Error term corresponds to training error and margin term represents model robustness. The constraint optimization in SVM can be further transformed into quadratic programming, which guarantees the convexity and convergence to the global optimal. The outline of SVM is given in this section, and more details are described in Chapter 5, SUPPORT VECTOR MACHINE FOR LIGAND CLASSIFICATION.

In SVM, a set of training samples  $\{\mathbf{x}_i, y_i\}$  derive a decision boundary by minimizing empirical generalization error, with  $\mathbf{x}_i \in \mathbf{R}^K$  being the explanatory variables and  $y_i \in \{+1, -1\}$  being its label. A linear decision boundary or decision surface can be defined as  $\mathbf{w}^T \mathbf{x} + b$ , where  $\mathbf{w}$  is the normal to the decision surface,  $\mathbf{x}$  is the descriptor vector, and  $b$  is a bias. And the classifier can be formulated as  $f_{\mathbf{w},b}(\mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x} + b)$ . Non-linear case will be discussed in Chapter 5. Any decision surface that satisfies  $y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \forall i$ , can perfectly separate the training data. SVM assumes that the optimal decision surface has the largest distance to the nearest data points of both categories, *i.e.* margin maximization. Figure 2-16 demonstrates this concept. Skipping some technical details, the margin optimization problem can be formulated as

$$\begin{aligned} & \text{Minimize}_{\mathbf{w},b} \|\mathbf{w}\|^2 \\ & \text{Subject to } y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \forall i, i = 1, \dots, n \end{aligned}$$





**Figure 2-16: SVM, maximum margin classifier, CMU 10-701 2008 spring.**

This constraint optimization can be further transformed into the dual problem of the Lagrangian

$$\text{Maximize } L_{\alpha} = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle$$

$$\text{Subject to } \sum_{i=1}^n \alpha_i y_i = 0 \text{ with } 0 \leq \alpha_i \leq C \forall i, i = 1, \dots, n$$

Once we have  $\alpha_i$ , vector  $\mathbf{w}$  can be recovered by equation,  $\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i$ .

The introduction of support vector machine into drug discovery starts a new era. Machine-learning-based QSAR modeling becomes an alternative choice to CoMFA. SVM is reported in many cheminformatics applications, including but not limited to screening for bioactive compounds<sup>102 103</sup>, drug design for orphan receptor<sup>104</sup>, selectivity prediction<sup>105</sup>, active learning in high-throughput screening<sup>106</sup>, structure ranking<sup>107</sup> and calculation of docking score<sup>108</sup>.

The advantage of SVM is robustness and flexibility to build linear and non-linear models. SVM models generally have decent performance even if not the best. The multi-class classification is still an active research subject for SVM. Also, the choice of non-linear kernel function and parameters complicates SVM model training. The training of SVM models is an iterative procedure, so its time complexity is significantly higher than the statistical methods introduced earlier.

### **2.3.8 Ensemble Methods**

“Weak learners” may hardly capture the versatile distribution patterns in real life. Combining multiple models to attain better performance is the motivation of ensemble learning methods. An important chapter of this thesis, LICABEDS, describes an ensemble learning algorithm for ligand classification. Theory shows that any better-than-random-guess learners can be combined to build a strong learner. Besides support vector machine, ensemble methods become another interesting research topic in machine learning. Adaptive boosting and bootstrap aggregating are representative ensemble methods.

Adaboost, short for adaptive boosting, systematically develops an ensemble model by assigning weights to weak learners. Thorough discussion on this method can be found in Chapter 4. Bagging, another name for bootstrap aggregating, trains each weak learner using a randomly drawn subset of the whole training set with replacement. Weak learners in bagging ensemble model vote equally for the outcome. Famous “random forest” belongs to this category.

The choice of weak learner is influential in ensemble methods. Because of lengthy computation time, some fast algorithms are preferred as weak learners, such as decision trees, linear discriminant analysis. Some studies show that ensemble methods exhibit better performance than SVM, ANN, Naive Bayes classifier. The ensemble nature allows the estimation of variance and confidence band, but its optimization stems from robust assumptions. As emerging techniques, their development and applications will catch researchers’ attention.

### **2.3.9 Miscellaneous**

Some successful machine learning and statistical models for QSAR/QSPR are summarized above. The advance of machine learning in a decade builds far more algorithms than these, like K Nearest Neighbor, Fisher Linear Discriminant Analysis, Genetic Algorithm, to name a few. No algorithm is always superior to another in all aspects. Depending on the applications, hypothesis, complexity, and available training data, choosing the appropriate technique is the foundation of successful modeling. Examples in Figure 2-3 and Figure 2-4 are provided to demonstrate this point.

An overview of QSAR/QSPR theory is covered so far. Biologists, chemists, and pharmacologists who are not familiar with statistics and computer science need some automated tools for QSAR/QSPR modeling. Tripos Sybyl, Molecular Operating Environment, Discovery Studio, and Schrödinger are well-known drug discovery platforms, implementing the whole pipe line of QSAR/QSPR modeling. Table

2-11 lists some commercial QSAR software products. These commercial products are highly automated, including management of compound library, descriptor generation, model training, validation, prospective prediction, and visualization. This is definitely an advantage for most users, but the flexibility and transparency are their disadvantage, even if these platforms provide scripting language to customize computation. As a scientific platform, Accelrys' Pipeline Pilot is another choice for QSAR/QSPR modeling. In Pipeline Pilot, users design the whole experiment by connecting functional modules and controlling data flow. Nevertheless, this approach heavily relies on the availability of published modules.

Researchers who have sufficient IT techniques may customize descriptor generation and statistical modeling. All of major drug discovery platforms are capable of generating and exporting a large collection of molecular descriptors. For example, Unity fingerprint in Sybyl, ECFP in Discovery Studio, pharmacophore fingerprints in MOE. Software development kits, such as OpenBabel, ChemAxon, OpenEye, CDK, can be also used to generate molecular descriptors.

**Table 2-11: Machine learning algorithms in major drug discovery platforms**

<i>Software Platform</i>	<i>Algorithms</i>	<i>Product Name</i>
Tripes Sybyl	PLSR	CoMFA, HQSAR
Molecular Operating Environment (MOE)	Tree, Linear Regression, PCA, PLSR	QuaSAR
Discovery Studio	ANN, Bayesian model, PLSR, Linear Regression	QSAR and Library Design
Schrödinger	PLSR, Linear Regression, PCA	Strike Field-based QSAR

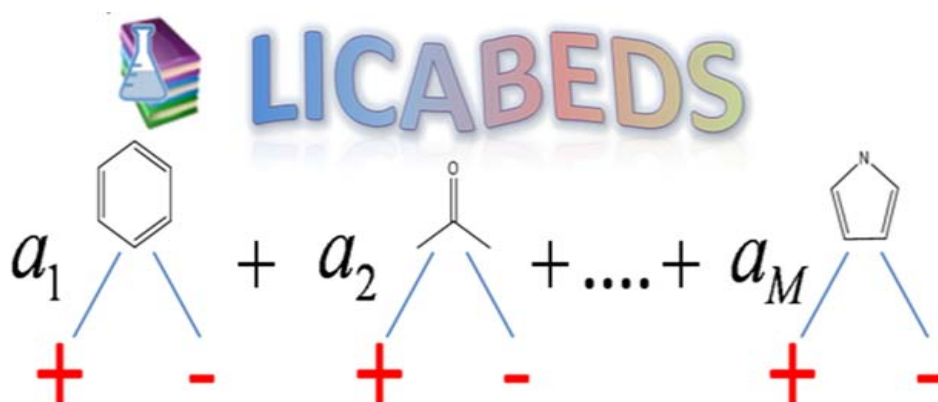
As to machine learning and statistics software, SPSS is a famous interactive platform implementing routine algorithms. WEKA implements many machine learning algorithms with graphical interface support. R, Matlab, S-PLUS and SAS are powerful statistical analysis software that delivers scripting languages. Most of the machine-learning algorithms are implemented as package importable to these platforms, such as LibSVM. Sometimes, modeler may seek implementations from third party, for example, the preference ranking algorithm in SVMlight package (<http://svmlight.joachims.org/>).

### **3 AIMS OF THE STUDY**

The aims of the study cover the development and implementation of cutting-edge machine learning and statistical modeling algorithms for handling large-scale cheminformatics data in order to guide drug discovery programs and boost productivity. The more specific aims of this dissertation include:

1. Developing general-purpose linear and non-linear machine learning classifiers for the prediction of a variety of molecular properties that influence drug-likeness of candidate compounds (LiCABEDS and Support Vector Machine);
2. Evaluating their performance on diverse pharmacological properties and validating the predictions in prospective screening (ligand functionality, blood-brain-barrier passage and ligand selectivity);
3. Designing high-throughput parallel computing strategy to accelerate existing data mining algorithms in order to extend their application to terabyte compound database (GPU computing);
4. Proposing and justifying compound acquisition approach to construct structurally diverse screening libraries that could potentially enhance hit rate and enrichment factor.

## 4 LICABEDS



LICABEDS is the acronym of Ligand Classifier of Adaptively Boosting Ensemble Decision Stumps. It is an instance of adaptive boosting, and it belongs to ensemble methods in supervised learning. This chapter is focused on the description of LiCABEDS algorithm, its implementation, and applications. First, mathematical concepts of LiCABEDS are provided, which is followed by a case study on modeling 5-HT<sub>1A</sub> ligand functionality. LiCABEDS is implemented as user friendly software that integrates data import/export, modeling training, prediction, etc. The software specification is given in the second section. Then, another case study, the prediction of CB1/CB2 ligand selectivity using LiCABEDS, demonstrates its effectiveness in quantitative structure-property relationship. This chapter ends with brief results for predicting ligand blood-brain-barrier passage.

## 4.1 LICABEDS AND MODELING LIGAND FUNCTIONALITY

Advanced high-throughput screening (HTS) technologies generate great amount of bioactivity data, and this data needs to be analyzed and interpreted with attention to understand how these small molecules affect biological systems. As such, there is an increasing demand to develop and adapt cheminformatics algorithms and tools in order to predict molecular and pharmacological properties based on these large datasets. In this section, a novel machine-learning-based ligand classification algorithm, named Ligand Classifier of Adaptively Boosting Ensemble Decision Stumps (LiCABEDS), is reported for data-mining and modeling of large chemical datasets to predict pharmacological properties in an efficient and accurate manner. The performance of LiCABEDS was evaluated through predicting GPCR ligand functionality (agonist or antagonist) using four different molecular fingerprints, including MACCS, FP2, Unity and Molprint 2D fingerprints. Studies showed that LiCABEDS outperformed two other popular techniques - classification tree and Naive Bayes classifier - on all four types of molecular fingerprints. Parameters in LiCABEDS, including the number of boosting iterations, initialization condition, and a “reject option” boundary, were thoroughly explored and discussed to demonstrate the capability of handling imbalanced datasets, as well as its robustness and flexibility. In addition, the detailed mathematical concepts and theory are also given to address the principle behind statistical prediction models. The LiCABEDS algorithm has been implemented into a user-friendly software package that is accessible online at <http://www.cbligand.org/LiCABEDS/>.

### 4.1.1 Introduction

As a complement to modern high-throughput screening, one of the primary goals of virtual screening and cheminformatics techniques is to explore the enormous chemical and biological properties in a time-efficient manner as well as to help reduce the cost of experimental screening<sup>9-11</sup>. In particular, great emphasis is placed on the “drugability” or “drug-likeness” of compounds using cheminformatics tools in the early stages of drug development, with the hope of increasing the probability of “lead” compounds, or their derivatives, to pass through the later phases of drug clinical trials<sup>12</sup>.

Despite numerous molecular properties and various mathematical models, the prediction of ligand binding activity and biological properties can be addressed by two types of approaches: a classification model for categorical response and a regression model for continuous response. For example, some pharmaceutical properties, such as mutagenicity, can be modeled by ligand classification<sup>109</sup>. To build up a

quantitative structure-property relationship (QSPR) model, pattern recognition methodology can be applied to map molecular descriptors to continuous value or categorical value via regression or classification<sup>19</sup>. These molecular descriptors are usually binary or continuous vectors describing various aspects of molecular attributes or structural patterns. Many ligand properties pertaining to drug discovery have been successfully modeled with hundreds of molecular descriptors or fingerprints through statistical or machine learning techniques<sup>110-112</sup>. As one of the representative regression models, Comparative Molecular Field Analysis (CoMFA)<sup>13</sup> applies partial least square regression to make predictions from the principal components that are linear combinations of electrostatic and steric energy fields at 3D grids. CoMFA was successfully applied in the prediction of membrane flux<sup>113</sup>, modeling structure-pharmacokinetic relationships<sup>114</sup> and antagonist binding affinities at cannabinoid receptor subtype<sup>115</sup>. A CoMFA model was also developed to distinguish 5-HT<sub>1A</sub> agonists and antagonists<sup>116</sup>, which is also one of the focuses in this section. Another classification technique, Naive Bayes classifier, has also been used to model quantitative structure-selectivity relationships<sup>117</sup>.

Despite the advance of cheminformatics methodology, it remains a challenge to develop a robust, reliable, and interpretable ligand classifier to tackle different scenarios in computer-aided drug design. Although any regression method like CoMFA can be adapted as a ligand classifier, such an approach often suffers from overfitting due to the model complexity of the regression method. In addition, the ability to find a 3D bioactive conformer remains as one of the limits<sup>14</sup>. Many existing modeling methods may require researchers to perform variable selection. In practice, variable selection is still a complicated procedure that ultimately has a large effect on the final predictive model. Free parameters are manually specified in most computational models, for example, the number of components in the CoMFA method. Besides, cross-validation is often carried out to find optimal values of those parameters, but this practice could be computationally inefficient, and its performance also heavily relies on the choice of cross-validation datasets.

Thus, reliable and robust ligand classifiers are needed to aid scientists and researchers in discovering compounds with desired properties in both the lead discovery as well as the drug development process. The adaptive boosting algorithm, or Adaboost, introduced by Freund and Schapire<sup>118</sup>, is a general method used to produce a “strong” classifier by combining a series of “weak learners”. Sharing certain resemblance with the support vector machine (SVM) algorithm, Adaboost is also a maximum-margin classifier and tends not to overfit the training data<sup>119</sup>. Advantageously, the number of boosting rounds is the only essential parameter in Adaboost training, which simplifies the computational process of machine learning algorithms. In spite of the advantages, this algorithm has rarely been applied and discussed in drug discovery. In this study, a novel ligand classifier, LiCABEDS, was proposed and implemented by adaptively boosting sets of decision stumps based on 2D molecular fingerprints. In the established

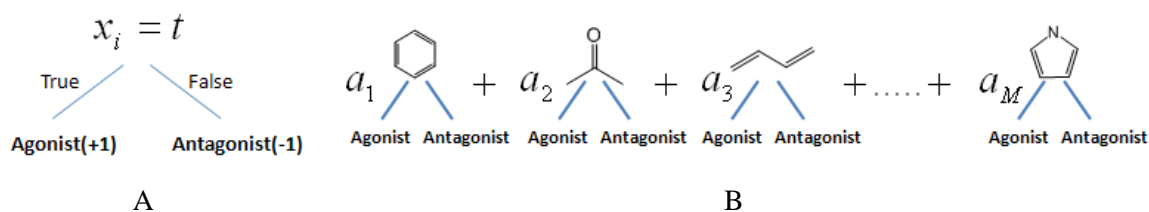
algorithm, important features are automatically selected and weighted accordingly to build “weak learners” in model training. The performance and the characteristics of our novel algorithm are demonstrated and tested through the application on modeling ligand functionality for serotonin receptors or 5-hydroxytryptamine (5HT) receptors, belonging to an important family of G protein-coupled receptors (GPCRs). In addition, across-target studies indicate the potential application of LiCABEDS on orphan receptors. This section also describes the detailed mathematical concepts of the LiCABEDS algorithm. It is anticipated that LiCABEDS, as a general-purpose ligand classifier, can be applied to model more biochemical and pharmacological properties. The model development is free of conformation search and is readily automated with the robustness of 2D molecular fingerprints. Its performance and application are described below. Finally, the algorithm is implemented in a freely available and user-friendly software package, allowing the easy importing of datasets and model development. The fully functioning software package is available online to the scientific community.

#### **4.1.2 Methods, Materials and Calculations**

The detailed mathematics concepts of Ligand Classifier of Adaptively Boosting Ensemble Decision Stumps (LiCABEDS) and its application on modeling ligand functionality are described below. As case studies, LiCABEDS was first used to model the ligand functionality for the 5HT-subtype GPCR families by predicting a given ligand to be either an agonist or an antagonist. For a parallel study, the performance of LiCABEDS was compared to two other popular data-mining methods: classification tree<sup>120</sup> and Naive Bayes classifier<sup>121</sup>. The underlying theory of these two methods is also introduced in this section.

**4.1.2.1 Ligand Classifier of Adaptively Boosting Ensemble Decision Stumps** Adaptive boosting, initially introduced by Yoav Freund and Robert Schapire<sup>118</sup>, is a general machine learning technique to create a strong classifier by combining a series of “weak learners” for improving the accuracy of prediction. In LiCABEDS, “decision stumps” are designed to be the weak learners. As illustrated in Figure 4-1, the “decision stump” denotes a heuristic classification hypothesis that a compound will be classified as an agonist (+1) if the  $i$ th bit of fingerprint ( $x_i$ ) is equal to a target value ( $t$ ); or as an antagonist (-1), otherwise.





**Figure 4-1: Graphical illustration of LiCABEDS**

(A) Illustration of a “decision stump based on molecular fingerprint. (B) Illustration of the composition of LiCABEDS.

Instead of using the graphic representation, a “decision stump” can be formulated by a function:  $y(x, i, t) = 2I(x_i = t) - 1$ , where  $I$  is an indicator function,  $I(Z) = 1$  if the statement  $Z$  is true;  $I(Z) = 0$ , otherwise.  $x$  is the molecular fingerprint vector,  $i$  is the index of the fingerprint, and  $t$  is the target value. For example, if  $x_i$ , the  $i^{\text{th}}$  bit of fingerprint, is equal to the target value  $t$ , then  $I(x_i = t) = 1$  and  $y(x, i, t) = 1$  (agonist). If  $x_i$ , the  $i^{\text{th}}$  bit of fingerprint is different from  $t$ , then  $I(x_i = t) = 0$  and  $y(x, i, t) = -1$  (antagonist).

Different from many other machine-learning algorithms, LiCABEDS, as an ensemble method, is designed to achieve stronger classification power by boosting many “weak” classification hypotheses. As illustrated in Figure 4-1, a series of “decision stumps” with corresponding weights  $a_m$  vote for the final prediction, which can be formulated as the weighted summation of the outcome of every “decision stump”:

$$Y_M = \text{sign}\left(\sum_m^M a_m y_m(x, i_m, t_m)\right) \quad (4-1)$$

$\text{sign}(z) = 1$  if  $z > 0$ , or  $-1$  otherwise. The unknown variables,  $a_m$ ,  $i_m$  and  $t_m$ , for each weak classifier  $m$  can be “learned” from training datasets using the following algorithm:

1. Initialize the sample weights for each training compound  $n$ ,  $w_n = 1/N$ ,  $n = 1, \dots, N$ ,  $N$  is the total number of training compounds.

2. For each round of calculation  $m = 1, \dots, M$

Find  $i_m, t_m$  for weak learner  $y_m$  by minimizing the weighted error function

$$(i_m, t_m) = \arg \min_{i_m, t_m} \sum_{n=1}^N w_n I(y_m(X_n, i_m, t_m) \neq l_n) \quad (4-2)$$

where argmin is the function to return the arguments which minimize the object function,  $X_n$  is the descriptor vector for compound  $n$ ;  $l_n = \pm 1$  is the label of compound  $n$ .  $i_m, t_m$  uniquely define a “decision stump”, and their optimal values can be found by enumerating all possible combinations of  $i_m, t_m$ .

Evaluate the quantities:

$$\varepsilon = \frac{\sum_{n=1}^N w_n I(y_m(X_n, i_m, t_m) \neq l_n)}{\sum_{n=1}^N w_n}; a_m = \ln \frac{1 - \varepsilon}{\varepsilon} \quad (4-3)$$

$a_m$  becomes the weight for the “decision stump”  $m$ . Then update the weights of training compounds for next round of calculation:

$$w_n \leftarrow w_n \exp(a_m I(y_m(X_n, i_m, t_m) \neq l_n)) \quad (4-4)$$

The number of training steps,  $M$ , is the only parameter that must be specified manually in the algorithm. Cross-validation is one of the options to specify the optimal value of  $M$ ,  $M_{optimal}$ . Training error is steadily minimized as  $M$  increases. While the training algorithm aims to minimize the exponential loss function, boosting algorithm may have potential to overfit the training data as pointed by others<sup>122</sup>. Despite such potential, Freund and Schapire have shown the underlying mechanism that adaptive boosting does not

often suffer from overfitting<sup>119</sup>. Discussion is given later on the difference between a large value of  $M$  (by default) and  $M_{optimal}$  in order to address the overfitting issue.

Training compound datasets may potentially be overwhelmed by one category of training samples. In this case, the majority class is usually favored in the prediction. To minimize the effect of disproportionate training samples in each category, balanced class weight can be set as an alternative initialization condition to equal initial weight. In other words, the total weights for each class are equal at the initialization step:  $\sum_{n=1}^N w_n I(l_n = 1) = \sum_{n=1}^N w_n I(l_n = -1)$ . For example, all the labeled agonists in the training set may have initial weights  $1/N_{+1}$ , where  $N_{+1}$  is the total number of agonists in the training data. Similarly, all of the antagonists may have an initial weight  $1/N_{-1}$ .

Heuristically, the absolute value of  $A = \sum_m^M a_m y_m(x, i_m, t_m)$  indicates the degree of confidence in the prediction, because a relatively large population of “decision stumps” vote for the corresponding class. On the other hand, a low absolute value of  $A$  indicates uncertainty in the prediction. Better prediction accuracy is anticipated by avoiding uncertain cases, which we also refer to as “reject option”. In our study, a prediction is only made for a test compound if  $|A| > c$ , where  $c$  is rejection threshold. Otherwise, an “unknown” label is assigned to the test compound.

**4.1.2.2 Classification Tree** Classification tree is a straightforward and effective data-mining technique. It has been widely applied to different areas of computer-aided drug design, such as virtual screening<sup>99</sup>, drug-likeness prediction<sup>123</sup> and ligand blood-brain-barrier passage<sup>101</sup>.

A classification tree consists of a set of split criteria and leaf nodes. The split criteria control the region that a ligand belongs to, while the leaf nodes represent classification hypotheses that are derived from training datasets in the same regions. The structure of a decision tree can be induced from training datasets in a greedy manner. By recursively partitioning the entire training dataset into regions, impurity  $impurity(t)$  is minimized regarding each possible partitioning  $t$ :

$$\min_t impurity(t) = \sum_s \lambda_s(t),$$

where  $s$  is the new region created from split  $t$ , and  $\lambda_s(t) = 1 - \sum_{j=0}^1 \hat{p}_s(j)^2$ . In this study, splitting rule is chosen from  $x_i = 0$  or  $x_i = 1$ , where  $x_i$  is the  $i^{th}$  bit of descriptor vector.  $\hat{p}_s(j)$  is the maximum likelihood

estimator of a ligand being  $j$ ,  $j = 0$  or  $1$  ( $0$  represents antagonists, and  $1$  represents agonists), in region  $S$ . Training data can be perfectly fitted by growing the tree until 100% purity is achieved at each node. To avoid overfitting, k-fold cross-validation is commonly employed to control the “height” of a decision tree. After “pruning” the whole tree according to the cross-validation score that is defined as the percentage of correct predictions on cross-validation sets in this study, the optimal tree structure will be used to make predictions for novel ligands.

**4.1.2.3 Naive Bayes Classifier** The Naive Bayes classifier method is a simple classification method based on applying Bayes' theorem with independence assumptions<sup>121</sup>. The method relies on the assumption that the presence or absence of a particular feature or class is unrelated to the presence or absence of any other feature. This independence assumption, with regard to molecular fingerprints simplifies the estimation of the likelihood function, which makes the method applicable to many computer-aided drug design tasks, such as virtual screening<sup>124</sup> and selectivity prediction<sup>117</sup>. In this study, Naive Bayes classifier was used to model the probability of one ligand being an agonist or an antagonist, given its molecular fingerprint:  $\Pr(Cl | Fp)$  where  $Fp$  is the molecular fingerprint vector, and  $Cl = 1$  for agonist or 0 for antagonist. By applying Bayesian theory,  $\Pr(Cl | Fp) \propto \Pr(Fp | Cl) \Pr(Cl)$ , the predicted class of a given ligand is antagonist,  $\hat{Cl} = 0$ , if  $\Pr(Cl = 0 | Fp) \geq 0.5$ ; and  $\hat{Cl} = 1$ , otherwise.  $\Pr(Fp | Cl)$  can be approximated by applying the independence assumption to molecular fingerprints:

$$\Pr(Fp | Cl) = \prod_i \Pr(Fp_i | Cl) \text{ where } Fp_i \text{ is the } i^{\text{th}} \text{ bit of fingerprint.}$$

Due to the difference between Molprint 2D and other types of fingerprints, the equation used in calculating the likelihood was also different. For example, we had Molprint 2D string “2;0-1-0; 2;0-2-2;” and MACCS fingerprint “0101” for a testing compound (only for illustration). The likelihood of the Molprint 2D fingerprint was calculated as:

$$\Pr(Fp = 2;0-1-0; 2;0-2-2; | Cl) = \Pr(Fp_1 = 2;0-1-0; | Cl) \times \Pr(Fp_2 = 2;0-2-2; | Cl)$$

The likelihood of MACCS key was calculated as:

$$\Pr(Fp = 0101 | Cl) = \Pr(Fp_1 = 0 | Cl) \times \Pr(Fp_2 = 1 | Cl) \times \Pr(Fp_3 = 0 | Cl) \times \Pr(Fp_4 = 1 | Cl)$$

The presence or absence of predefined MACCS features are considered in the likelihood calculation, while only present Molprint 2D features are modeled in the calculation.

#### 4.1.2.4 Dataset Preparation, Molecular Fingerprint and Computation Protocol

To evaluate the performance of LiCABEDS, all the labeled human 5-HT<sub>1A</sub>, 5-HT<sub>1B</sub>, 5-HT<sub>1D</sub>, and 5-HT<sub>4R</sub> agonists and antagonists were retrieved from the GLIDA database<sup>125-126</sup>. The ligand quantity and their properties are summarized in Table 4-1 (properties were calculated using the Sybyl8.0 [www.tripos.com](http://www.tripos.com)).

**Table 4-1: Molecular properties of agonists and antagonists**

Receptor	Ligand Category	Quantity	Molecular Weight	No. of Rotatable Bond	No. of H-Bond Acceptor	No. of H-Bond Donor
HUMAN	Agonist	1102	346.7 ± 72.6	5.4 ± 2.4	2.8 ± 1.5	1.3 ± 0.9
5-HT <sub>1A</sub>	Antagonist	595	385.8 ± 70.7	5.0 ± 2.5	3.2 ± 1.3	1.7 ± 0.9
HUMAN	Agonist	104	348.4 ± 104.1	4.3 ± 2.5	3.6 ± 1.8	1.2 ± 0.8
5-HT <sub>1B</sub>	Antagonist	38	359.6 ± 88.4	4.1 ± 1.8	2.6 ± 1.4	1.2 ± 0.7
HUMAN	Agonist	685	339.1 ± 82.3	5.8 ± 2.6	3.3 ± 1.6	1.2 ± 0.7
5-HT <sub>1D</sub>	Antagonist	335	420.2 ± 71.0	4.1 ± 2.2	3.9 ± 1.4	1.7 ± 0.9
HUMAN	Agonist	287	369.4 ± 65	3.9 ± 2.5	2.8 ± 1.1	1.1 ± 0.5
5-HT <sub>4R</sub>	Antagonist	262	367.3 ± 61.5	5.7 ± 2.1	3.6 ± 1.4	1.1 ± 0.4

\*Molecular properties are given by sample average and standard deviation.

With the published compound datasets, the prediction accuracy of different data-mining methods along with different molecular descriptors was assessed on the labeled agonists and antagonists of the human 5-HT<sub>1A</sub> subtype G-Protein Coupled Receptor (GPCR) by ten rounds of calculation. For each round of calculation, three classification methods were compared, including LiCABEDS, classification tree, and Naive Bayes classifier. Each was trained on the same randomly selected training compounds. The set of training compounds was composed of 75% labeled agonists and antagonists (827 5-HT<sub>1A</sub> agonists and 446 5-HT<sub>1A</sub> antagonists). The remaining 25% ligands (275 5-HT<sub>1A</sub> agonists and 149 5-HT<sub>1A</sub> antagonists) were used as a testing dataset in order to evaluate the prediction accuracy of different methods. The prediction accuracy was estimated by comparing the predictions to the real ligand labels (agonists or antagonists). With Molprint 2D<sup>77, 124</sup> as descriptor, the across-target ligand functionality prediction was also made by LiCABEDS. In this case, a LiCABEDS model was trained on all the labeled human 5-HT<sub>1A</sub> ligands (totally 1697 ligands), and then predictions were made on the ligands for human 5-HT<sub>1B</sub>, 5-HT<sub>1D</sub>, and 5-HT<sub>4R</sub> receptors. (Dr. Lirong Wang collected structure and bioactivity data from GLIDA database)

In this study, four types of molecular fingerprints were generated for each compound, including MACCS key<sup>13</sup>, Unity ([www.tripos.com](http://www.tripos.com)), FP2<sup>127</sup>, and Molprint 2D<sup>77, 124</sup> fingerprint. (Unity and FP2

fingerprints were generated by Dr. Lirong Wang). The MACCS key fingerprint was calculated by Chemistry Development Kit (CDK)<sup>128-129</sup> and the Unity fingerprint was calculated by Sybyl 8.0 (www.tripos.com). Openbabel<sup>127</sup> was used to generate the FP2 fingerprint, and Molprint 2D package was used to generate the Molprint 2D fingerprint. Variable selection was not carried out before model trainings, so all the dimensions of these molecular fingerprints were exposed to the learning algorithms. To ensure a fair amount of overlapping in Molprint 2D patterns, a three-layer atom environment was used for predicting ligand functionality for the human 5-HT<sub>1A</sub> receptor, while a two-layer atom environment was preferred for across-target predictions. Each feature defined by Molprint 2D was mapped to a unique bit in the descriptor vector by an in-house program.

The implementation of the classification tree presented in this study came from a Tree package in “R”<sup>120, 130</sup>. To avoid overfitting, a tree node was not split unless more than ten training compounds were observed in the parent node and more than five were present in both child nodes. Lastly, each classification tree was “pruned” according to ten-fold cross-validation scores. The implementation of Naive Bayes classifier on the Molprint 2D fingerprint was from the Molprint 2D software package. An in-house Naive Bayes classifier was also developed for other fingerprints. The implementation of LiCABEDS is discussed in the following section (LiCABEDS Software Package).

LiCABEDS models were initially developed using a large value of  $M$  ( $M = 10000$ ), for all fingerprint types to ensure a convergence of training error. Furthermore, the influence of  $M$  was studied, and optimal values of  $M$  were determined by running cross-validation with 10% of training compounds as a cross-validation set. LiCABEDS models were developed using balanced weights when compared to Classification Tree and Naive Bayes classifier on human 5-HT<sub>1A</sub> ligand datasets. Next, equal weights were compared to balanced weights as initialization conditions, which was conducted on the same training and testing datasets. Finally, the prediction accuracy was assessed with the “reject option” boundary ranging from 0 to 3.

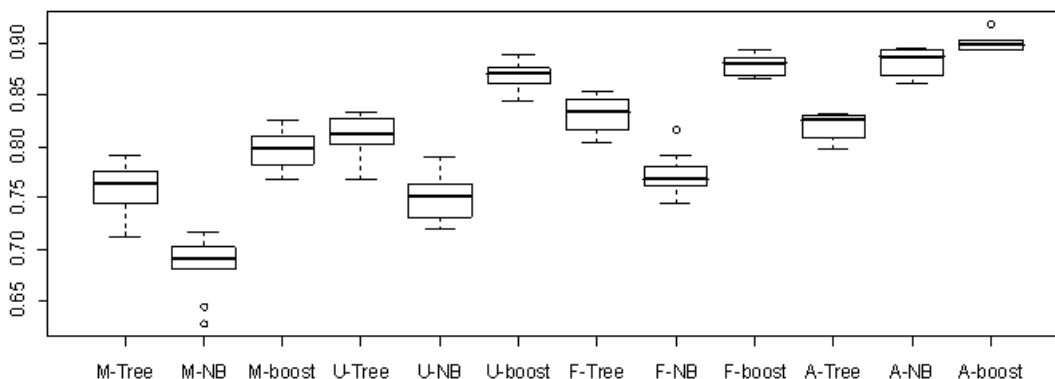
**4.1.2.5 LiCABEDS Software Package** A user-friendly interface was developed for LiCABEDS in order to simplify the steps involved in project management, model training and making predictions. The software integrates automated importing of training and testing datasets. The training module features automatic cross-validation, flexible initialization and interruptible model development. The “Reject option” is also implemented for making predictions. The graphical user interface allows for flexible model editing, prediction browsing, and result exporting. In addition, a work session can be saved to local hard disk, so that the previous workspace can be restored by the program. The program has been tested on Intel i7 860 2.8GHz CPU to evaluate the computational time. To iterate 10000 steps on 1697

compounds, model training takes 44 minutes for Molprint 2D fingerprint, 5 minutes for FP2 or Unity fingerprints, and 1 minute for MACCS fingerprint. The calculation time for model training is related to the amount of training samples, the number of boosting iterations and the dimension of descriptors. Once the model is established, the predictions can be made instantly by the program.

### 4.1.3 Results and Discussion

According to the distribution of physical properties listed in Table 4-1, simple classification hypotheses do not distinguish agonists from antagonists very well, if even at all. On the other hand, molecular fingerprints encode a large amount of chemical information regarding structural patterns of small molecules. The strength of the LiCABEDS method lies in its ability to robustly process this fingerprint information and make better predictions on the small molecules. I will discuss the performance of different fingerprints and computational models. The effect of different parameters in LiCABEDS is also discussed in detail.

#### 4.1.3.1 Accuracy of LiCABEDS, Tree, and Naive Bayes Classifier



**Figure 4-2: The distribution of prediction accuracy from ten rounds of calculation**  
X-axis is organized according to descriptor and prediction model. Y-axis is simply prediction accuracy. M: MACCS key fingerprint; U: Unity fingerprint; F: FP2 fingerprint; A: Molprint 2D fingerprint. Tree: Classification tree; NB: Naive Bayes classifier; boost: LiCABEDS.

Even if the predictions are made in the same descriptor space, the derived decision boundaries of different machine learning algorithms rarely agree, because of the discrepancy of underlying model assumptions, as well as object optimization functions. The results of systematic comparisons among classification tree, Naive Bayes classifier and LiCABEDS are plotted in Figure 4-2 (M: MACCS key; U: Unity fingerprint; F: FP2 fingerprint; A: Molprint2D fingerprint. Tree stands for classification tree, NB stands for Naive Bayes classifier and boost is short for LiCABEDS.). A summary of the results can also be found in Table 4-2, which reports the distribution of prediction accuracy out of ten rounds of calculation on human 5-HT<sub>1A</sub> ligands.

**Table 4-2: Sample mean and standard deviation of prediction accuracy**

Fingerprint	Model	Tree	Naive Bayes	LiCABEDS
	MACCS	0.759 ± 0.026	0.685 ± 0.027	0.799 ± 0.016
	Unity	0.810 ± 0.023	0.753 ± 0.023	0.869 ± 0.013
	FP2	0.831 ± 0.018	0.771 ± 0.021	0.879 ± 0.010
	Molprint2D	0.820 ± 0.013	0.883 ± 0.013	0.901 ± 0.008

The table lists the sample average and standard deviation of prediction accuracy out of 10 rounds of calculation.

As shown in Figure 4-2 and Table 4-2, LiCABEDS uniformly outperforms both classification tree and Naive Bayes classifier regardless of the choice of molecular fingerprints. First, LiCABEDS exhibits the highest average prediction accuracy on ten different testing compound datasets. When Unity and FP2 fingerprints are used as the descriptor, the highest number of mistakes made by LiCABEDS on testing sets is still lower than the lowest of the Tree or Naive Bayes classifier methods. With the Molprint 2D fingerprint as descriptor, the lowest accuracy from LiCABEDS is 0.894, which is almost the same as the highest accuracy from Naive Bayes classifier, 0.896. Not only does LiCABEDS show the highest average prediction accuracy, but also it possesses the lowest standard deviation on four kinds of fingerprints. This indicates model stability as well as model reliability. This is further seen in the standard deviation of prediction accuracy from LiCABEDS. As listed in Table 4-2, the standard deviation is 0.008 on the Molprint 2D fingerprint, while the standard deviation of both the Classification Tree and Naive Bayes classifier methods is 0.013. A similar pattern is also observed using the other three types of molecular fingerprints. The standard deviation from LiCABEDS ranges from 0.010 to 0.016 using the FP2, Unity and MACCS fingerprints. On the other hand, the standard deviation of both the Tree and Naive Bayes



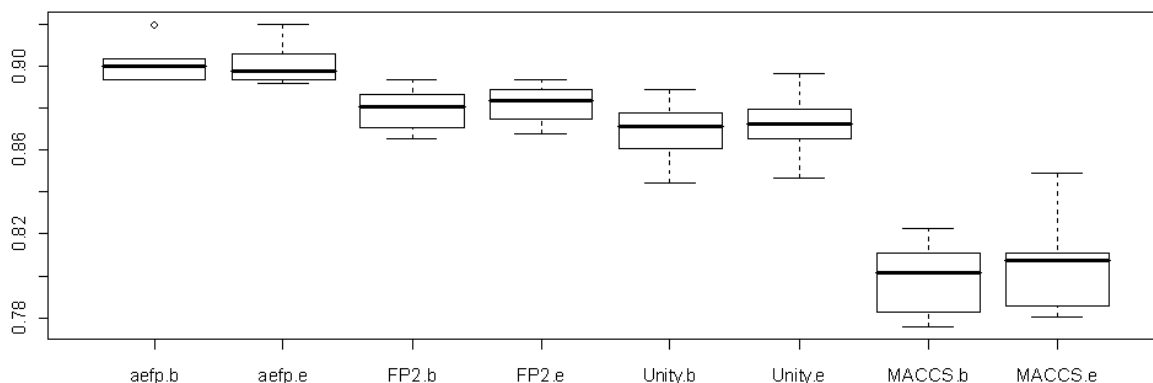
methods range from 0.018 to 0.027 using these three fingerprints. Therefore, LiCABEDS is less affected by the distribution of training compounds compared to the Tree and Naive Bayes methods.

Molprint 2D was the most predictive descriptor among the four types of fingerprints. In this study, a total of 6839 features were defined in the whole human 5-HT<sub>1A</sub> dataset. The length of the Unity and FP2 fingerprints were 992 and 1024, respectively. The MACCS key had the shortest bit length of 168. Although Molprint 2D encoded many structural patterns in comparison to the Unity or FP2 fingerprints, it did not significantly improve the performance of the classification tree. As the “height” of tree was limited after “tree pruning” to avoid overfitting, a limited number of features could be considered in the classification hypothesis. The LiCABEDS method, on the other hand, consisted of 10000 weighted “decision stumps” and many factors contributed to the final prediction. This might explain the reason why LiCABEDS yielded more accurate and reliable predictions than the classification tree method.

The Naive Bayes method outmatched the Tree method using Molprint 2D as a descriptor, but the Tree method outmatched the Naive Bayes method with other fingerprints. As previously mentioned in the method section, the Naive Bayes models were slightly different with different fingerprints. Molprint 2D, as an atom environment descriptor, only considered the features present, while FP2, MACCS and Unity predefined a set of substructures and modeled both the presence and absence of structural patterns. When the Naive Bayes classifier from Molprint 2D was applied to the other three fingerprints, the test calculation showed that the result was even worse. The Naive Bayes classifier treated each dimension in the fingerprint equally. Thus, the performance could be affected by noise from irrelevant features. The independence assumption in the Naive Bayes model was not necessarily true for molecular fingerprints, which was one of the factors impairing the estimation of the likelihood function. In addition, the training algorithm of LiCABEDS selected the most predictive “decision stump” and assigned its weight accordingly in order to build the classifier systematically. In that sense, not all the dimensions of the fingerprint vectors contributed to the prediction equally, and predictive features and corresponding “decision stumps” were emphasized with relatively large weights,  $a_m$ . Without much assumption regarding the fingerprints, LiCABEDS built robust models and produced more accurate predictions than the Naive Bayes classifier.

**4.1.3.2 Initialization Condition** As previously mentioned in the Methods section, the equal initial weight in the training algorithm considers each training compound equally, while the balanced initial weight considers two compound categories (agonist and antagonist) equally. The two different initializations in LiCABEDS were compared on the same ten sets of training and testing compounds with  $M = 10000$ . Figure 4-3 shows the distribution of the overall accuracy of predictions from combinations of

different initialization conditions and molecular fingerprints. aefp stands for Molprint2D, b stands for balanced initial weight and e stands for equal initial weight. Table 4-3 lists the average accuracy and standard deviation for each ligand category, as well as for the whole testing dataset.



**Figure 4-3: Prediction accuracy with different initialization conditions**

“aefp” represents Molprint 2D fingerprint. When letter “b” follows a fingerprint, it means the model is trained with balanced initialization condition. Otherwise, equal weight is specified as initialization condition

**Table 4-3: Initialization condition for LiCABEDS training**

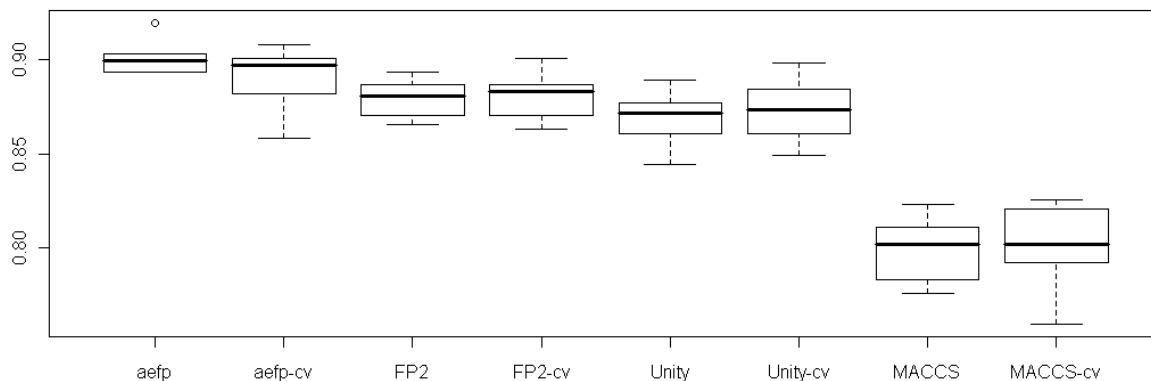
		Agonist	Antagonist	Overall
Molprint 2D	Balanced weight	0.913 ± 0.018	0.879 ± 0.041	0.901 ± 0.008
	Equal weight	0.924 ± 0.018	0.856 ± 0.038	0.900 ± 0.009
FP2	Balanced weight	0.888 ± 0.021	0.864 ± 0.026	0.879 ± 0.010
	Equal weight	0.901 ± 0.021	0.848 ± 0.029	0.882 ± 0.009
Unity	Balanced weight	0.891 ± 0.018	0.830 ± 0.023	0.869 ± 0.013
	Equal weight	0.901 ± 0.020	0.816 ± 0.022	0.871 ± 0.014
MACCS	Balanced weight	0.803 ± 0.025	0.792 ± 0.033	0.799 ± 0.016
	Equal weight	0.860 ± 0.018	0.702 ± 0.049	0.805 ± 0.020

The sample mean and standard deviation of prediction accuracy for each category of ligands, using equal initial weight and balanced initial weight.

According to Figure 4-3 and Table 4-3, these two initialization conditions result in differences with respect to the overall performance, even if equal initial weight is slightly better. As displayed in Table 4-3, the balanced initial weight correctly predicts antagonists at a percentage of 87.9%, 86.4%, 83.0% and 79.2% with Molprint 2D, FP2, Unity and MACCS descriptors, while the equal initial weight predicts

antagonists at the accuracy of 85.6%, 84.8%, 81.6% and 70.2% on the same descriptors. The opposite pattern is observed for agonist prediction, in which the equal initial weight uniformly outperforms the balanced initial weight. To explain this, LiCABEDS training algorithm with equal initial weight aims to minimize the error function by making fewer mistakes on the training datasets, while balanced weight emphasizes both ligand categories and each training sample. For example, at the initial step of training algorithm with balanced weights, the cost to make a mistake is  $1/N_{-1}$  for one antagonist and  $1/N_{+1}$  for one agonist. As there are 827 agonists and 446 antagonists in the training datasets, LiCABEDS may tend to avoid making mistakes on antagonists because one mistake on an antagonist costs more than the one on an agonist ( $1/N_{-1} > 1/N_{+1}$ ). On the other hand, the equal initial weight favors the majority category, because the mistake on any training compound costs  $1/N$ . Although the weights for each training sample are updated in the follow-up training iteration, the initialization condition still significantly affects the model development. As a result, the balanced initial weight makes the predictions that are more accurate on antagonists. In reality, training sets are sometimes overwhelmed by one category of samples, but correct predictions are still desired for the minority group. The balanced initial weight seeks a tradeoff between the accuracy for each category and the overall performance, which makes the algorithm generally applicable to many data mining situations.

**4.1.3.3 Training Parameter** Besides the initialization condition, another parameter crucial to the LiCABEDS training algorithm is  $M$ , the number of boosting iterations. To minimize overfitting, the optimal value of  $M$ ,  $M_{\text{optimal}}$  ( $M_{\text{optimal}} < 10000$ ) can be determined through cross-validation. In this process, a fraction of the training compounds (10% of the whole training sets) was left out as a cross-validation set.  $M_{\text{optimal}}$  was then compared to the default condition, a large value of  $M$ ,  $M = 10000$ , on the same training and testing datasets. Models were developed using a balanced initial weight and the four types of fingerprints. The percentages of correct predictions on the ten testing datasets are shown in Figure 4-4. The x-axis denotes the choice of fingerprint and  $M$ . aefp stands for Molprint2D fingerprint. cv means that number of boosting rounds is set to  $M_{\text{optimal}}$ . The label without cv means  $M=10000$  by default.



**Figure 4-4: The boxplot showing the effect of number of boosting rounds**

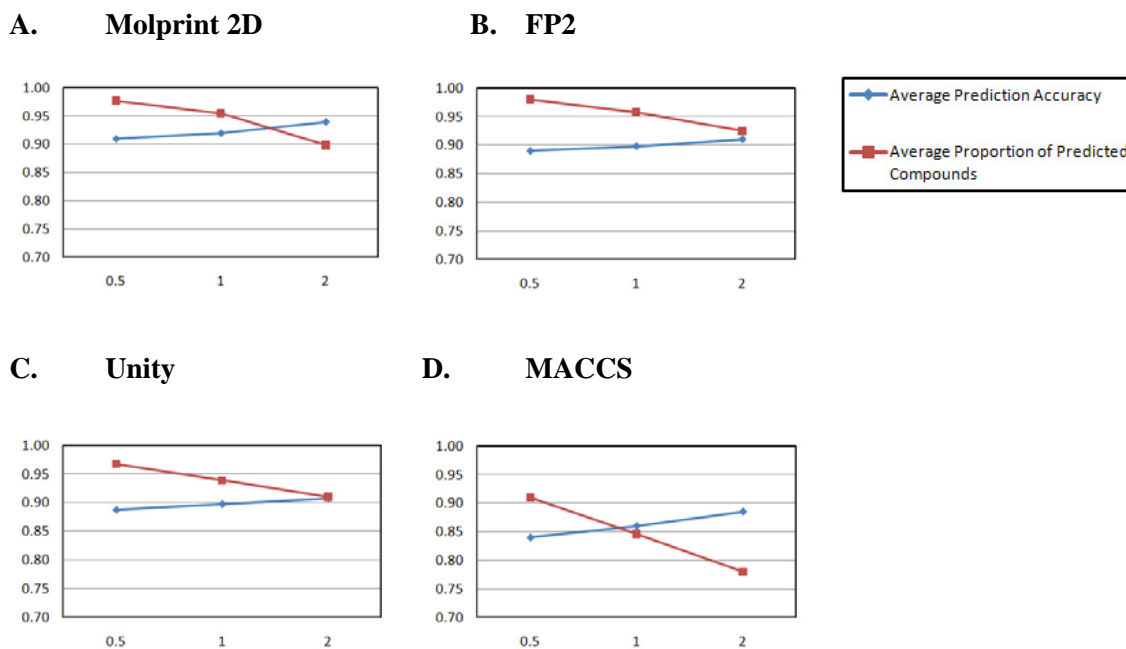
According to the distribution shown in Figure 4-4, models developed by  $M_{\text{optimal}}$  iterations are moderately better than  $M = 10000$  on FP2, Unity and MACCS fingerprints, but not as good as  $M = 10000$  on Molprint 2D fingerprint. Because  $M = 10000$  is much larger than the length of FP2, Unity and MACCS fingerprints, some dimensions in the fingerprint are overrepresented in the classifier. This may result in overfitting. If this is the case, running cross-validation could control the overfitting and improve the performance. Nevertheless, the length of the Molprint 2D fingerprint used in this study is 6839, which is at the magnitude of  $M = 10000$ . Thus, cross-validation is not essential for the Molprint 2D fingerprint.

Hypothesis testing was carried out to quantify the difference between  $M_{\text{optimal}}$  and  $M = 10000$ . The distribution of correct predictions on the testing datasets was examined. The null hypothesis was “the models trained with and without cross-validation have the same performance”, while the alternative hypothesis was “cross-validation improves the model performance”. Student’s t-test showed that the one-sided p-values for the four types of fingerprints (Molprint 2D, FP2, Unity and MACCS) were 0.954, 0.357, 0.290 and 0.354, respectively. This result does not significantly favor the alternative hypothesis, indicating cross-validation does not significantly influence on the prediction generated. The data mining studies and the results presented also indirectly support the conclusion that LiCABEDS is not so susceptible to overfitting in the studied datasets, which is also supported by boosting theory<sup>119</sup>. Although cross-validation is not strictly required by LiCABEDS, parameter tuning may still be beneficial under certain circumstances, such as the application of LiCABEDS on FP2, Unity and MACCS fingerprints.

**4.1.3.4 Reject Option** To assess the confidence-rated predictions, LiCABEDS uses the raw value of

$A = \sum_m^M a_m y_m(x, i_m, t_m)$  to address the degree of belief for each prediction. By applying the concept of

“reject option”, accurate prediction is anticipated, provided a high absolute value of  $A$ , whereas an “unknown” label is output to prevent uncertain prediction for a low absolute value of  $A$ . To validate this hypothesis, predictions were made on the ten testing compound datasets with different “reject” boundaries and molecular descriptors. The difference in the average performance of different “reject option” boundaries is reported in Figure 4-5. Each figure corresponds to a type of fingerprint. X-axis denotes the value of decision boundary. When Molprint 2D was used as the descriptor, an average accuracy of 90.1% (Table 4-2) was reported without using the “reject option”. As shown in Figure 4-5A, the average proportion of predictions made on the testing datasets readily decreases when the “reject” boundary increases from 0.5 to 2. Because more “unknown” labels are output when a higher boundary value is specified, a relatively smaller fraction of predictions is made. In the meantime, the average prediction accuracy increases from the original value of 90.1% (reported in Table 4-2) to 91.1%, 92.1%, and 93.8% with corresponding boundaries being 0.5, 1 and 2, respectively. A similar trend can be observed with the other three types of fingerprints as well. For example, using the FP2 fingerprint, the predictions are made on 92.3% of the testing compounds when the boundary is set to 2. An accuracy of 91.1% is obtained from the “reject option”, which is a noticeable improvement from the original accuracy of 87.9% (reported in Table 4-2). Therefore, LiCABEDS is not only able to attain better performance by making selective predictions, but is also able to estimate the classification risk for testing compounds through the absolute value of  $A$ , or the confidence-rated prediction. In practice, it is sometimes economical to sacrifice some testing compounds to achieve accurate predictions. The boundary value can be determined by examining the distribution of  $A$  and leaving a fair prediction ratio.



**Figure 4-5: “Reject option” with LiCABEDS**

The plot shows average prediction accuracy and percentage of prediction by changing the prediction boundary. X-axis represents a value above which a prediction is made.

**4.1.3.5 Across-target Ligand Functionality Prediction** In addition to the ligand functionality classification, I have explored the potential of across-target ligand bioactivity prediction using LiCABEDS program. With the assumption that agonists and antagonists might share some common pharmacological features for similar receptor subtypes, the LiCABEDS model, which was developed from human 5-HT<sub>1A</sub> ligands on Molprint 2D fingerprint, was used to predict the ligand functionality for other human 5-HT subtype receptors, including 5-HT<sub>1B</sub>, 5-HT<sub>1D</sub>, and 5-HT<sub>4R</sub> receptors. 5-HT<sub>1A</sub>, 5-HT<sub>1B</sub> and 5-HT<sub>1D</sub> GPCRs can be classified as serotonin receptor subtype 1 (or 5-HT<sub>1</sub>) while 5-HT<sub>4R</sub> belongs to the family of serotonin subtype four. 5-HT<sub>1B</sub> receptor has the shortest evolution distance to 5-HT<sub>1A</sub>. On the other hand, 5-HT<sub>4R</sub> receptor has the largest evolution distance to 5-HT<sub>1A</sub> of all. As sufficient number of known agonists and antagonists has been reported for these receptors, the correlation between model predictability and target similarity can be studied in order to understand the scope of application of established models.

**Table 4-4: Across-target ligand functionality prediction**

Receptor	Accuracy for agonists	Accuracy for antagonists	Overall accuracy	Blastp similarity score <sup>a</sup>	Sybyl similarity score <sup>b</sup>
Human 5-HT <sub>1B</sub>	0.875	0.816	0.859	304	397.80
Human 5-HT <sub>1D</sub>	0.812	0.609	0.745	279	393.60
Human 5-HT <sub>4R</sub>	0.826	0.267	0.559	142	340.90

<sup>a</sup> Blastp similarity score is calculated by blastp using default scoring parameters

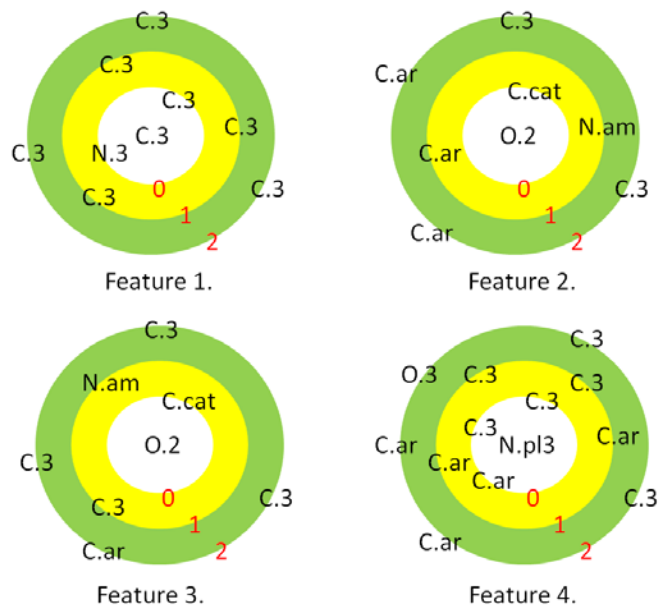
<sup>b</sup> Sybyl similarity score is calculated by Sybyl Biopolymer<sup>131</sup> using “pmutation” scoring matrix.

The performance of LiCABEDS models for each category of the 5-HT ligands, as well as entire datasets, can be seen in Table 4-4. The sequence similarity scores compared to human 5-HT<sub>1A</sub> are calculated by blastp<sup>132</sup> and Sybyl Biopolymer<sup>131</sup>, respectively. The calculations based on the 5-HT<sub>1A</sub> model show that 85.9% of predictions are correct on 5-HT<sub>1B</sub> ligands, with 87.5% accuracy for agonists and 81.6% accuracy for antagonists, respectively. The data is congruent with the relative high sequence similarity between 5-HT<sub>1B</sub> and 5-HT<sub>1A</sub> (blastp score: 304; Sybyl score: 397.80). The studies show that the model trained from 5-HT<sub>1A</sub> ligands is still predictive for 5-HT<sub>1B</sub> ligands, but the overall accuracy for 5-HT<sub>1D</sub> ligands drops to 74.5%, which may be attributed to the lower sequence similarity between 5-HT<sub>1D</sub> and 5-HT<sub>1A</sub> (blastp score: 279; Sybyl score: 393.60). 5-HT<sub>4R</sub>, which possesses the lowest sequence similarity to 5-HT<sub>1A</sub> (blastp score: 142; Sybyl score: 340.90), was also evaluated, and its prediction is not necessarily better than a random guess. The results are consistent with the known data that the drugs Ergotamine (agonist) and Methiothepin (antagonist) are active to 5-HT<sub>1A</sub>, 5-HT<sub>1B</sub> and 5-HT<sub>1D</sub> receptors but not to 5-HT<sub>4R</sub>. The results may also suggest that LiCABEDS prediction models may have potential of applying to other targets with limited known ligands, as long as the models are developed for a closely related receptor family. This concept could extend the application of LiCABEDS to the drug discovery process targeting at orphan receptors that has no known ligand reported.

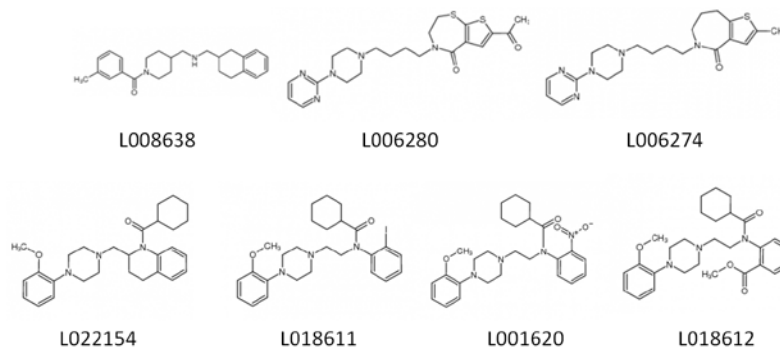
**4.1.3.6 Model Interpretation** The interpretability of the LiCABEDS model may help us understand the underlying classification mechanism and significant features regarding ligand properties. The model developed on the first set of 5-HT<sub>1A</sub> training compounds, which consist of randomly selected 827 agonists and 446 antagonists, is used to demonstrate this process. As presented in the method section, each “decision stump” contributes to the final prediction according to its weight,  $a_m$ , as described in equation (4-1). In the LiCABEDS model, four out of 6839 highly weighted Molprint 2D features, which are the

few highly weighted ones to distinguish agonists and antagonists, are listed in Table 4-5. Feature 1 and 2 are favored by agonists, while feature 3 and 4 are preferred in antagonists. In order to illustrate the structural patterns of these features, Figure 4-6 shows a graphical atom environment according to the four features. Each feature depicts a central atom and its atom environment up to a specific topological distance. The atom environment in Molprint2D is defined as the quantity of heavy atoms surrounding the central atom. Heavy atoms are distinguished by Sybyl atom types. For example, feature 1 (0;0-1-0;0-1-4;1-3-0;2-3-0;) translates to a substructure of a central sp<sup>3</sup> carbon atom (C.3) neighbored by one sp<sup>3</sup> carbon atom and one sp<sup>3</sup> nitrogen atom (N.3), and surrounded by three sp<sup>3</sup> carbon atoms (C.3) located two or three bonds away. The Molprint2D features in Table 4-5 are generated by Molprint2D software package, and the detailed explanation can be found in original publication<sup>77, 124</sup>.





**Figure 4-6: Four sample Molprint 2D features in graphic representation**



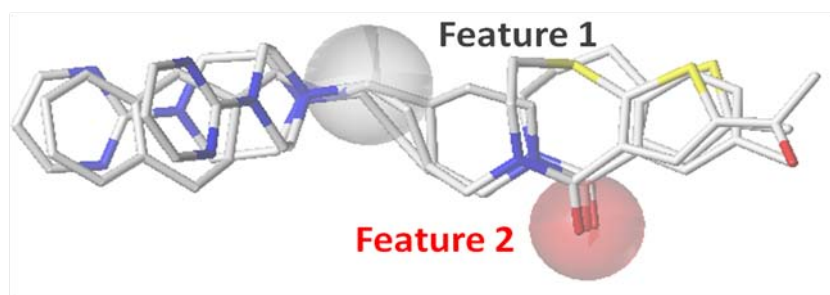
**Figure 4-7: Compounds used to exemplify four features**

**Table 4-5: List of important Molprint features regarding ligand functionality**

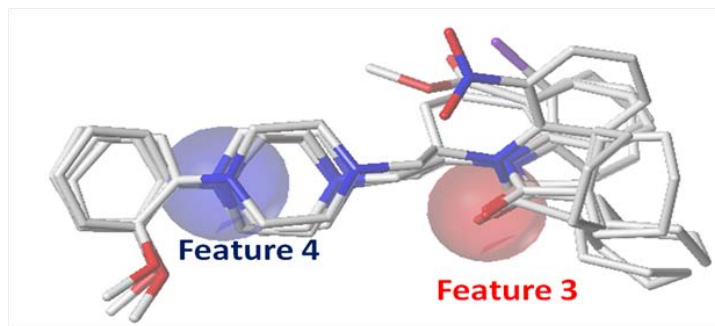
	Bit Index	Molprint 2D feature <sup>a</sup>	Weight
Feature 1	3474	0;0-1-0;0-1-4;1-3-0;2-3-0;	0.869
Feature 2	808	8;0-1-1;1-1-2;1-1-27;2-2-0;2-2-2;	0.328
Feature 3	362	8;0-1-1;1-1-0;1-1-27;2-3-0;2-1-2;	0.343
Feature 4	4322	18;0-2-0;0-1-2;1-2-0;1-2-2;2-2- 2;2-2-4;2-1-7;	0.639

<sup>a</sup> Features listed in this table are extracted from Molprint 2D software package. The interpretation of the listed features is presented graphically in Figure 4-6.

Figure 4-7 lists seven compounds selected from the testing compound dataset in order to exemplify the four features. The first three compounds (L008638, L006280, L006274) are labeled as agonists, and the other four compounds (L022154, L018611, L001620, L018612) are labeled as antagonists. It is worth pointing out that thousands of features are involved agonist/antagonist prediction, but only four highly weighted Molprint 2D features are picked up to illustrate model interpretation. Molprint2D fingerprint (feature) is a highly sparse descriptor, and the number of features that a compound possesses is equal to the number of its heavy atoms. The agonists from the testing set, which possess both feature 1 and 2 (listed in Table 4-5), are involved in Model Interpretation. The same analogy is applied to antagonists. Even if the agonists or antagonists do not share the same structural scaffold, certain substructures may still match in three-dimensional space. Figure 4-8 displays that feature 1, 2 from the three agonists are well aligned, with the central carbon atom from feature 1 labeled in grey and the central oxygen atom from feature 2 labeled in red. Similarly, Figure 4-9 displays the alignment of feature 3, 4 for the four antagonists. The result suggests that those features might be related to ligand functionality and ligand-protein interaction. The interpretability of LiCABEDS models is rooted in the explanation of each “decision stump”, especially the highly weighted ones. Therefore, LiCABEDS models can be easily understood and interpreted, which could potentially guide chemical modification to achieve better pharmacological or physicochemical profile.

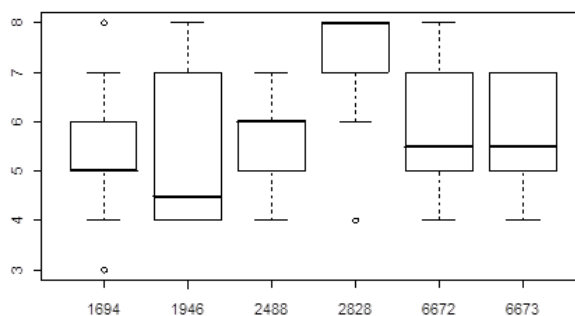


**Figure 4-8: 3-D alignment of three agonists**



**Figure 4-9: 3-D alignment of four antagonists**

**4.1.3.7 Model Robustness** Model robustness is the potential to handle diverse training data and provide consistent predictions. Section 3.1 has shown that LiCABEDS models render the most consistent and accurate predictions on any of the molecular fingerprints. To analyze the composition of the classifiers, important dimensions of Molprint 2D fingerprints are extracted from the LiCABEDS models, which are developed on ten different 5-HT<sub>1A</sub> training datasets. All the Molprint 2D features are observed totally more than 50 times in all the models and possess weights,  $a_m$ , larger than 0.08. To visualize the major components of the classifiers, Figure 4-10 shows the distribution of occurrence of six important Molprint 2D features that are favored in agonists, for which LiCABEDS training algorithm may select a feature several times to minimize generalization error. The occurrence of features mainly ranges from 4 to 7, except dimension 2828. Even if the ten models are developed on the randomly selected training datasets, only three outliers (labeled as circles in Figure 4-10, two in dimension 1694, one in dimension 2828) are identified in the boxplot. Thus, the occurrence of the six major components has moderate variance in each of the ten models. The stability of the influential “weak-learners” leads to consistent prediction accuracy, although only a few features are visualized.



**Figure 4-10: The occurrence of the six major LiCABEDS components in ten 5-HT<sub>1A</sub> models**

#### 4.1.4 Conclusion

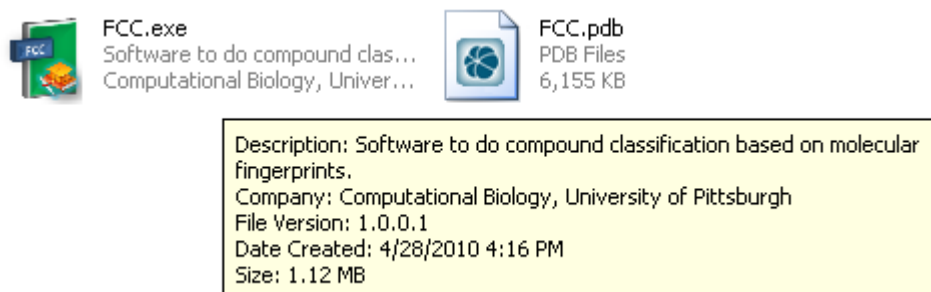
This chapter reports a novel ligand classification algorithm, Ligand Classifier of Adaptively Boosting Ensemble Decision Stumps (LiCABEDS), and thoroughly investigated it through the case studies of ligand functionality prediction for the GPCR 5-HT subtypes. The performance of LiCABEDS is compared to the Classification Tree model and Naive Bayes classifier using four types of molecular fingerprints: Molprint 2D, FP2, Unity and MACCS. The results show that LiCABEDS uniformly produces the most accurate and consistent predictions, especially with Molprint 2D fingerprints as the descriptor. Additionally, unique characteristics of LiCABEDS make it applicable to model various ligand properties. The flexible initialization conditions of LiCABEDS allow the development of predictive models and emphasize minority categories on unbalanced training datasets. Parameterization is usually a complicated procedure in many machine-learning algorithms, however, model development in LiCABEDS is simplified because the number of boosting iterations,  $M$ , is the only parameter required for model training. The result from cross-validation suggests that a large value of  $M$  still yields satisfactory performance, which makes the model training process simplified in practice. Another valuable characteristic of LiCABEDS is the “reject option”, which returns the degree of confidence for each prediction. Higher prediction accuracy can be achieved by rejecting some “low-confident” testing samples. The capability of LiCABEDS is further demonstrated through the application on a cross-target prediction. The interpretation of LiCABEDS models may reveal the correlation between structural pattern and molecular properties of interest. The robustness of LiCABEDS models is further demonstrated by examining the principal components of “decision stumps”. Lastly, LiCABEDS has been implemented into an easy-to-use and freely available (<http://www.CBLigand.org/LiCABEDS/>) software platform that provides a graphical user interface for automating model development and predictions. As a general classifier, LiCABEDS may also have great potential for modeling and predicting other ligand properties, such as ADME prediction and other applications on in-silico drug design research. These are ongoing projects and will be reported in future studies.

## 4.2 SOFTWARE MANUAL FOR LICABEDS

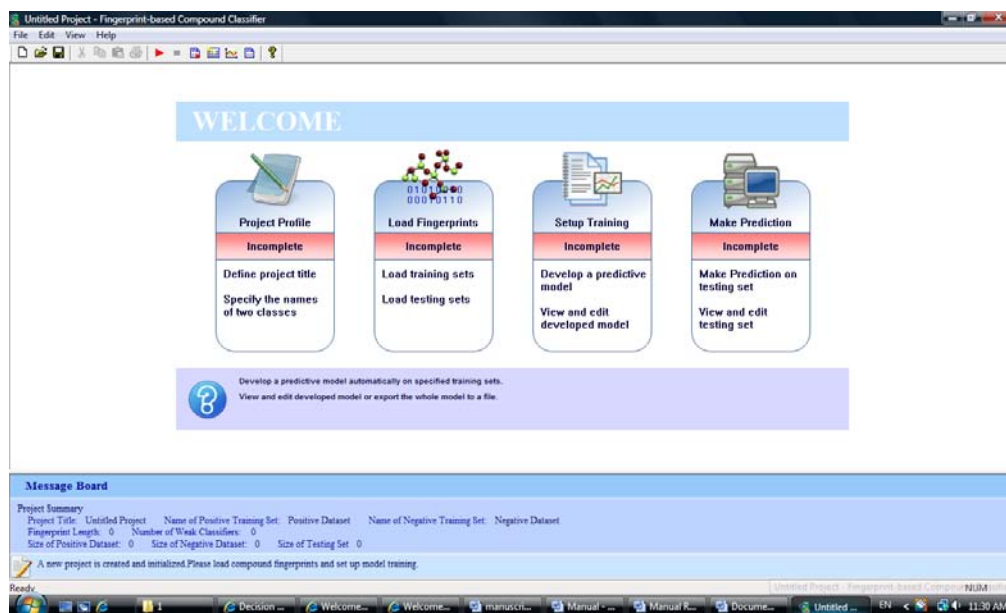
LiCABEDS is implemented as highly automated software to simply the whole modeling process. Computer program, Fingerprint-based Compound Classifier (FCC), integrates data import/export, project

management, model training, and model testing into an integrated pipeline. Currently, LiCABEDS is the major machine learning algorithm in this platform. This section introduces the functions of FCC program and provides a walk-through project. The program and sample tutorial files are available at [www.cbligand.org/LiCABEDS](http://www.cbligand.org/LiCABEDS).

FCC is developed as “green” software for Microsoft Windows system, which means that software installation is not strictly required. To launch the program, locate the executable file and double click the “FCC.exe” icon.



The interface of the program is displayed in the figure below. Similar to most Windows programs, FCC has a title bar, showing the title of current project and “minimize, maximize, close” buttons. A menu bar and a tool bar are displayed below the title bar. The tool bar provides shortcuts to some frequently used functions, such as new project, copy-and-paste function, model training, etc. Besides toolbar, all the program functions are listed in the menu. A main window is designed for information exchanging. In the screenshot, the main window displays a welcome page, project status and brief introduction to each module. Underneath the welcome page is a message board, displaying system message and project parameters.




In FCC, a ligand classification project mainly consists of four steps:

- Define a project in “Project Profile”: name the project title and specify ligand categories of interest.
- Load Fingerprints: import training and testing compound datasets in predefined fingerprint format.
- Setup Training: develop a LiCABEDS model.
- Make Predictions: use the developed model to make predictions on testing compound dataset.

The homepage also shows the progress of current project by putting different colors on each tag. A module tag will turn green after the corresponding step is completed. A red tag indicates that this step remains unexplored, whereas a yellow tag indicates this step is partially finished.

The toolbar lists frequently accessed functions:

File functions: 

- New project: current project profile and data are restored to default settings. All the changes made to the project are discarded. A warning message will be displayed to remind saving your work if any modification is detected.
- Open project: a previous saved workspace will be restored. All the changes made to the current project are discarded.
- Save project: a work session will be saved to a FCC file on hard drive. The project will resume next time by opening the FCC workspace file.

Editing Functions: 

- Cut: delete selected items and deposit the content into system clipboard. The button turns grey if the action is not applicable.
- Copy: Deposit the selected items into system clipboard. The button turns grey if the action is not applicable.
- If a compatible format is available in the clipboard, the contents will be pasted to the current project. This function only works in prediction window.

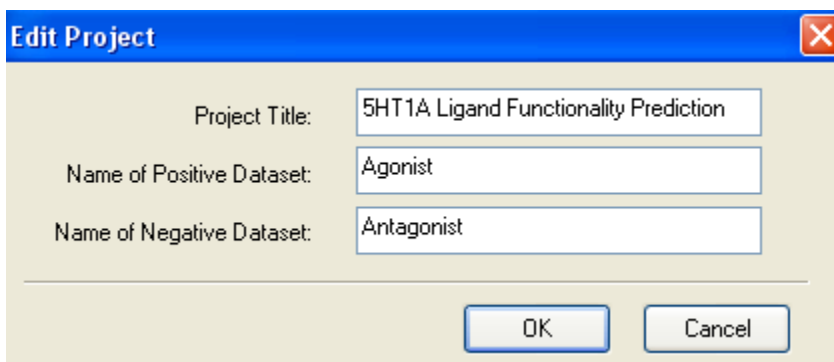
Project Functions: 


- Start a new model training process or continue with previous model training
- Stop undergoing model training.

- Make predictions using developed predictive model
- Go to home page
- Go to training window for model browsing
- Go to prediction window to view prediction results.

### Step One: Define a Project

A new project starts with defining its aim and two compound categories. Click “Project Profile” tag at home page or go to “Edit Project Title” in “Edit” menu to define a project. A dialog box will pop up and its interface is displayed below:



In “Project Title”, input a name to distinguish the current project from others. The title name will become the default file name to save the workspace (to be discussed later). The project title is also displayed in the “Message Board”, so users can easily identify the aim and content in this project. Next, specify two classification categories. For convenience, the categories could be the properties to be modeled, e.g. agonist and antagonist in the screenshot. The program will automatically output the predicted category for each testing compound after predictions are made. Click “OK” button to finish project definition. Users can always repeat the same procedure to change the settings. User may start a new project by clicking  button on the toolbar or go to menu “File->New Project...”.


### Step Two: Load Fingerprints

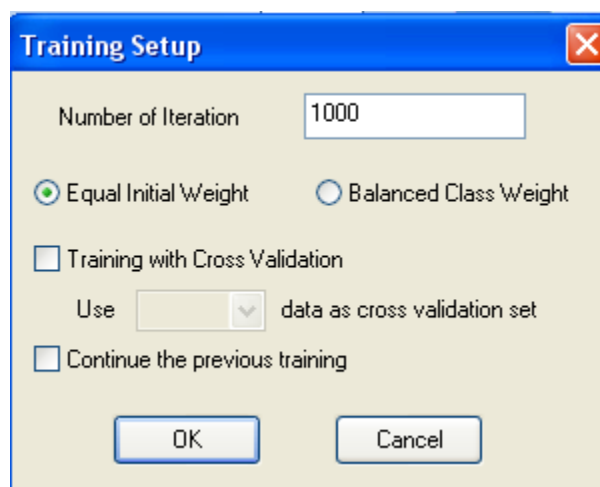
The imbedded classification algorithm in FCC is called Ligand Classifier of Adaptively Boosting Ensemble Decision Stumps (LiCABEDS). The theory and performance of LiCABEDS have been discussed previously. Like many other supervised learning algorithms, LiCABEDS also relies on training data to derive predictive models.





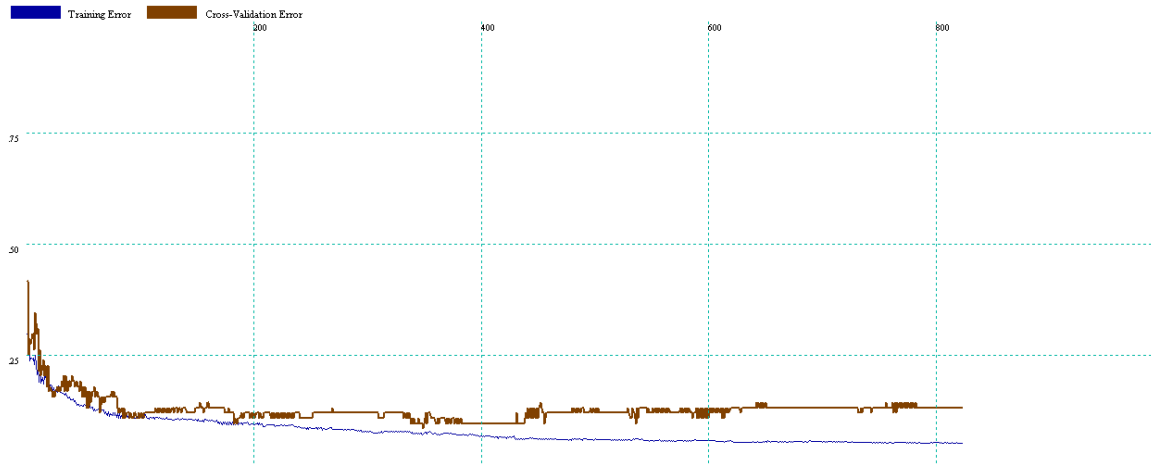
Once both categories of training compound datasets are imported, automatic datamining can be carried out to derive a predictive LiCABEDS model. This procedure is called model training or model development. In model training, FCC will examine the distinct patterns between two categories of compounds and build classifiers accordingly.


To setup training, click  on the toolbar, or go to “welcome page” and click “Setup Training”. Then, a dialog will pop up to collect a few parameters:



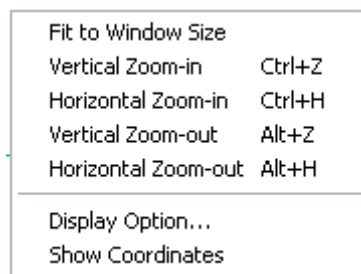
The most important parameter in LiCABEDS training is the number of iteration, which is equal to  $M$  mentioned previously. Users have the option to carry out cross-validation to search for its optimal value. In this case, check “Training with Cross Validation” and select the percentage of training datasets to be used as cross-validation set. Users can also choose initialization condition: equal initial weight or balanced class weight. If equal initial weight is chosen, then all the training compounds are treated equally. Nevertheless, this initialization condition may not be suitable for many ligand classification tasks, e.g. virtual screening. In virtual screening, thousands of compounds are labeled as inactive but only a few are reported as active. The program can simply treat all the compounds as inactive to achieve nearly 100% accuracy. To solve this problem, balanced class weight assigns equal weight to each category instead of treating each training sample equally. Balanced class weight is encouraged if unbalanced training datasets are provided. Interruptible training is another competitive feature of LiCABEDS. Users can always continue with the previous training by checking “Continue the previous training” checkbox. In this case, training error will be minimized based on the established model by adding more weighted decision stumps. For example, a model is developed using  $M = 1000$ , but the model performance is not satisfactory on the cross-validation set or testing dataset. Users can simply continue the previous training by setting  $M = 1000$  again. Then, the final model will contain totally 2000 decision stumps. This feature allows reusing the developed model to generate a new classifier instead of start-over from the beginning.

After clicking “OK” button, model training will start automatically and a real-time error curve will be displayed in the main window.

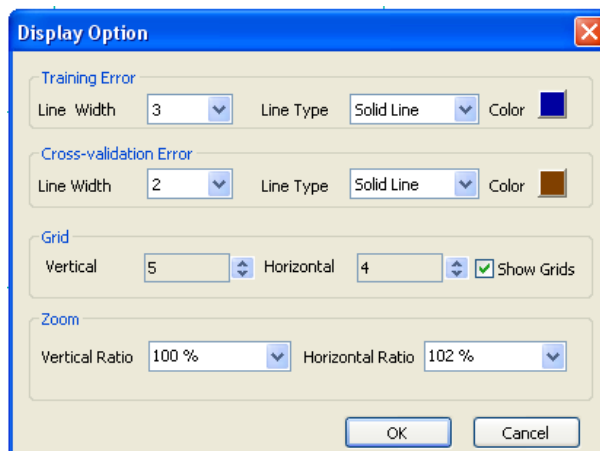


The screening shot displays undergoing model training. Certain functions are disabled during model training. To activate disabled functions, click  button to interrupt the calculation. The vertical grids indicate the number of decision stumps, and the horizontal grids indicate training error or cross-validation error (measured by error ratio). As time passes by, the training error curve moves towards right-hand side until M steps are finished. The legends in the upper-left corner explain the meaning of each curve. For example, the blue curve in this figure shows training error whereas the brown curve shows cross-validation error. In general, training error is minimized in a stepwise manner. As m (the number of finished iterations,  $m < M$ ) increases, training error usually decreases. Conversely, cross-validation error may not necessarily follow the same trend. The cross-validation (CV) error may forms a “U” or an “L” shape as a function of m. LiCABEDS also has the risk of overfitting like most supervised learning algorithms. Therefore, the CV error may reduce at the beginning when the training error is minimized, and then increase due to overfitting, which forms a “U” shape error curve. Running cross-validation is a choice to pick up an optimal value of M.

The program has a default style for displaying training curves. Other styles can be specified as well. Click right mouse button in the main window to show a pop-up menu:

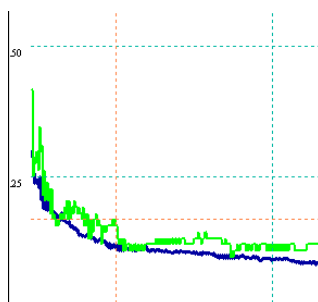


Select “Display Option..”:



First, choose line width of either training error curve or cross-validation error curve in the unit of pixels. Then, users can specify line color and style, such as dotted line and dashed line. The grid section controls the number of vertical and horizontal grids. Un-checking “Show Grids” box will disable any assistant grids in the main window. Finally, set the zoom ratio through “Vertical Ratio” and “Horizontal Ratio” drop-boxes. After clicking “OK”, the setting will become the default display format for all FCC program sessions.

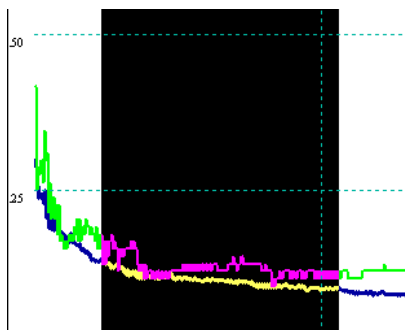
To keep track of coordinate information, check “Show Coordinates” from the popup menu (click right mouse button). Next, move the cursor in the window, FCC will automatically track selected “weak learner” or decision stump, which is illustrated in the screenshot below:




The selected “weak learner” is shown in the status bar:

Num:77 T Err:0.113 CV Err 0.167; 000675 0 -> 1

In this example, the cursor is pointing to the 77<sup>th</sup> “weak learner” (Num: 77 in zero-based index). When  $m = 78$ , the training error is 0.113, which means the LiCABEDS model can classify 88.7% of training samples correctly. As cross-validation is enabled, cross-validation (CV) error is also reported at the same time. In this case, the CV error is 0.167. The last part of the text string, “000675 0 -> 1” depicts the 77<sup>th</sup> decision stump in the LiCABEDS model. This decision stump states that if the 675<sup>th</sup> bit (zero-based index) of fingerprint is 0, this compound is classified as positive (positive or negative class is defined in project profile).



FCC also supports selection of range “weak learners” by “dragging” the mouse with left button down. The color of selected decision stumps is inverted, and the summary will be given in the Message Board. All of these features help users to interpret the model and search for the optimal parameters.

 182 classifier(s) selected. Average training error: 0.105; Average cv error 0.126

Tip: FCC can calculate the best display ratio to fit the training curve in the window. Go to View menu and select “Fit to Window Size”. If the size of the 2D-plot is larger than the display window, scroll bars will automatically show up. Shortcut keys are listed to scroll vertically or horizontally and zoom in/out:

Roll the mouse wheel to scroll horizontally

Roll the mouse wheel with “Shift” key down to scroll vertically

Vertical Zoom-in: Ctrl + Z

Vertical Zoom-out: Alt + Z

Horizontal Zoom-in: Ctrl + H

Horizontal Zoom-out: Alt + H

Selected “weak learners” can be copied and pasted into any text-editing software, e.g. Wordpad.

Select part of the model, click  button, and paste the content into a text editor:

```
id 158 bit 626==1->1 weight 0.124393 T Error 0.102607 CV Error 0.130952
id 159 bit 327==0->1 weight 0.122432 T Error 0.102607 CV Error 0.142857
id 160 bit 262==1->1 weight 0.119565 T Error 0.101766 CV Error 0.142857
id 161 bit 186==0->1 weight 0.125820 T Error 0.103448 CV Error 0.130952
id 162 bit 807==1->1 weight 0.115970 T Error 0.100925 CV Error 0.130952
id 163 bit 718==0->1 weight 0.124027 T Error 0.103448 CV Error 0.130952
id 164 bit 574==1->1 weight 0.122359 T Error 0.100084 CV Error 0.130952
id 165 bit 684==0->1 weight 0.120857 T Error 0.101766 CV Error 0.130952
id 166 bit 562==1->1 weight 0.114031 T Error 0.098402 CV Error 0.130952
id 167 bit 359==0->1 weight 0.122086 T Error 0.103448 CV Error 0.130952
id 168 bit 86==1->1 weight 0.119489 T Error 0.096720 CV Error 0.130952
id 169 bit 440==0->1 weight 0.119795 T Error 0.099243 CV Error 0.130952
id 170 bit 654==1->1 weight 0.099239 T Error 0.094197 CV Error 0.130952
```

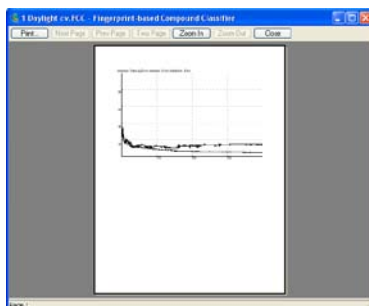
id field indicates the zero-based index of decision stumps.

bit field tells the hypothesis in the corresponding decision stump, e.g. the 158<sup>th</sup> decision stump means if 626<sup>th</sup> bit is equal to 1, then the compound is classified as positive.

weight is equal to  $a_m$  in LiCABEDS model, which tells the contribution to final prediction.


These fields are followed by T error (training error) and CV error (cross-validation error if enabled).

LiCABEDS error curve can be output to any installed physical or virtual Windows printer, like PDF printer. FCC supports what-you-see-what-you-get printing preview. Click “File -> Print Preview” and the printing preview will be shown in the program window:



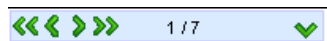
The figure can be sent to a specific printer through a uniform printing dialog for archive or publication.

#### Step Four: Make Predictions


Once a predictive LiCABEDS model is developed and testing compound dataset is loaded, categorical labels can be predicted for the testing compounds. Click  button on the toolbar, a prediction dialog will popup:

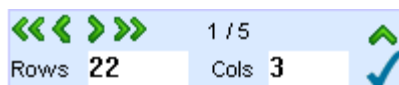
It is optional to use part of the decision stumps to make predictions, even if model is trained using a large value of  $M$ . For example, a LiCABEDS model is trained with 10000 iterations, only the first 1000 decision stumps may be used for predictions. If “Use All the Weak Classifiers” is checked, the corresponding edit box turns grey and the whole LiCABEDS model is used for prediction. Otherwise, users can input arbitrary number, as long as it is smaller than  $M$ . The other parameter in the dialog is the boundary of reject option. LiCABEDS can not only output the categorical value for each testing sample, but also output the degree of confidence for each prediction. The rationale and benefit have been discussed in previous chapter. If prediction confidence is below the boundary value, LiCABEDS will

output an “unknown” label instead of making a risky prediction. A typical view of the prediction window is displayed below:



ID	Compound Name	Prediction	Raw Value	ID	Compound Name	Prediction	Raw Value
1	L011394	Agonist	5.284	21	L022655	Agonist	2.602
2	L001401	Antagonist	-4.036	22	L023057	Antagonist	-3.111
3	L005495	Agonist	5.317	23	L010107	Agonist	5.747
4	L008641	Agonist	4.229	24	L021250	Agonist	3.183
5	L008820	Agonist	5.665	25	L015126	Agonist	4.586
6	L012936	Agonist	8.152	26	L003590	Agonist	4.200
7	L004095	Agonist	6.149	27	L014499	Agonist	6.854
8	L004092	Agonist	9.518	28	L007004	Agonist	2.345

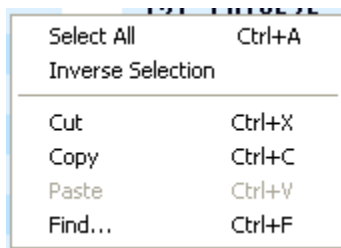
At the same time, a prediction summary is also given in the Message Board. The toolbar at the top of the prediction page allows user to browse through the predictions and change display layout. Click the button  to expand this toolbar:



“<<” : go to the first page; “<”: go to the previous page; “>” go to the next page; “>>” go to the last page. These buttons are followed by page number. It is allowed to change the number of rows and columns displayed in each page. When the program starts, the number of rows and columns are specified automatically to fit content in one page.

In the prediction window, four data fields are displayed for each testing compound: compound ID, compound name, predicted categorical value and raw output value from LiCABEDS model. By default, all the testing compounds are ranked according to their indices in the testing dataset. They can also be sorted according to other fields. Move the mouse cursor to the first row of the table. The shape of the mouse cursor will be automatically changed to up-arrow or down-arrow. Click left mouse button to sort all the testing compounds according to the pointed data field.

Click left mouse button to select a testing compound entry or multiple entries with “Ctrl” key down. Users can open a popup menu by clicking right mouse button to select all the data (Ctrl + A) or inverse the selection. The “Cut” function deletes selected testing data and put them into clipboard (Note: the data in the clipboard will be overwritten if another copy or cut request is made).

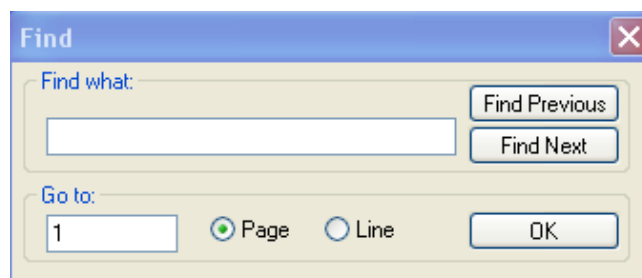


The selected data entries can be copied to clipboard and pasted to text editing software or a session of FCC program. The content copied to text editor still includes compound id, name, predicted category and raw prediction value (screenshot is shown below)

74	L001578	Antagonist	-4.199
2	L001401	Antagonist	-4.037
178	L009388	Antagonist	-3.840
212	L011302	Antagonist	-3.661

The data saved in the clipboard can be also pasted to the same session of FCC or another session of FCC program. Nevertheless, different sessions of FCC may hold different predictive LiCABEDS models, so the copied predicted category value is not necessarily consistent with the existing model. Predictions have to be made for the pasted data entries; otherwise, “unknown” labels will be assigned.

A “Find” dialog is provided to facilitate text searching. The dialog consists of text search box and page/line locating box.





In the “Find what” textbox, input the exact or partial string of a compound name. Then click “Find Previous” or “Find Next” to carry out forward or backward searching. If no item is selected, the searching will start from the first entry in the table. Message board will report whether a matched compound name is found. The retrieved item will be automatically selected and labeled by red rectangle. It may be a good idea to jump directly to a certain page or line instead of clicking “>” button many times, especially when large amount of testing data is imported into the program. Check “Page” or “Line” option button and input a page or line number. After clicking “OK” button, the desired page or data entry will be displayed if available.

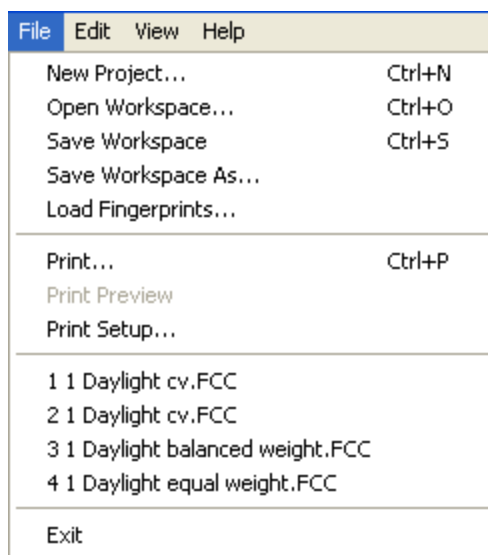
Similar to the printing function in model training module, the prediction results can also be sent to a virtual or physical printer. Once paper size is set, the program formats the layout and outputs the predictions line by line. Printing preview function is not implemented for this purpose. The snapshot shows part of the PDF printout.

ID	Compound Name	Prediction	Raw Value	ID	Compound Name	Prediction	Raw Value
67	L001621	Antagonist	-6.907	28	L007004	Agonist	2.558
74	L001578	Antagonist	-4.351	117	L007968	Agonist	2.497
2	L001401	Antagonist	-4.445	248	L008466	Agonist	2.439
178	L009388	Antagonist	-3.744	147	L022433	Agonist	2.679

### Save Your Workspace and Export Your Results

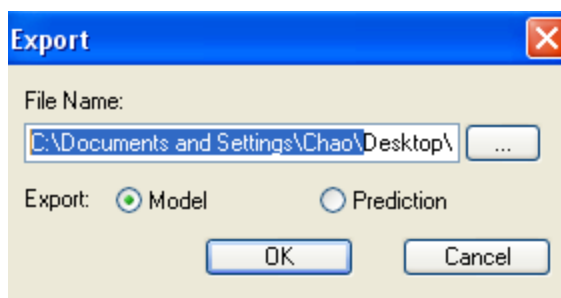
A work session can be serialized and saved into a file, so that the project can be reloaded by another program instance. The saved workspace contains project profile, training and testing compound datasets, developed model and predictions. Before exiting the program, a dialog box will popup to remind saving the workspace if any changes has been made to the project. To save the project, click  button on the toolbar. If the project is not associated with a workspace file, a file name is needed to save the project. Otherwise, the project will automatically overwrite the associated file. To provide an alternative file name, go to “File” menu and select “Save Workspace As”.

The project workspace file has a default “FCC” extension. To restore a previous workspace, open another FCC program and drag a FCC workspace file into the main window. Another way to open a FCC workspace file is to click  button and select a saved workspace file with extension .FCC. FCC automatically records the most recent workspace files and provides shortcuts for opening those files in the “File” menu.



FCC workspace format is in binary format that can only be understood by the FCC program. Meanwhile, an “export” function is implemented to enable exporting legible text results, such as predictive models and LiCABEDS predictions. To get access to this function, go to “Edit menu” and select “Export”.





Specify an output file name by clicking “...” button and select to output either predictive model or predictions. The exported model is in pure text format:

```
Totally 1100 weak classifiers:
ID      bit      value  weight
0       722      0      0.866193
1       54       0      0.491575
2       92       0      0.469724
3      641      0      0.434143
4      355      0      0.490002
```

Each row includes the index of a weak classifier, its hypothesis and its contribution weight to the final prediction (All the indices are zero-based). For example, the first weak classifier is “if the 722nd bit of fingerprint is equal to 0, the compound is classified as positive, which has weight 0.866193”. The final prediction is the weighted summation of the outcome of each weak learner. If the summation is larger than 0, then output is positive categorical value, otherwise, negative categorical value. Use the exported result to interpret the model and extract key features. The exported predictions are similar to the clipboard text format that is mentioned in the previous section:

```
Totally 276 testing samples:
ID      name      class  raw value
1      L011394  Agonist  5.404472
2      L001401  Antagonist -4.444728
3      L005495  Agonist  5.931770
4      L008641  Agonist  4.333997
5      L008820  Agonist  6.151025
```

### Molprint2D Fingerprint Generation

Molprint2D fingerprint characterizes a two-dimensional compound structure by a set of atom environment defined by Sybyl atom types. Reference regarding Molprint2D fingerprint is available at [http://cheminformatics.org/molprint\\_download/](http://cheminformatics.org/molprint_download/). In our test calculation, Molprint2D outperforms FP2, Unity and MACCS fingerprint. Thus, a short tutorial is given on how to generate Molprint2D fingerprints and import the descriptors into an FCC project.

First step, convert your compound structure file to mol2 format using Openbabel or other cheminformatics toolkit. Molprint2D prefers chemical structures with implicit hydrogen atoms. A FCC project requires three separate files: positive-category training data, negative-category training data and

testing data (if available). Molprint2D software package is developed for Linux platform. Install the software on a Linux computer and use the following command to generate raw Molprint2D fingerprint:

```
mol22aefp input output
```

Input is your mol2 structure file and output is a file name to save the raw fingerprints. Generate the fingerprints for all the datasets.

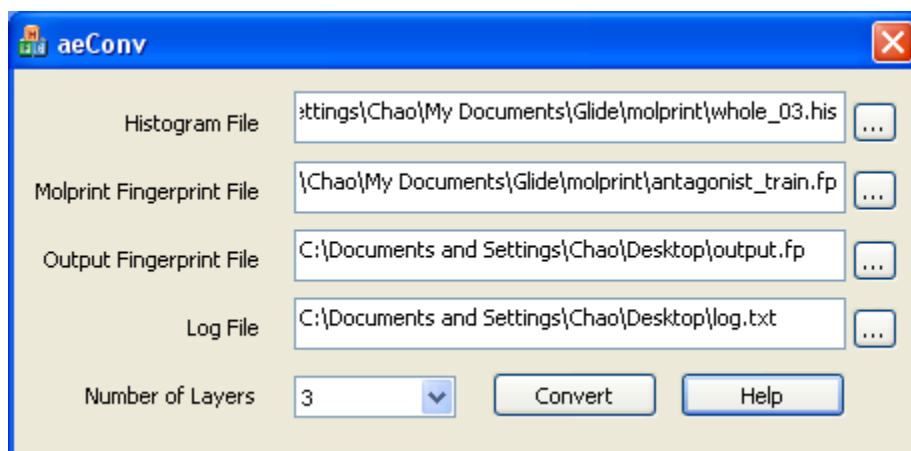
The raw Molprint2D fingerprint is in the following format:

```
L000007 4;0-3-0;1-4-0;2-4-0;2-1-2; 10;0-1-2;0-1-10;1-1-0;1-2-2;2-2-0;2-2-2; 10;0-1-2;0-1-10;1-2-2;2-2-0;2-2-2;.....
```

Each of the Molprint2D feature will be mapped to a unique index that indicates the presence or absence of the feature in a compound. To achieve this, a mapping list should be created. Molprint2D package provides a command “build.pl” to generate a histogram of all the features. In this case, merge all of your structure data files into a single mol2 file and generate raw Molprint2D fingerprints of the merged structure file. Then, use the following command

```
build.pl input his-file -l min max
```

“input” is raw fingerprint file of the merged dataset. His-file is output histogram file. The last parameter “-l min max” means how many layers of neighboring atoms are considered in atom environment. It is usually set to “-l 0 2” for “-l 0 3”. Finally, variable selection can be carried out by modifying the output histogram file or deleting noisy features. Note that only features present in the histogram will be mapped to a dense-format descriptor. A feature will be discarded if it does not exist in the list.



Download Molprint2D converter from the LiCABEDS website. The screenshot of the program is displayed above. A string-matching algorithm is implemented in the converter using hash table. It only takes a few seconds to process a megabyte data file. The converter requires five parameters, including two input files, two output files and one fingerprint parameter. First, locate histogram file by clicking adjacent “...” button and find the histogram file that is generated from “build.pl” command. In the command

“build.pl”, users have an option to specify the number of layers that are considered in atom environment, such as “-l 0 2”. “-l 0 2” means the atom environment contains neighboring atoms up to two bonds away (more details can be found from official Molprint2D website). Choose the number of layers from the drop box. This value has to be consistent with the number of layers in the histogram file. For example, if no “-l” argument follows the “build.pl”, the default maximum layer is 2. If a “-l min max” argument is specified in the command, then the number of layers in the Molprint2D converter should be equal to “max”. The histogram file serves as a mapping table. Next, in the “Molprint Fingerprint File” edit box, locate the raw Molprint2D fingerprint file that will be converted into binary vector format. Finally, specify “log” file name and binary fingerprint file name for output. Press “Convert” button to start data processing. Sample data files are available online:

Histogram.his: a preprocessed histogram file for the converter

Sample Molprint2D.fp: a raw Molprint2D fingerprint file

The output binary fingerprint and corresponding log file are given below:

```
L004275 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
L004425 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
L004429 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
L004433 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
L004458 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
L004473 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
```

The “log” file illustrating mapping mechanism:

```
0;0-1-0;0-1-27;1-1-0;1-2-1;2-1-0;2-2-2;2-2-8; 1
2;0-2-2;0-1-10;1-4-2;2-4-2;2-1-9; 2
2;0-3-2;1-1-0;1-4-2;2-1-0;2-4-2; 3
2;0-1-0;0-1-2;0-1-17;1-2-2;2-1-0;2-1-1;2-2-2; 4
18;0-2-0;0-1-2;1-2-0;1-2-2;2-2-2;2-2-4;2-1-15; 5
2;0-2-2;1-2-2;1-1-18;2-2-0;2-2-2;2-1-3;2-1-7; 6
```

Note: only part of the result is displayed.

The output binary fingerprint complies with the requirement of FCC program, so it can be imported to the program without any modification. Log files reveal the interpretation of fingerprint vector. For example, feature “2;0-2-2;0-1-10;1-4-2;2-4-2;2-1-9;” is mapped to the second bit in the fingerprint vector. In other words, if the second bit of the fingerprint is 1, it indicates the presence of this feature in a compound; otherwise, this feature is absent. Therefore, users can interpret a predictive model and results according to the mapping schema.

### Simple Walk-through Project

Sample training and testing datasets can be downloaded together with the program, FCC. The goal of the walk-through project is to model ligand functionality for 5-HT<sub>1A</sub> G-protein coupled receptor (GPCR). Ligands are classified into two categories: agonists that activate the receptor and antagonists that inhibit or deactivate the receptor. In this experiment, a LiCABEDS model will be developed based on labeled

agonists and antagonists. Then, the model will be used to make predictions on the testing compound datasets. The model performance is evaluated by comparing predicted categorical values with the real labels. Molecular fingerprints have already been generated for all the compounds so that the datasets can be directly imported into FCC program. Here is a list of training and testing data files:

daylight_agonist_train.fp daylight_antagonist_train.fp The training agonist and antagonist datasets. These two datasets are used to develop a predictive model.
daylight_agonist_test.fp daylight_antagonist_test.fp The testing agonist and antagonist datasets. These two datasets are used to evaluate the performance of a developed LiCABEDS model. All the compounds in “daylight_agonist_test.fp” are labeled as agonists. Ideally, the program should output “agonist” for all the testing samples in this dataset. Similarly, all the testing samples in “daylight_antagonist_test.fp” are known to be antagonist.

Protocols:

- Launch the “FCC.exe”
- Click “Project Profile” and input the content as shown below:


Project Title:	<input type="text" value="Ligand Functionality Prediction"/>
Name of Positive Dataset:	<input type="text" value="Agonist"/>
Name of Negative Dataset:	<input type="text" value="Antagonist"/>

Click “OK” button.

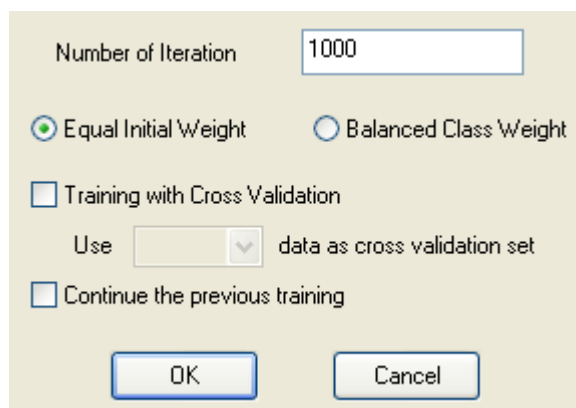
- Click “Load Fingerprints”. Select “Agonist” option and locate file “daylight\_agonist\_train.fp” (click “...” button for browsing).
- Select “Antagonist” option and locate file “daylight\_antagonist\_train.fp”
- The Message Board should display the following information

**Message Board**

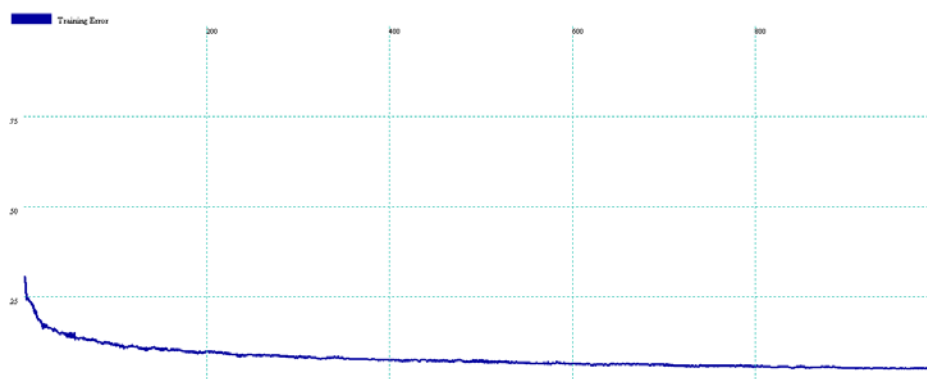
Project Summary  
Project Title: Ligand Functionality Prediction    Name of Positive Training Set: Agonist    Name of Negative Training Set: Antagonist  
Fingerprint Length: 1024    Number of Weak Classifiers: 0  
Size of Agonist: 827    Size of Antagonist: 446    Size of Testing Set: 0

 Compound fingerprints loaded successfully.

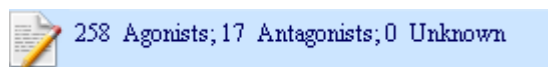
- Click “Setup Training” and input parameters:



- g) Click “OK” and training will start automatically. This process may take a few minutes.
- h) The training error is minimized in a stepwise manner. The screenshot of the error curve is given below:




- i) Load the testing dataset. Go to “File menu” and click “Load Fingerprints...”. Select “testing dataset” and locate file “daylight\_agonist\_test.fp”. Click “OK”.
- j) The main window will automatically display the home page. Click “Make Prediction”. Take default parameters and click “OK”.
- k) The predictions are listed in the main window, and a summary is given in the Message Board. The programs identified 258 compounds as agonists and 17 as antagonists. As we know, all the compounds were labeled as agonists. Therefore, the program made 17 mistakes out of 275 samples.



- l) Load the other testing dataset. Go to “File menu” and click “Load Fingerprints...”. Select “testing dataset” and check “Clean Existing Compounds in the Selected Dataset”. This is an important set to clean up the data imported before. Locate file “daylight\_antagonist\_test.fp”. Click “OK”.

m) Click “Make Predictions” and the result is summarized in the Message Board:

 28 Agonists; 121 Antagonists; 0 Unknown

n) As all the testing compounds are labeled as antagonists, 28 mistakes were made by the program.

The overall prediction accuracy is about 89% (45 mistakes out of 424 testing samples). Finally, press “Ctrl + S” to save your work session. The workspace will be restored to the previous status by loading the work session file.

### 4.3 LICABEDS AND MODELING LIGAND SELECTIVITY

$$\hat{Y} = \text{sign} \left\{ a_1 \begin{array}{c} \text{O} \\ \parallel \\ \text{R}-\text{C}-\text{OH} \\ \swarrow \quad \searrow \\ \text{True} \quad \text{False} \\ \text{Selective (+1)} \quad \text{Non-selective (-1)} \end{array} + a_2 \begin{array}{c} \text{Num of} \\ \text{Rings} \geq 2 \\ \swarrow \quad \searrow \\ \text{True} \quad \text{False} \\ \text{Selective (+1)} \quad \text{Non-selective (-1)} \end{array} + \dots + a_M \begin{array}{c} \text{R} \\ | \\ \text{N} \\ \swarrow \quad \searrow \\ \text{True} \quad \text{False} \\ \text{Selective (+1)} \quad \text{Non-selective (-1)} \end{array} \right\}$$

The cannabinoid receptor subtype 2 (CB2) is a promising therapeutic target for blood cancer, pain relief, osteoporosis, and immune system disease. The recent withdrawal of Rimonabant, which targets at another closely related cannabinoid receptor (CB1), accentuates the importance of selectivity for the development of CB2 ligands in order to minimize their effects on CB1 receptor. In the previous study, LiCABEDS (Ligand Classifier of Adaptively Boosting Ensemble Decision Stumps) was reported as a generic ligand classification algorithm for the prediction of categorical molecular properties. Here, I attempted to extend the application of LiCABEDS to modeling cannabinoid ligand selectivity with molecular fingerprints as descriptors. The performance of LiCABEDS was systematically compared with another popular classification algorithm, support vector machine (SVM), according to prediction precision and recall rate. In addition, the examination of LiCABEDS models revealed the difference in structure diversity of CB1 and CB2 selective ligands. The structure insight from data mining could be useful for the design of novel cannabinoid lead compounds. More importantly, the potential of LiCABEDS was demonstrated through successful identification of a newly synthesized CB2 selective compound.

#### 4.3.1 Introduction

The selectivity to specific drug targets has been a crucial pharmacological property of drug candidates<sup>105</sup>. Since the advent of chemical genetics and chemical genomics<sup>133-134</sup>, more attention has been paid to ligand selectivity for studying biological systems and exploring mechanism of action.

The endocannabinoid system is assumed to regulate psychological process, and is directly related to human mental and physical health. The first cannabinoid receptor was discovered in 1988<sup>135</sup>, later named as CB1 receptor. And a second cannabinoid receptor, CB2 receptor, was identified in human peripheral organs in 1993<sup>136</sup>. These two receptors share high sequence similarity, especially in transmembrane portions. Thus, selective cannabinoid ligands are desired to ensure minimal effect on the other receptor. Recent studies show that cannabinoid CB2 receptor may serve as potential targets for many diseases, including neurodegenerative disorders<sup>137</sup>, blood cancer<sup>138</sup> and osteoporosis<sup>139</sup>. On the other hand, a famous drug, rimonabant that targets cannabinoid CB1 receptor to treat obesity, was withdrawn from the market due to its side psychotropic effects<sup>140</sup>. In spite of the therapeutic potential of CB2 receptor, the suspension of rimonabant reemphasizes the importance of selectivity regarding the design of CB2 agonists and antagonists.

Felder et al reported the first CB2 selective ligand, WIN-55,212-2, that showed 19 fold higher binding affinity for CB2 than for CB1<sup>141</sup>. Since then, the discovery of novel CB2 selective ligands has become the endeavor of many scientists, including medicinal chemist J.W. Huffman<sup>142</sup>. Nowadays, computer-aided drug design has been integrated into the pipeline of drug discovery process to accelerate traditional experimental screening, which requires significant efforts. Thus, the search for CB2 selective ligands is undoubtedly one of its application domains. For example, Ashton et al applied homology modeling and molecular docking to develop CB2 specific ligands<sup>143</sup>. Besides structure-based drug design, machine learning and pattern recognition are gaining popularity for virtual screening and the prediction of various molecular properties. In the context of selectivity profiling, Wassermann et al presented the prediction of ligand selectivity using support vector machine ranking strategies<sup>105</sup>.

Ligand Classifier of Adaptively Boosting Ensemble Decision Stumps (LiCABEDS)<sup>144</sup> is a generic ligand classification algorithm for the prediction of categorical ligand properties. LiCABEDS is established on Freund and Schapire's ensemble learning framework<sup>118</sup>. The underlying theory and its application on modeling ligand functionality have been outlined before. In this study, the performance and applicability of LiCABEDS for the prediction of cannabinoid ligand selectivity are explored. LiCABEDS was compared with famous supervised learning algorithm, support vector machine (SVM), in order to evaluate their performance in recovering true selective ligands. Performance measures were given by precision, recall rate, the geometric mean of both, and ROC curve with area-under-curve (AUC). An ideal classifier would recover all selective ligands from a screening library (100% recall rate), and have no false positive error (100% precision). In addition, the investigation of LiCABEDS models brought certain insight into structure diversity of CB1 selective and CB2 selective ligands. Supported by scaffold and fragment analysis, it was hypothesized that CB2 selective ligands tended to be more structurally diverse than CB1 selective compounds among discovered cannabinoid ligands. More



interestingly, LiCABEDS model successfully identified a true CB2 selective ligand from newly synthesized compounds. The following sections report datasets involved in model training and validation, comprehensive computation protocol of LiCABEDS and SVM, and results of systematic calculations.

### 4.3.2 *Methods, Materials and Calculation*

To demonstrate the performance and robustness of machine learning algorithms in the prediction of cannabinoid ligand selectivity, 703 chemical structures and their bioactivity (Ki value) to CB1 and CB2 receptors were retrieved from public cannabinoid ligand database ([www.cbligand.org](http://www.cbligand.org)). A selective ligand is usually defined descriptively as its differing binding affinity to form ligand-protein complex with different receptors. In this study, a ligand was regarded as CB1selective (or CB2 selective) as long as its ratio of CB1 Ki to CB2 Ki (CB2 Ki to CB1 Ki) was less than 0.1. In other words, a ligand possessed selectivity of cannabinoid receptor subtypes if it exhibited more than 10-fold Ki difference. The rationality of this criterion was supported by an expert in selective cannabinoid ligand discovery, J.W. Huffman<sup>142</sup>, who commented that *O*,2-propano- $\Delta^8$ -THC analogues<sup>145</sup> exhibited modest CB2 selectivity. The CB1/CB2 Ki ratio of these analogues, reported by Reggio et al, ranged from 2.8 to 4.5. Following 10-fold Ki threshold, 149 compounds were identified as CB1 selective; 147 compounds were identified as CB2 selective; the remaining compounds were treated as non-selective. The goal of the study was to distinguish CB1 selective compounds from the CB1 non-selective (including CB2 selective and non-selective), and CB2 selective compounds from the CB2 non-selective (including CB1 selective and non-selective).

**4.3.2.1 LiCABEDS** The performance, robustness, interpretability and parameters of LiCABEDS were thoroughly discussed through modeling 5-HT<sub>1A</sub> ligand functionality<sup>144</sup>. Here, LiCABEDS is extended to model cannabinoid ligand selectivity. Briefly, the prediction in LiCABEDS is determined by weighted summation of a set of “weak” classifiers, i.e. decision stumps. Each decision stump outputs a predicted categorical label according to whether the testing sample possesses a specific compound fragment or structure pattern.

Figure 4-11 visualizes the underlying mechanism of LiCABEDS prediction model. Each decision stump outputs a categorical value (+1 or -1) that represents selectivity label by examining the presence of a predefined structural pattern. The final prediction is the summation of the predictions of individual decision stumps weighted by constant  $a_i$ . Even though the performance of each individual decision stump

is not necessarily much better than a random guess, ensemble learning theory shows that strong classifiers can be obtained by boosting these weak learners.

$$\hat{Y} = \text{sign} \left\{ a_1 \begin{array}{c} \text{O} \\ \parallel \\ \text{R} \text{---} \text{C} \text{---} \text{OH} \\ \swarrow \quad \searrow \\ \text{True} \quad \text{False} \\ \text{Selective} \quad \text{Non-selective} \\ (+1) \quad (-1) \end{array} + a_2 \begin{array}{c} \text{Num of} \\ \text{Rings} \geq 2 \\ \swarrow \quad \searrow \\ \text{True} \quad \text{False} \\ \text{Selective} \quad \text{Non-selective} \\ (+1) \quad (-1) \end{array} + \dots + a_M \begin{array}{c} \text{R} \\ \swarrow \quad \searrow \\ \text{True} \quad \text{False} \\ \text{Selective} \quad \text{Non-selective} \\ (+1) \quad (-1) \end{array} \right\}$$

Figure 4-11: Graphical illustration of the constitution of a LiCABEDS classifier

**4.3.2.2 Support Vector Machine** Support Vector Machine (SVM) shares certain similarity with adaptive boosting regarding margin maximization. Both algorithms have native margin-maximization mechanism to control overfitting except that margins in these two algorithms are expressed in different norms. The binary classification in standard SVM setting is formulated by function  $f_{w,b}(\mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x} + b)$ .  $\mathbf{w}$  is the normal vector to the decision hyperplane. The motivation of support vector machine lies in that the optimal decision surface has the largest distance to the nearest data points of both categories, *i.e.* margin maximization, and still classifies training samples correctly by satisfying  $y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \forall i$  where  $y_i$  is label of training data. Given the same training data sets  $\{\mathbf{x}_i, y_i\}$ ,  $i = 1$  to  $n$ , the margin optimization problem can be transformed into constraint optimization:

$$\begin{aligned} & \text{Minimize}_{w,b} \|\mathbf{w}\|^2 \\ & \text{Subject to } y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \quad \forall i, i = 1, \dots, n \end{aligned}$$

Among all the constraints, the training samples that are relevant to the determination of margin are called “support vectors”. Nevertheless, this optimization is not guaranteed with a solution since the training data may be not linearly separable due to outliers, higher-order data patterns, and many other reasons. To avoid this issue, some “hard-to-classify” training data are ignored by introducing the concept of soft margin. In this case, the previous optimization setting becomes

$$\begin{aligned} & \text{Minimize}_{w,b} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i \\ & \text{Subject to } y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i \text{ with } \xi_i \geq 0, \forall i, i = 1, \dots, n \end{aligned}$$

$\xi_i$  is an error term. Besides margin maximization (or minimizing  $\|\mathbf{w}\|^2$ ), another goal is to maintain a low training error ( $\sum_{i=1}^n \xi_i$ ). The tradeoff between margin and training error is regularized by a specified constant,  $C$ . Even if this constraint optimization can be solved by standard quadratic programming

packages, it is beneficial to investigate its dual form because it brings some valuable properties, for example, insight into support vectors:

$$\text{Maximize } L_{\alpha} = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \quad (1)$$

$$\text{Subject to } \sum_{i=1}^n \alpha_i y_i = 0 \text{ with } 0 \leq \alpha_i \leq C \forall i, i = 1, \dots, n$$

$\alpha_i$  are support vector coefficients, and vector  $\mathbf{w}$  can be recovered by equation,  $\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i$ . Four types of molecular fingerprints, including MACCS key<sup>146</sup>, Unity (www.tripos.com), FP2<sup>127</sup>, and Molprint 2D<sup>77, 124</sup>, were generated as descriptors to represent each cannabinoid ligand in machine learning algorithms. MACCS key (166 bit) and FP2 (1024 bit) were calculated using OpenBabel, and Unity fingerprints (992 bit) were generated in Tripos Sybyl. The whole cannabinoid ligand dataset produced 2530 three-layer Molprint 2D features from Bender et al's program. These features were mapped to binary vector according to previously published protocol<sup>144</sup>. To systematically evaluate prediction accuracy, 50% CB1 selective and 50% CB1 non-selective compounds were randomly selected as training set in order to build a prediction model. The remaining compounds, which were not present to learning algorithms, were used to test the model. The calculation strategy was also applied to CB2 selectivity modeling. Two machine learning algorithms (LiCABEDS and SVM) combined with four types of fingerprint and two receptor subtypes resulted in 16 calculation settings. Each calculation setting was repeated for 20 times on different randomly selected training and testing samples, with intention to assess stability and reliability.

All calculation regarding LiCABEDS was automated with published program ([www.cbligand.org/LiCABEDS](http://www.cbligand.org/LiCABEDS)). The SVM-based selectivity modeling approach was carried out using library "e1071", which included an R wrapper of LIBSVM<sup>147</sup>. This chapter also reports the effect of cross-validation on the prediction performance of testing data sets. With cross-validation, 30% of training data was left out as cross-validation set in order to choose optimal training parameters. For LiCABEDS, the training parameter was the number of "decision stumps", or training iterations; while the parameter of SVM was the constant  $C$  that regularized the tradeoff between training error and margin size. Then, all the training data was used to train a model with the optimal parameter specified during cross-validation. During this study, a LiCABEDS model was developed using all labeled compounds to predict the CB2 selectivity of 12 newly synthesized compounds.

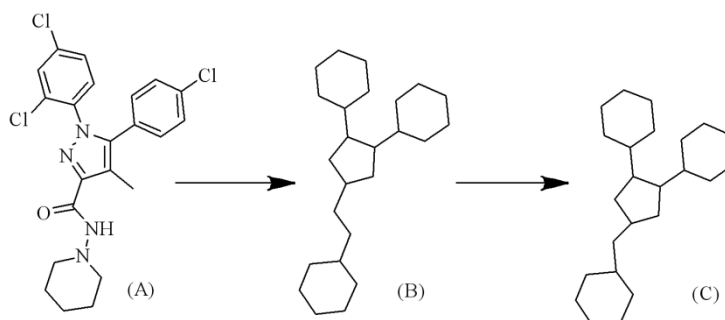
The following performance metrics were calculated for each calculation setting:

- Precision =  $\frac{\text{\#True Selectives}}{\text{\#True Selectives} + \text{\#False Selectives}}$
- True Positive Rate (TPR) =  $\frac{\text{\#True Selectives}}{\text{\#True Selectives} + \text{\#False Non-selectives}}$
- False Positive Rate (FPR) =  $\frac{\text{\#False Selectives}}{\text{\#False Selectives} + \text{\#True Non-selectives}}$

Precision measured the percentage of correctly identified selective compounds, and Recall Rate (or True Positive Rate) depicted the capability of prediction model to retrieve or recover selective compounds.

Geometric mean of Precision and Recall Rate could serve as a single criterion for performance rating. ROC (receiver operating characteristic) curve, plotting TPR versus FPR, showed the enrichment of true selective compounds with varying decision threshold.

Structural skeletons of cannabinoid ligands were generated and compared for the exploration of structure diversity in selective CB1 and CB2 ligands. Briefly, compounds were reduced to carbon skeletons by deleting all non-ring substituent except linkers between ring systems, replacing all heteroatoms with carbon atoms, and converting all bond orders to single bonds<sup>148</sup>. Following these steps, generic compound scaffolds were generated by shrinking the linker chain from a sequence of two or more CH<sub>2</sub>s to only one CH<sub>2</sub>. Figure 4-12 exemplifies the procedure of scaffold generation with a CB1 selective ligand, SR141716. (A) The original structure of SR141716. (B) Its carbon skeleton after deleting the side chains, such as the '=O', '-CH<sub>3</sub>' and '-Cl' groups, replacing all non-carbon atoms, in this case nitrogen atoms, with carbon atoms and converting all bond orders to single bonds. (C) General carbon skeleton is produced by shrinking the linker chain of the carbon skeleton from two or more CH<sub>2</sub>s to one CH<sub>2</sub>. (Scaffold analysis was carried out by Dr. Lirong Wang)

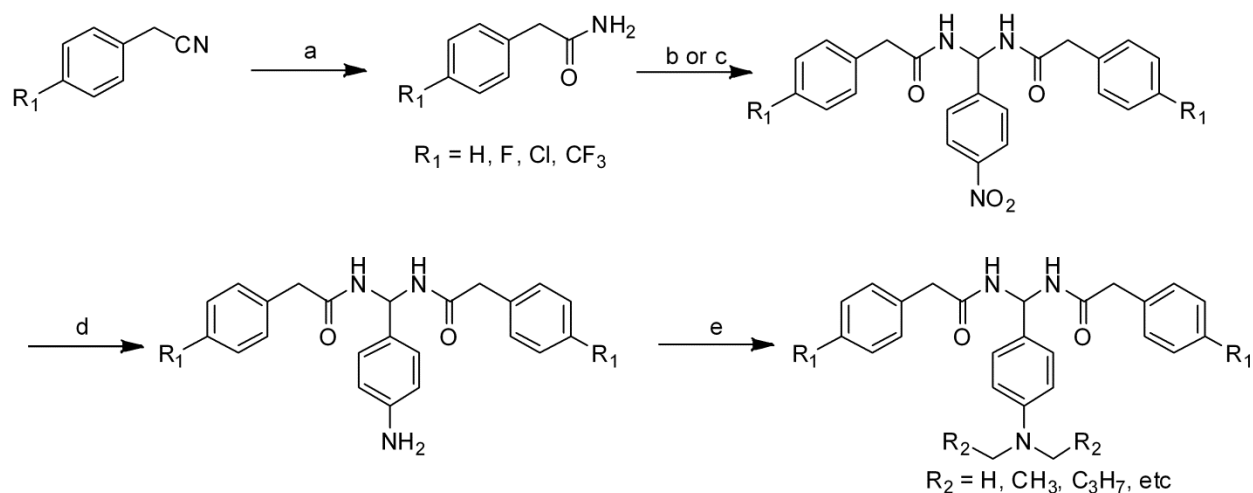


**Figure 4-12: Illustration of scaffold generation**

SR141716, a CB1 selective ligand, as an example to show the procedure of scaffold generation.

**4.3.2.2 Experimental Section** 12 compounds with a novel scaffold were synthesized and tested against CB1 and CB2 receptors. The synthetic route is shown in Scheme 1. The hydrolysis of the substituted 2-phenylacetonitrile in concentrated H<sub>2</sub>SO<sub>4</sub> gave the intermediate amides. And then, the coupling reaction between the amide and aldehyde was performed in anhydrous dichloroethane with the catalyst TMSCl<sup>149</sup>. Alternatively, the coupling reaction was performed in anhydrous DCM with the catalyst F<sub>3</sub>CSO<sub>3</sub>SiMe<sub>3</sub><sup>150</sup>. The nitro compound was reduced with palladium (10%) and hydrazine in ethanol to give the amine, which react with halide to give the final product.

**Scheme 1.** General Synthesis of PAM Analogues



Reagents and conditions: (a) concentrated  $\text{H}_2\text{SO}_4$ ,  $0\text{ }^\circ\text{C}$ , 12 h; (b) method 1: aldehyde, anhydrous dichloroethane,  $\text{TMSCl}$ ,  $70\text{ }^\circ\text{C}$ , 3-12 h; (c) method 2: aldehyde, anhydrous DCM,  $\text{F}_3\text{CSO}_3\text{SiMe}_3$ , r.t., 12 h; (d) ethanol, palladium (10%), hydrazine,  $70\text{ }^\circ\text{C}$ , 3 h; (e) DMF,  $\text{K}_2\text{CO}_3$ , r.t., 12 h.

The binding affinities of these 12 derivatives to  $\text{CB}_2$  receptor were determined by performing [ $^3\text{H}$ ]CP-55,940 radioligand competition binding assays using membrane proteins of the CHO cells stably expressing human  $\text{CB}_2$  receptor. The  $\text{CB}_1$  binding assay was also conducted for those compounds with high  $\text{CB}_2$  receptor binding potency ( $K_i < 1,000\text{ nM}$ ) using membrane proteins harvested from the CHO cells stably transfected with the human  $\text{CB}_1$  receptors.  $\text{CB}_2$  receptor ligand SR144528 and  $\text{CB}_1$  ligand SR141716 were used as positive controls respectively along with the tested compounds. (experimental section was summarized by Dr. Peng Yang)

### 4.3.3 Results and Discussion

This section is focused on the analysis of LiCABEDS, SVM and different molecular fingerprints for their capability of distinguishing cannabinoid selective compounds from non-selective ones. The results are quantitatively supported by various performance metrics that depict aspects of prediction outcomes. The effect of cross-validation on LiCABEDS and SVM is also discussed. In addition, the cross-validation studies reveal that the LiCABEDS model complexity is positively correlated with the recall rate of CB2 selective compounds, but not CB1 selective ones. A possible explanation to this is provided in terms of structure diversity, which gives us deeper understanding of the LiCABEDS mechanism. In the end, a case study shows how LiCABEDS could direct drug discovery and chemical modification by predicting the selectivity of newly synthesized compounds.

#### 4.3.3.1 LiCABEDS and SVM in Default Settings

LiCABEDS was originally designed as a general-purpose ligand classifier for the prediction of categorical ligand properties. The theoretical framework of LiCABEDS shows that LiCABEDS is not necessarily a linear classification algorithm. Nevertheless, when the feature space is represented in binary molecular fingerprints, the training algorithm mentioned in the method section produces a linear classifier. Given that a type of fingerprint defines a pattern set  $S$ , and that a function  $f : S \rightarrow N$  maps each structure pattern to a unique index, there are  $2^{|S|}$  possible decision stumps  $y(\mathbf{x}, i, t) = 2I(\mathbf{x}_i = t) - 1$ , because  $i \in \{id : id = f(s), s \in S\}$  and  $t \in \{0, 1\}$ . Due to the sample space of  $t$  in the context of binary fingerprint, the indicator function,  $I$ , can be omitted by expressing a decision stump as  $y(\mathbf{x}, i, t) = k(2\mathbf{x}_i - 1)$ ,  $k \in \{+1, -1\}$ .  $k = +1$  if  $t = 1$ ,  $k = -1$  otherwise. Therefore, the LiCABEDS

prediction function  $Y_M = \text{sign}(\sum_m^M a_m y_m(\mathbf{x}, i_m, t_m))$  can be updated as  $Y_M = \text{sign}(\sum_m^M a'_m (2\mathbf{x}_{i_m} - 1))$ ,  $a'_m = a_m k_m$ . For a specific  $i$ , define a set  $A_i$  that contains any  $a'_m$  associated with  $\mathbf{x}_i$ .

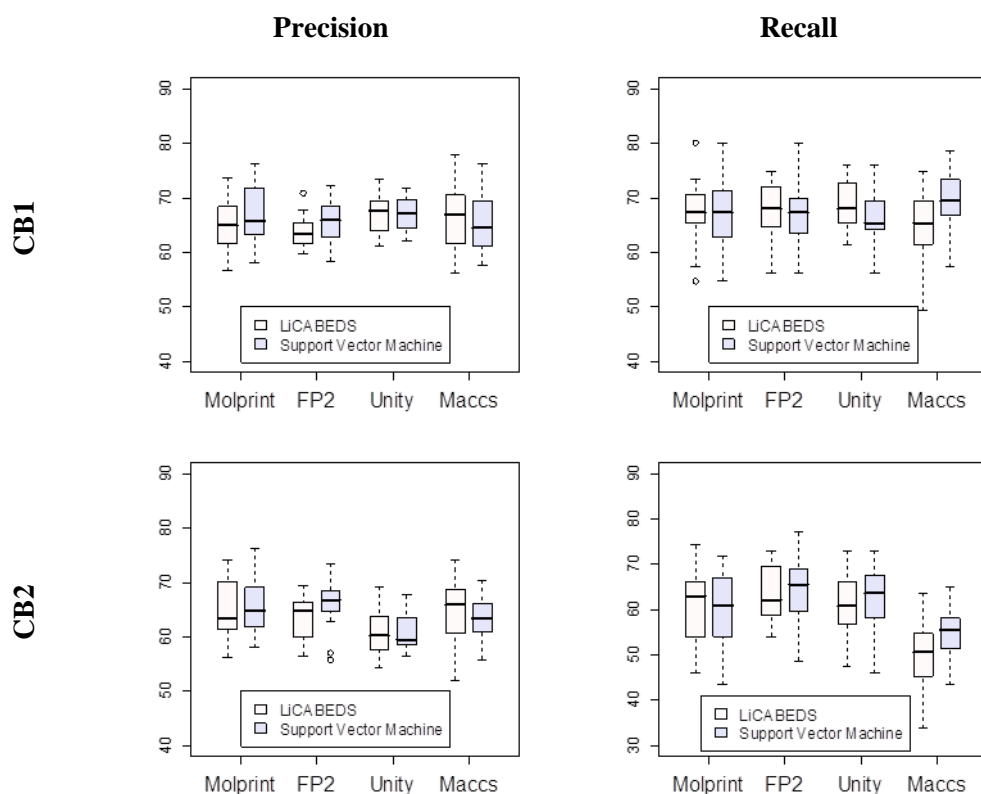
Thus,  $Y_M = \text{sign}(\sum_{i=1}^K (\sum_{a'_k \in A_i} a'_k (2\mathbf{x}_i - 1))) = \text{sign}(\sum_{i=1}^K \beta_i \mathbf{x}_i + \beta_0)$ . This proves the linearity of

LiCABEDS with decision stumps created in a binary feature space.

In previous studies, LiCABEDS was compared with naive Bayes classifier and classification tree (or recursive partitioning method). Support Vector Machine (SVM) is another popular classification algorithm for building linear and non-linear models. SVM has been widely applied in cheminformatics and ligand-based drug design. For the linear nature of LiCABEDS models here, we present a back-to-

back comparison between LiCABEDS and linear SVM. Note that non-linear SVM models may exhibit advantages over linear ones in certain circumstances<sup>151</sup>. Nevertheless, the selection of non-linear kernel functions and parameters is computationally expensive, and could become a risk factor for over-fitting.

This section presents the performance of LiCABEDS and SVM models trained with default parameters (parameter tuning will be discussed later), to show the robustness of these two algorithms. The only parameter in linear SVM is the “*C*”-constant of regularization. Its default value is 1, meaning that training error and margin size are equally important. The only parameter in LiCABEDS training is the number of training iterations or decision stumps, *M*. A reasonable default value of *M* is  $2|S|$ , so that every possible decision stump could be incorporated in the ensemble classifier, even though only some relevant decision stumps contribute to the prediction in real-world problems. By following this principal, the number of LiCABEDS training iterations for Molprint 2D, FP2, Unity and MACCS fingerprints are 5000, 2000, 2000, 400, respectively.



**Figure 4-13: The performance of selectivity prediction without cross-validation**

The boxplot displays the precision and recall rate out of 20 rounds of calculation, using either LiCABEDS or SVM.

**Table 4-6: Model performance without cross-validation**

Molecular Fingerprint	Classification Algorithm	CB1 Selectivity			CB2 Selectivity		
		Precision(%)	Recall(%)	GM	Precision(%)	Recall(%)	GM
Molprint	LiCABEDS	64.8±4.5	67.4±5.7	66.1	64.9±5.1	61.2±7.6	63.0
	SVM	66.8±5.1	66.6±6.3	66.7	65.3±5.1	60.3±7.8	62.8
FP2	LiCABEDS	63.8±2.9	67.8±5.2	65.8	63.7±3.7	63.4±5.9	63.5
	SVM	65.6±3.8	67.2±5.7	66.4	66.3±4.5	64.5±7.0	65.4
Unity	LiCABEDS	67.1±3.9	68.3±4.6	67.7	61.0±4.6	60.7±6.6	60.8
	SVM	67.0±3.0	65.8±5.5	66.4	60.9±3.6	62.6±7.0	61.7
MACCS	LiCABEDS	66.0±6.1	64.6±6.5	65.3	64.4±6.0	50.3±7.0	56.9
	SVM	65.6±5.3	69.5±6.0	67.5	63.4±3.9	54.7±5.2	58.9

The average and standard deviation of precision and recall rate out of 20 rounds of calculation

The results of systematic evaluation of cannabinoid-subtype selectivity prediction are summarized in Figure 4-13 and Table 4-6. Figure 4-13 plots the distribution of precision and recall rate of different computational methods in 20 rounds of calculation. The performance metrics of LiCABEDS and SVM that are trained with different fingerprints are shown. LiCABEDS and SVM models are trained with four types of fingerprints (indicated along X-axis) and default training parameters for the prediction of either CB1 selective or CB2 selective ligands. Y-axis shows performance metrics in percentage value. Correspondingly, Table 4-6 lists the average, standard deviation, and geometric mean of precision and recall rates. The numbers are reported for each combination of machine learning algorithm (SVM or LiCABEDS), molecular fingerprint (Molprint, FP2, Unity or MACCS), and selectivity type (CB1 or CB2 selective). All the SVM and LiCABEDS models are trained with default parameters. GM: geometric mean.  $GM = \sqrt{\text{Precision} \times \text{Recall}}$ . Overall, satisfactory results are achieved. LiCABEDS + Unity outperform other combinations for CB1 selectivity prediction, with the highest geometric mean, 67.7. SVM + FP2 lead the CB2 selectivity prediction, with the highest geometric mean, 65.4. Regardless of fingerprint types, the range of precision and recall of LiCABEDS models covers 56.1%-77.9% and 49.3%-80.0% for CB1 selectivity, 52.1%-74.1% and 33.8%-74.3% for CB2 selectivity. At the same time, SVM models yield precision of 57.7%-76.4% and recall of 54.7%-80.0% for CB1 selectivity, and precision of 55.7%-76.3% and recall of 43.2%-77.0% for CB2 selectivity. In spite of the variability, the precision and recall rate stay above 50% in most cases.

In general, all the fingerprints exhibit decent predictability in LiCABEDS and SVM. According to Table 4-6, Unity fingerprint is the optimal choice for screening CB1 selective compounds, since it achieves a 67% precision rate in both machine learning algorithms. FP2 fingerprints produce the highest geometric mean of precision and recall for CB2 selective compounds (63.5% for LiCABEDS and 65.4%).



Also, the best recall rate of CB2 models are established on FP2 fingerprints (63.4% for LiCABEDS and 64.5% for SVM). Figure 4-13 reveals consistent high precision rate of Molprint 2D fingerprint for both CB1 and CB2 selectivity prediction. MACCS key is weak at recovering the CB2 selective, but is surprisingly sufficient for the CB1 selective. Altogether, Molprint 2D, FP2 and Unity have subtle difference in terms of performance metrics, leaving MACCS key slightly behind.

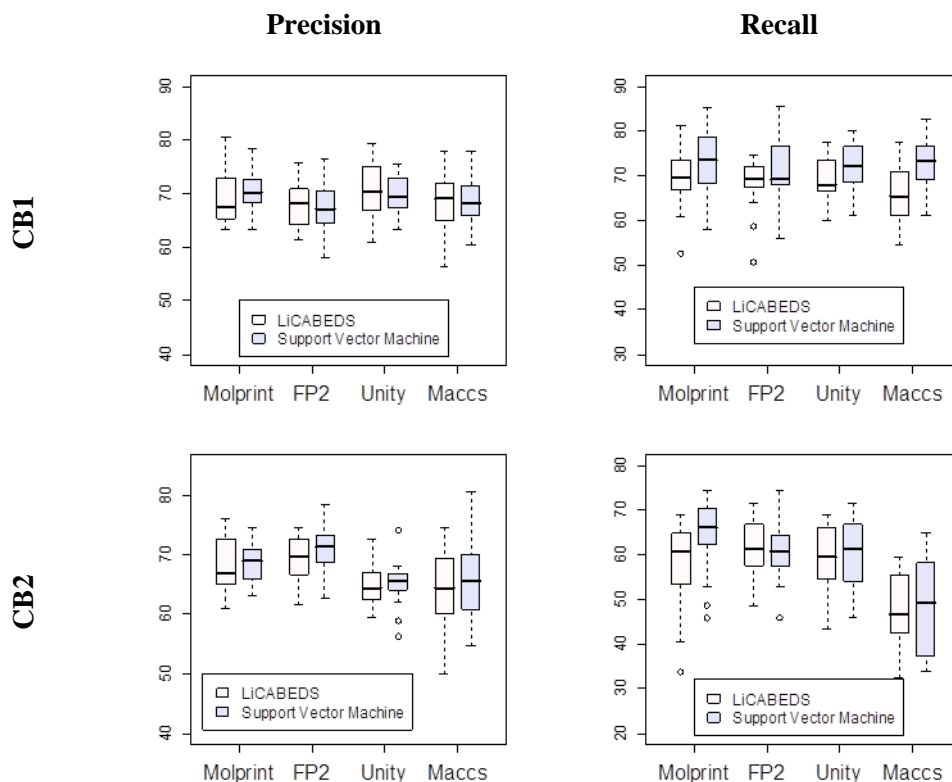
With four types of fingerprints, the geometric mean (GM) gap of LiCABEDS and SVM models falls in the interval (-2.2, 1.3). Table 4-6 suggests that SVM generally outperforms LiCABEDS. However, the outperformance is insignificant when visualized in Figure 4-13 or compared with the standard deviation in Table 4-6. Another interesting fact is that the precision and recall rate of CB1 models are uniformly higher than those of CB2 models, even if the computation protocol is identical. Section 3.3 will explain and discuss this in depth.

The evaluation and comparison of modeling strategies are quantified according to precision, recall rate, and the geometric mean of both. Precision should be emphasized when the cost of validating a predicted selective is high. On the other hand, recall rate catches our attention when many lead compounds are desired at an early stage of virtual screening. Precision and recall rate are usually negatively correlated. An astringent strategy guarantees a high precision rate by selecting high-confident samples, but sacrifices recall rate by ignoring potential true positives. Even if geometric mean is not perfect as a single-number performance measure, it is still a reasonable way to rank screening strategies.

These analyses assume correct selectivity label of training and testing compounds. Due to systematic and random error in bioassays, this assumption is obviously not rigorous in practice. For example, Huffman et al pointed out that the CB2/CB1 affinity ratio of a famous cannabinoid, WIN-55,212-2, was reported in range 0.6 to 30<sup>142</sup>. Here, this compound could be CB2 selective or non-selective. The inconsistent selectivity label in training and testing compounds cause confusion and uncertainty. To minimize this effect, bio-affinity values of a cannabinoid ligand were extracted from the same literature if possible, and use the Ki values reported for the same cell line. However, this systematic error cannot be fully eradicated.

**4.3.3.2 LiCABEDS and SVM with Cross-validation** The ultimate goals of supervised learning algorithms are to minimize generalization error and achieve optimal performance for prospective predictions. In this sense, LiCABEDS and SVM are not exceptions. The sole pursuit of reducing training error may lead to poor predictions for new testing samples (overfitting), while the over-emphasis on margin maximization may result in lack-of-fit. This paradox is also known as “bias-variance tradeoff” in

statistics. Cross-validation is an effective approach to figure out optimal model parameters. In this study, the parameters evaluated in cross-validation are the number of training iterations in LiCABEDS ( $M$ ) and  $C$ -constant in SVM. For LiCABEDS,  $M \in [\text{minstep}, \text{maxstep}]$ .  $\text{maxstep}$  is the default training iterations mentioned previously.  $\text{minstep} = 50$  if MACCS is used as descriptor, otherwise  $\text{minstep} = 100$ .  $C \in \{2^i; i = -16, -15, \dots, 15, 16\}$ . Although the parameter space of LiCABEDS is much larger than SVM, dynamic programming technique makes the cross-validation in LiCABEDS as fast as training a single model. The  $M$  and  $C$  that perform best on cross-validation data are selected for model training. The performance of these models is further evaluated on the same testing sets, and results are shown in Table 4-7 and Figure 4-14.



**Figure 4-14: The performance of selectivity prediction with cross-validation**

The boxplot resembles Figure 4-13 in figure layout.

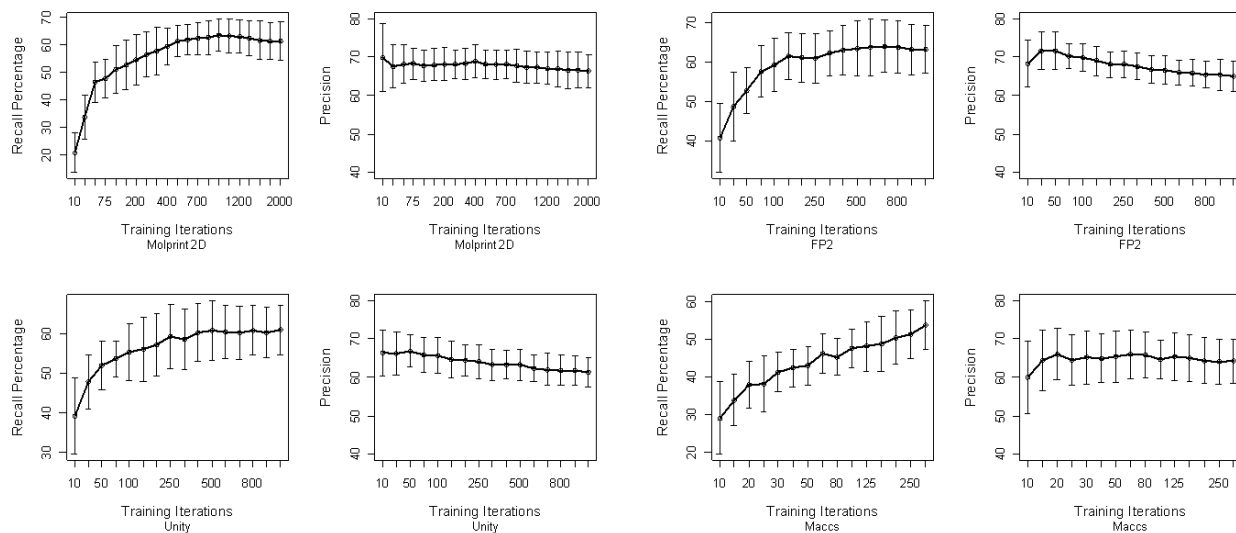
**Table 4-7: Model performance with cross-validation**

Molecular Fingerprint	Classification Algorithm	CB1 Selectivity			CB2 Selectivity		
		Precision(%)	Recall(%)	GM	Precision(%)	Recall(%)	GM
Molprint	LiCABEDS	69.1±5.0	68.5±6.5	68.8	68.4±4.5	58.2±8.9	63.1
	SVM	70.7±4.0	72±7.8	71.3	68.5±3.3	64.5±7.9	66.5
FP2	LiCABEDS	68.0±4.0	68.5±5.8	68.2	69.2±4.1	61.8±6.5	65.4
	SVM	67.5±5.0	70.8±7.4	69.1	71.1±4.0	60.9±6.7	65.8
Unity	LiCABEDS	70.7±4.8	69.2±4.7	69.9	65.0±3.6	59.2±7.4	62.0
	SVM	69.8±3.6	72.2±5.3	71.0	65.2±3.6	60.9±7.8	63.0
MACCS	LiCABEDS	68.0±5.5	66.1±6.2	67.0	63.8±6.5	47.7±8.4	55.2
	SVM	69.0±5.3	72.9±5.6	70.9	65.9±6.6	48.3±10.4	56.4

The performance of LiCABEDS and SVM combined with different fingerprints after running cross-validation

Cross-validation brings consistent improvement for both LiCABEDS and SVM, except for CB2 selectivity prediction using MACCS fingerprint. The geometric mean (GM) of LiCABEDS grows  $\Delta 1.7 - 2.7$  for CB1 selectivity prediction, and  $\Delta -1.7 - 1.9$  for CB2 selectivity prediction. Correspondingly, the geometric mean (GM) of SVM grows  $\Delta 2.7 - 4.6$  for CB1 selectivity prediction, and  $\Delta -2.5 - 3.7$  for CB2 selectivity prediction. Both algorithms are robust enough to deliver satisfactory predictions with default model parameters. Even if cross-validation has positive impact, the improvement is inconclusive when compared with performance variance. SVM seems to be more sensitive to the choice of parameters than LiCABEDS, since the increment of its geometric mean is relatively higher. After parameter tuning, SVM + Molprint 2D outruns other models in both CB1 and CB2 selectivity prediction for its highest GM (71.3 and 66.5). Figure 4-14 shows that, the precision rate of LiCABEDS and SVM is similar among all fingerprints, but SVM is capable of retrieving more selective compounds than LiCABEDS, especially for CB1 selective compounds. In the end, the GM of SVM is 0.4 to 3.9 higher than that of LiCABEDS.

**4.3.3.3 Training Iterations of LiCABEDS** Training iterations ( $M$ ) of LiCABEDS are directly related to model complexity, as every round of training adds one more “weak classifier” to the ensemble model. Parameter  $M$  was thoroughly discussed in the previous study<sup>144</sup>, and the hypothesis was that large  $M$  produced close-to-optimal models. Running cross-validation might improve prediction performance, but its effect was statistically insignificant. The results in section 3.2 also support this hypothesis. Thus, assigning a relatively large value to  $M$  not only guarantees model convergence, but also avoids costly cross-validation procedures and potential overfitting of cross-validation data.



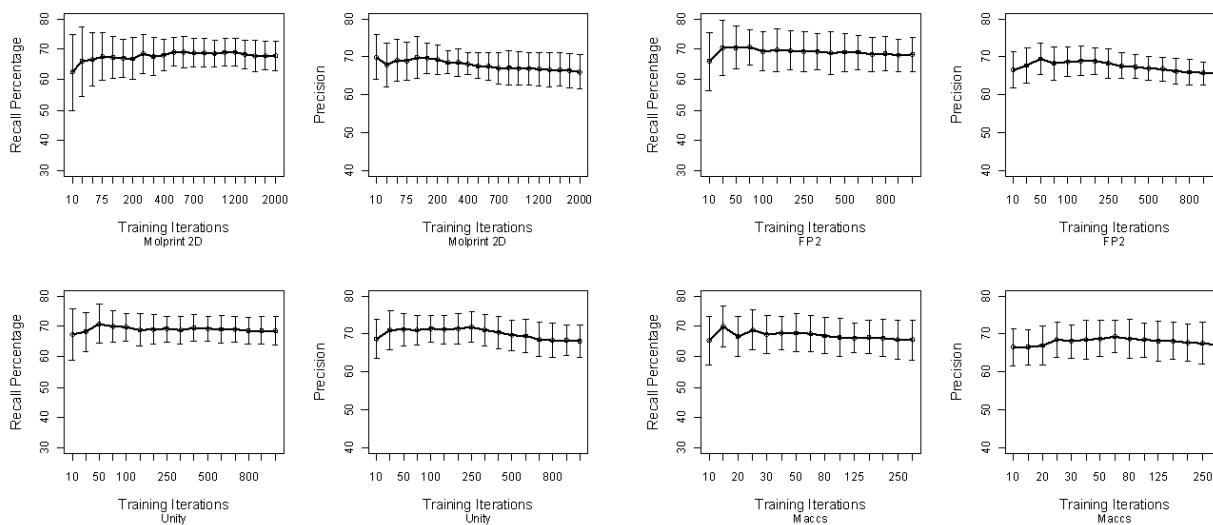
**Figure 4-15: Training iteration and CB2 selectivity prediction**

The plots show the average and standard deviation interval of recall and precision rate of CB2 selectivity models versus model training iterations.

Apart from machine learning language, the meaning of  $M$  can be also interpreted in applied domain knowledge. This section is primarily focused on how  $M$  affects the recovery of selective ligands. Naturally selective compounds are supposed to be overwhelmed by non-selective ones, which is also true for the training and testing sets. Thus, what LiCABEDS training algorithm faces is unbalanced data. Treating each training sample equally at the initialization stage, the training algorithm first assures the correct classification of non-selective compounds even at the cost of misclassification of selective ones. As majority is non-selective samples, this is an effective way of minimizing training error when the classification power of LiCABEDS is limited. Later when the model complexity grows, the training algorithm aims at recovering selective ligands from the training pool by picking up discriminative features and building “decision stumps” accordingly. This process is visualized in Figure 4-15. Figure 4-15 plots the average and standard deviation of recall and precision rates on the testing data sets as a function of training iterations,  $M$ . The average and standard deviation are calculated based on 20 rounds of test calculation. Each row represents a specific fingerprint type. The Y-axis of each plot is in percentage. The values along X-axis represent  $M$ . Note that the growth of X-axis values is not linear. As shown in the left column, the recall rate steadily increases as  $M$  grows. For example, with Molprint 2D fingerprint, the recall rate starts from 20%. It converges to a plateau (approximately 60%) when  $M$  is

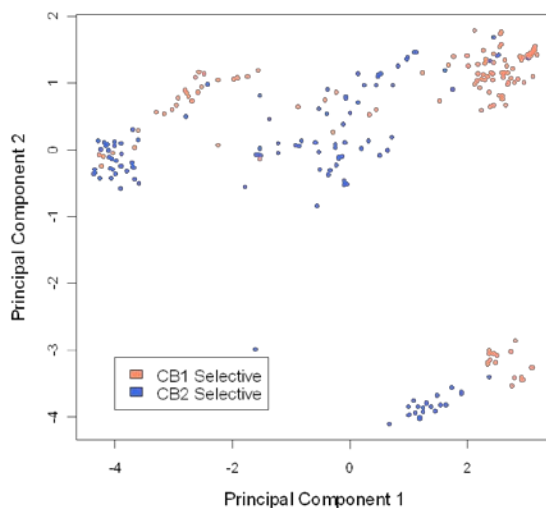
more than 500. Similar trend can be observed in other plots. Meanwhile, the precision is more stable compared to recall rate. It mainly fluctuates from 60% to 70% for Molprint 2D, FP2, Unity fingerprints, and from 50% to 60% for MACCS fingerprint. The precision rate seems to be negatively correlated with training iterations, meaning that a large  $M$  reduces prediction precision. To explain this, easy-to-classify selective ligands are correctly labeled at the beginning stage of training. As training continues, the algorithm tries to recover more selective ligands that are harder to classify. At the same time, more non-selective ligands may be also predicted as selective, which reduces precision. The positive effect of a large  $M$  on recall rate is obviously more significant than its negative effect on precision. Therefore, training error is effectively minimized in the early stage of training. As recall rate reaches the plateau, adding more “decision stumps” only impairs precision rate. The value that seeks a balanced trade-off between these two metrics can be figured out through cross-validation, which has been well addressed in the previous section.

The precision and recall rate of CB1 selectivity models are quite different from those of the CB2 selectivity models. Their values as a function of training iterations are displayed in Figure 4-16. One major difference is that the recall rate of CB1 selectivity models reaches the plateau when  $M$  is about 50 for Molprint 2D, FP2 and Unity fingerprints. The uptrend of recall rate is almost not observable with MACCS fingerprint. Figure 4-16 suggests that precision rate gradually reduces as  $M$  grows, which is similar to Figure 4-15. Thus, a large  $M$  is less favored compared to CB2 selectivity models. This also explains why cross-validation is more beneficial to CB1 selectivity prediction than CB2 selectivity prediction. As mentioned in Section 3.2, cross-validation enhances the geometric mean of CB1 selectivity models by  $\Delta 1.7 - 2.7$ , but  $\Delta -1.7 - 1.9$  for CB2 selectivity models. Furthermore, the average of optimal  $M$  for CB1 selectivity prediction is 283 with Molprint 2D fingerprint. The average for CB2 selectivity prediction is 736 with the same fingerprint. A straight-forward conclusion is that the complexity of CB2 selectivity models is higher than the CB1 models. It also indirectly suggests that the structure of CB2 selective ligands is more diverse, so more factors need to be considered in the classifier.



**Figure 4-16: Training iteration and CB1 selectivity prediction**

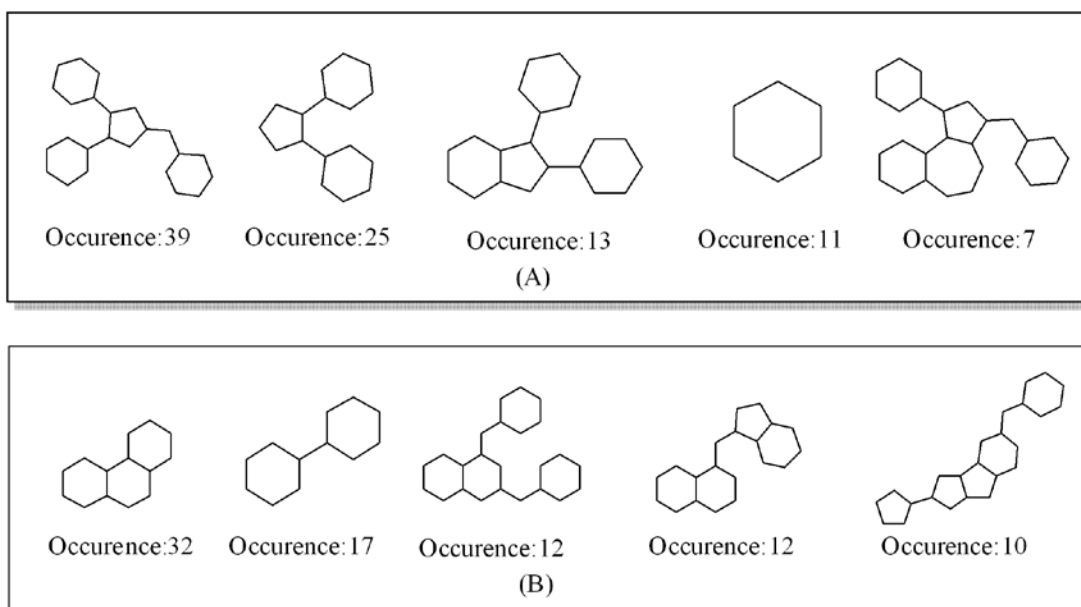
The average and standard deviation interval of recall and precision rate of CB1 selectivity models versus model training iterations.



**Figure 4-17: Principal component analysis of fragments of CB ligands**

2D scatter plot shows the spatial arrangement of selective ligands in the coordinates of two principal components.

A universal quantitative definition of structure diversity does not exist, since it can be evaluated according to aspects of criteria, such as molecular weight, number of rings, etc. We attempt to illustrate the structure diversity of cannabinoid selective ligands in both fragments and scaffolds. Principal component analysis (PCA) is an unsupervised learning skill that transforms original data into a new orthogonal coordinate system in order to capture the maximum variance. In the new coordinate system, the first component has the largest possible variance; the second component has the second largest possible variance; and so on. Figure 4-17 displays the first two components of MACCS fingerprints of all selectivity ligands. The principal components are solved according to MACCS fingerprints of all selective ligands. The X-axis and Y-axis represent the two most significant components. CB1 and CB2 selective ligands are represented in different color. PCA reduces the dimensionality of 166-bit fingerprint for visualization while maintaining minimum information loss. In Figure 4-17, CB1 selective ligands form three clusters that are approximately centered at (-3,1), (2,1), and (2,-3). Only a few points are scattered near origin. Similarly, CB2 selective ligands also form three clusters that roughly centered at (-4,0), (0,0), and (-3,-4). The radius of each cluster is associated with the structure variance of the compounds in the cluster. It is apparent that the CB2 selective ligands near the origin of Figure 4-17 show larger variance than the CB1 selective ligands in any cluster. The other two CB2 selective clusters have similar pattern to those of CB1 selective. Thus, CB2 selective compounds may have more diverse structure features defined in MACCS fingerprint than CB1 selective.



**Figure 4-18: The top five scaffolds in CB selective compounds**  
 (A) CB1 selective compounds (B) CB2 selective compounds

To assess structure diversity in a different aspect, the scaffolds of CB1 and CB2 selective compounds were generated according to the protocol described in Method section. Results showed that 149 CB1 selective compounds possessed 22 scaffolds. Meanwhile, 147 CB2 selective compounds were reduced to 38 scaffolds. Figure 4-18 lists five most populated scaffolds in each compound category. An interesting finding is that the top three CB1 scaffolds have significant overlapping. In other words, the second scaffold in Figure 4-18(A) is a substructure of the other two. On the other hand, CB2 scaffolds have relatively more variation. These observations also support the hypothesis that CB2 selective compounds could be more structurally diverse than CB1 selective compounds. This information could be useful clues for the design of CB selective compounds. (Scaffold analysis was conducted by Lirong Wang)

#### 4.3.3.4 ROC Analysis of LiCABEDS models

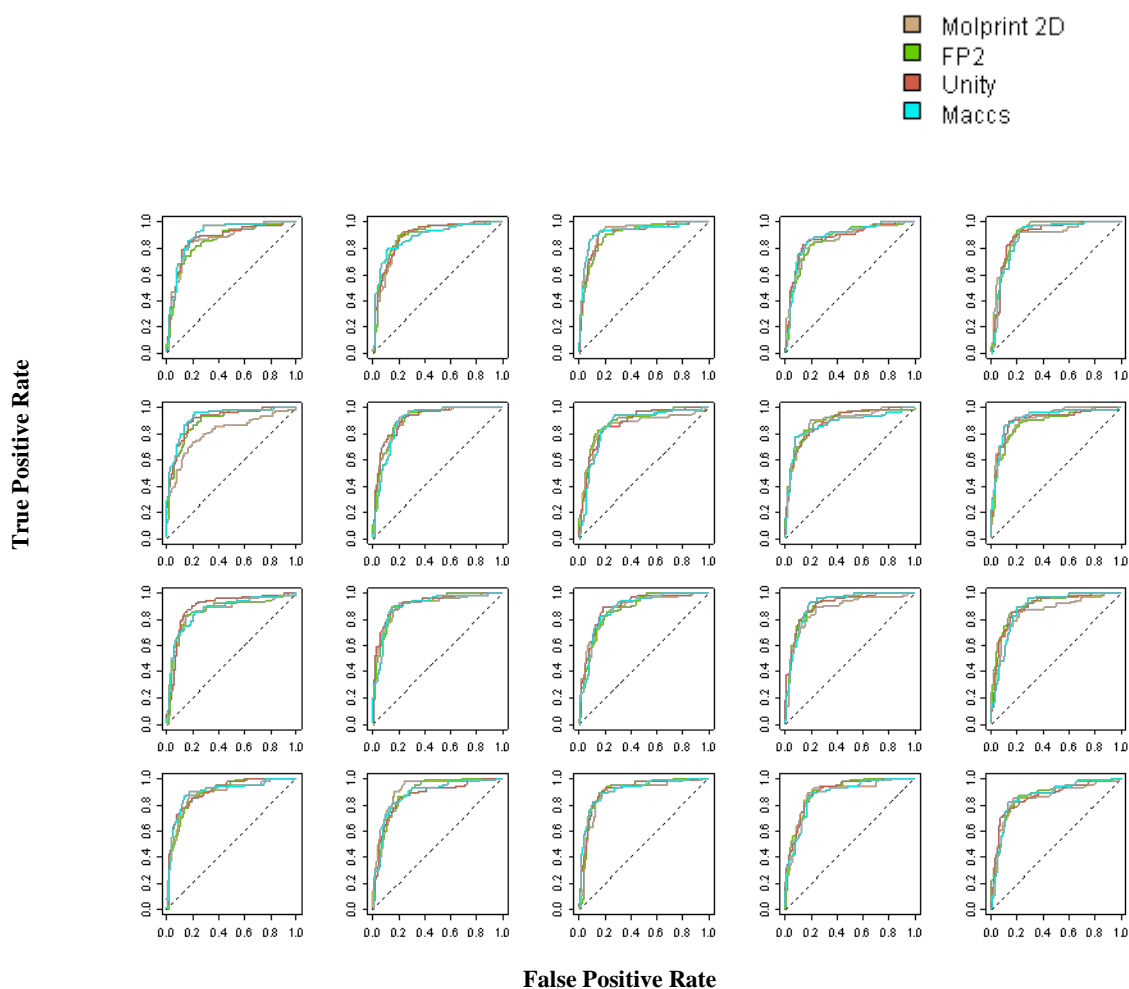


Figure 4-19: ROC curves of LiCABEDS models for the prediction of CB1 selectivity



Sensitive and specificity are classical statistical measures of the performance of binary classification function. ROC curve visualizes the true positive rate (sensitivity) as a function of false positive rate (1-specificity) by changing classification boundary value. Visual inspection of enrichment of positive samples is straightforward by examining ROC curve. Besides the convenience of visualization, the area under curve (AUC) serves as quantitative performance measure of enrichment of relevant samples. The ROC curve of a random guess function (curve in dashed line) yields an AUC of 0.5. A perfect classification function has an AUC of 1.

As discussed,  $A = \sum_m^M a_m y_m(\mathbf{x}, i_m, t_m)$  indicates the degree of confidence in each prediction. The

default classification boundary is  $A = 0$ . Thus, changing the decision criteria generates a set of TPR and FPR values and allows inspecting the enrichment of selective ligands. Figure 4-19 and Figure 4-20 plot the ROC curves of LiCABEDS models for the prediction of CB1 and CB2 selectivity with four types of fingerprints. Each individual plot contains four curves that represent different fingerprints. Out of 20 rounds of test calculation, the average AUC of CB1 selectivity models (Figure 4-19) are 0.883, 0.893, 0.897 and 0.895 with Molprint 2D, FP2, Unity and MACCS fingerprints respectively. Correspondingly, the average AUC of CB2 selectivity models (Figure 4-20) are 0.839, 0.880, 0.855 and 0.839 with the same set of fingerprints. The visualization of these two figures and AUC values confirm that LiCABEDS models effectively enrich selective ligands. In addition, the ROC curve also supports that large  $|A|$  suggests high probability of observing relevant samples, since the left region of ROC curve corresponds to stringent decision criteria. Figure 4-19 and Figure 4-20 reveal that the performance of fingerprints involved in LiCABEDS models is comparable, especially for CB1 selectivity models. Molprint 2D models produce relatively low AUC for CB2 selectivity prediction (average 0.839), which seems to be contradictory to previous findings. In real-world problem, a practical decision boundary corresponds to the left region of ROC curve because large FPR is mostly disfavored and remains unexplored. Partly, the slope of the left part of ROC curve is more important than AUC. As shown in Figure 4-19 and Figure 4-20, as well as AUC calculation, the LiCABEDS models for CB1 selectivity prediction exhibit better enrichment than CB2 models. This also agrees with the previous hypothesis that the structure patterns of CB2 selective ligands are more diverse.

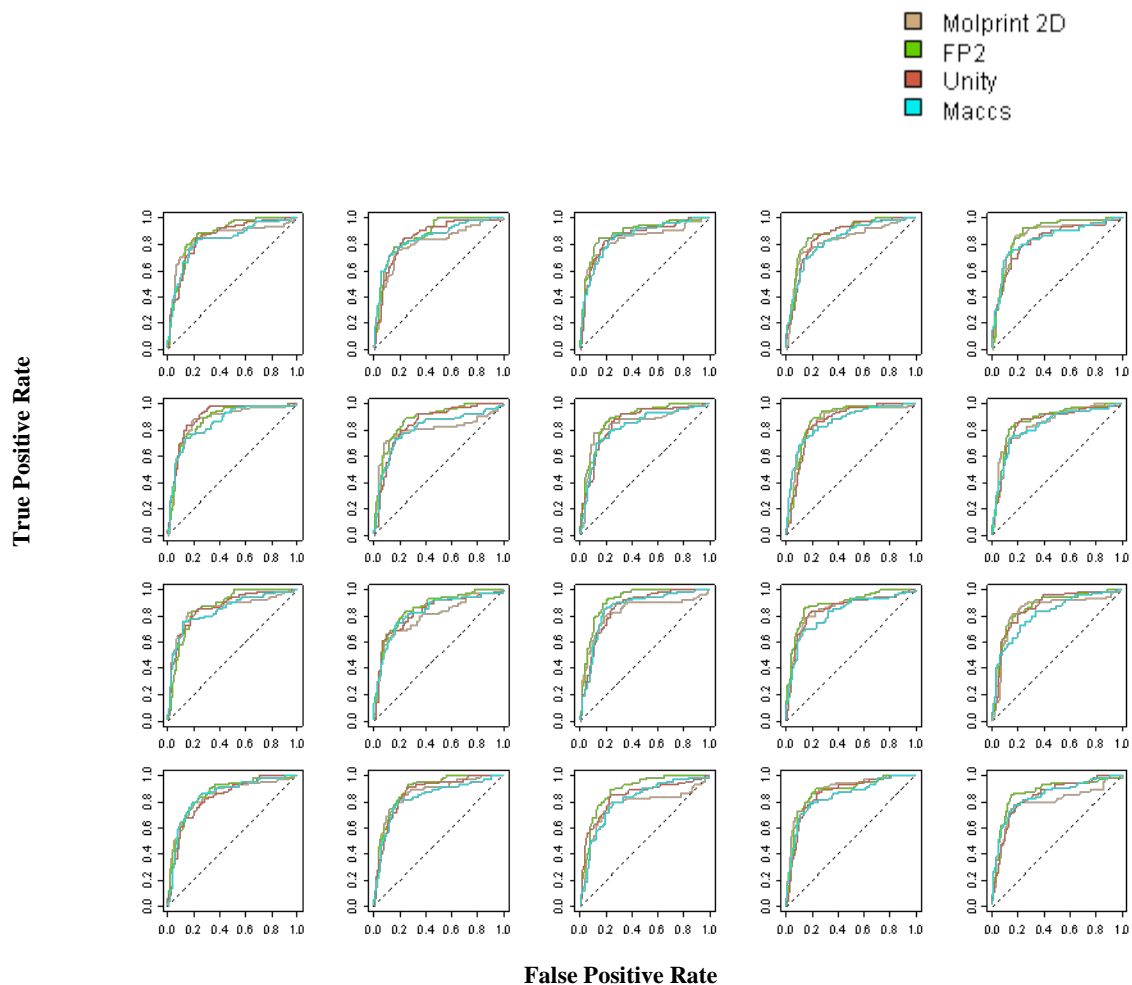


Figure 4-20: ROC curves of LiCABEDS models for the prediction of CB2 selectivity

#### 4.3.3.5 Prediction of Selectivity of Novel Compounds

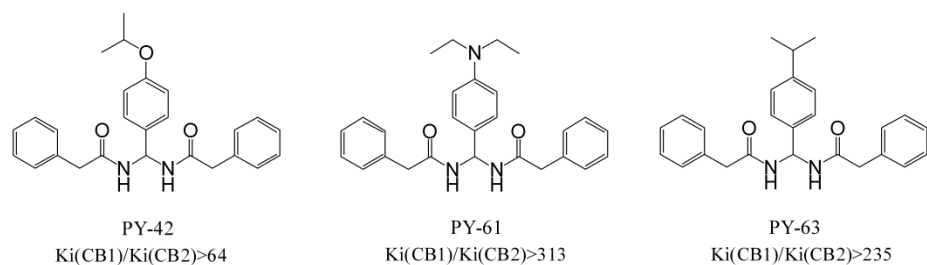


Figure 4-21: The structures of newly synthesized CB2 selective ligands.

Besides retrospective validation, a LiCABEDS model was developed to predict the CB2 selectivity of 12 novel compounds before they were tested. The training set consisted of all the CB2 selective and CB2 non-selective compounds in our cannabinoid database. The training iteration was set to 2000 with Molprint 2D as molecular descriptor. The LiCABEDS model predicted PY-63 (Figure 4-21) as CB2 selective, while the other 11 compound as CB2 non-selective. Later, the experimental validation showed that the three structures in Figure 4-21 (PY-42, PY-61 and PY-63) were CB2 selective, and that the remaining compounds were either non-selective or inactive. Therefore, LiCABEDS successfully discovered PY-63, but missed the other two potential selective compounds. These compounds possessed a novel scaffold that was not present in training data. Interestingly, the prediction model still recovered a true selective without reporting any false positive. Even though the number of testing compounds was limited, this case study illustrated that the expense of drug discovery projects could be minimized by introducing in silico ligand profiling calculation.

#### **4.3.4 Conclusion**

When introduced as a general-purpose ligand classifier, LiCABEDS demonstrated its application for the classification of 5-HT<sub>1A</sub> ligand functionality<sup>144</sup>. This chapter reports a follow-up study of LiCABEDS algorithm in the prediction of cannabinoid ligand selectivity. In-depth discussions are presented on LiCABEDS theoretical framework, performance measures compared with SVM, vulnerability to overfitting, choice of training parameter, and prospective validation. The results show that LiCABEDS models effectively recover 60%-70% selective compounds from testing data sets, and maintain satisfactory false positive rates in the meantime. All fingerprints deliver decent performance metrics, showing the flexibility of LiCABEDS to adapt assorted hypothesis space. More importantly, LiCABEDS successfully identifies a CB2 selective compound out of twelve newly synthesized ones without any false positive error. Another advantage of LiCABEDS is straightforward parameter setting. Default training iteration significantly reduces calculation time by skipping costly cross-validation procedure, but conveys close-to-optimal models. LiCABEDS is not a black-box method, since models are interpretable by examining individual “decision stumps” that ensemble models are composed of. Furthermore, the investigation of LiCABEDS models provides insight into structure diversity of cannabinoid selective ligands. The correlation between performance metrics and model complexity reveals that reported CB2 selective ligands are more diverse than CB1 selective ligands. This hypothesis is later supported by principal component analysis of fragments and analysis of compound scaffolds. It is also well agreed by

some performance metrics, such as AUC of ROC curve. This could suggest that there were more binding modes to form ligand-protein complex for CB2 receptor and for CB1 receptor. To conclude, LiCABEDS has potential to model the pharmacological properties that are not well addressed by traditional QSAR methodology. It could also guide screening strategy in early stage of drug development project.

#### 4.4 LICABEDS AND MODELING BBB PASSAGE

Blood-brain barrier (BBB) is protective mechanism to prevent hazardous substances, e.g. some drugs, from entering brain tissues, while maintaining the permeability of some chemicals to the brain. The ability to pass blood brain barrier is one of the most important pharmacological properties for ligands targeting at central nervous system. Evaluating the passage of the barrier also helps to predict the potential side effects on central nervous system, which may be caused by ligands targeting at periphery organs. Thus, BBB passage is part of Distribution (D) properties in ADME.

To demonstrate the feasibility, a LiCABEDS model was developed on published BBB compound datasets<sup>152</sup> using FP2 fingerprint as descriptor. The training datasets consisted of 832 BBB+ ligands (able to cross the barrier) and 261 BBB- ligands (unable to pass the barrier). The model performance was then evaluated on labeled testing compounds containing 451 BBB+ ligands and 49 BBB- ligands. The result is summarized in Table 4-8.

**Table 4-8: Blood-Brain-Barrier passage prediction for BBB+ and BBB- ligands**

	Label	BBB+	BBB-
Prediction			
BBB+		440	7
BBB-		11	42

The overall accuracy of LiCABEDS on the testing dataset is 96.4%, with 97.6% accuracy for BBB+ category and 86.7% accuracy for BBB- category respectively. The LiCABEDS model outperforms some methods in the publication<sup>152</sup> that reports an overall accuracy ranging from 75% to 97%.

LiCABEDS demonstrates straight-forward prediction logic and high prediction accuracy. Furthermore, an online BBB passage prediction tool is developed based on the LiCABEDS model. The prediction toolkit is accessible at [www.cbligand.org/BBB/predictor.php](http://www.cbligand.org/BBB/predictor.php).

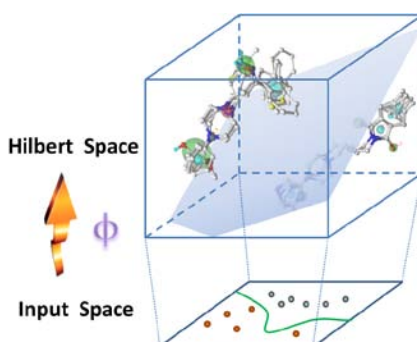


The screenshot shows the web interface for the Online BBB Predictor. It features a chemical sketching toolbar at the top with buttons for CLR, DEL, D-R, +/-, UDO, and JME. Below the toolbar is a large white area for sketching or displaying a chemical structure. The structure shown is 4-isopropylphenol, represented by a benzene ring with an -OH group and an -CH(CH<sub>3</sub>)<sub>2</sub> group. Below the sketching area, there is a text input field and a 'Browse...' button for uploading a file. At the bottom, there are two columns of radio button options for 'Algorithms' and 'Fingerprints'. The 'Algorithms' column has 'AdaBoost' selected. The 'Fingerprints' column has 'Openbabel(FP2)' selected. At the very bottom, there are 'Submit' and 'Return' buttons.

The screenshot atop displays the online BBB passage prediction, maintained by Dr. Lirong Wang. Users can either sketch a compound structure or upload a chemical structure file as queries. Two supervised learning algorithms have been implemented, SVM and AdaBoost (LiCABEDS); and four types of molecular fingerprints are available as descriptors, MACCS, FP2, Molprint 2D and PubChem fingerprints.

(\* Chao Ma developed the back-end prediction program. Dr. Lirong Wang developed SVM classifier and built this webpage.)

## 5 SUPPORT VECTOR MACHINE FOR LIGAND CLASSIFICATION



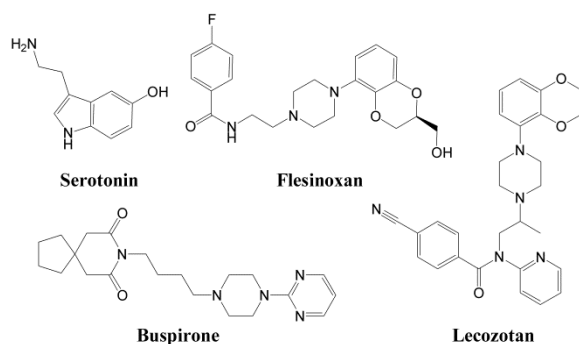
This chapter is an extension to differentiating 5-HT<sub>1A</sub> agonists and antagonists by developing robust non-linear support vector machine (SVM) classifiers. Upon binding to a receptor, agonists and antagonists can induce distinct biological functions and thus lead to significantly different pharmacological responses. Thus, *in silico* prediction or *in vitro* characterization of ligand agonistic or antagonistic functionalities is an important step toward identifying specific pharmacological therapeutics. In this study, we investigated the molecular properties of agonists and antagonists of human 5-hydroxytryptamine receptor subtype 1A (5-HT<sub>1A</sub>). Subsequently, intrinsic functions of these ligands (agonists/antagonists) were modelled by support vector machine (SVM), using five 2D molecular fingerprints and the 3D Topomer distance. Five kernel functions, including linear, polynomial, RBF, Tanimoto and a novel Topomer kernel based on Topomer 3D similarity were used to develop linear and non-linear classifiers. These classifiers were validated through cross-validation, yielding a classification accuracy ranging from 80.4% to 92.3%. The performance of different kernels and fingerprints was analyzed and discussed. Linear and non-linear models were further interpreted through the illustration of underlying classification mechanism. This study expands the scope and applicability of similarity-based methods in cheminformatics, which are typically used for the identification of active molecules against a target protein. These findings provide a

good starting point for further systematic classifications of other GPCR ligands and for the data mining of large chemical libraries.

## 5.1 INTRODUCTION

Agonists and antagonists usually bind to the same site or pocket of a receptor,<sup>153</sup> but they trigger distinct biological responses, resulting in significantly different pharmacological responses. It is known that agonists stimulate the receptor, while antagonists inhibit the receptor function. Nonetheless, agonists and antagonists may share many common features. Analysis of their structures-bioactivity correlation may help to identify important structural patterns and develop agonist/antagonist prediction models.<sup>154</sup> These models further help to design functional molecules. Nevertheless, many of the reported Quantitative Structure-Activity Relationship (QSAR) approaches suffer from low throughput due to the requirement of 3D structure alignments, which may limit their applications in the screening and data-mining of large chemical databases.<sup>155</sup> Herein, we report our investigation of agonists and antagonists for the 5-hydroxytryptamine receptor subtype 1A (5-HT<sub>1A</sub>), since diverse ligands are available to evaluate the various computational approaches for large-scale agonist and antagonist classification.

The 5-HT<sub>1A</sub> receptor is a 5-HT or serotonin (Figure 5-1) receptor belonging to the G protein-coupled receptor (GPCR) family. Primarily expressed in the central nervous system,<sup>156</sup> 5-HT<sub>1A</sub> receptor influences biological and neurological processes, such as aggression, anxiety, appetite, mood, and sleep<sup>157</sup>. Numerous studies have demonstrated the therapeutic potential of 5-HT<sub>1A</sub> ligands. For example, the 5-HT<sub>1A</sub> receptor agonists, flesinoxan and buspirone (Figure 5-1), mediate the efficient relief of anxiety<sup>158</sup> and depression<sup>159</sup>. Another agonist, serotonin, has recently been reported to be involved in bone formation<sup>160</sup>. Conversely, evidence supporting the role of 5-HT<sub>1A</sub> receptor antagonists, such as lecozotan (Figure 5-1), in facilitating an enhancement of certain types of learning and memory functions in rodents has led to their current development as novel treatments for Alzheimer's disease<sup>161</sup>.



**Figure 5-1: The structures of some representative 5-HT<sub>1A</sub> ligands**

In addition to elucidating several distinct biological responses induced by 5-HT<sub>1A</sub> agonists and antagonists, many studies have examined their structural requirements. Aganval et al. generated a pharmacophore model based on a small set of 5-HT<sub>1A</sub> agonists and antagonists using comparative molecular field analysis (CoMFA). Their analyses showed that the agonists tended to be “flatter” and more coplanar than the antagonists<sup>116</sup>. Sylte et al., using molecular dynamics simulation of the binding process of 5-HT<sub>1A</sub> ligands, found that the agonists induced larger conformational changes into helix 3 and 6<sup>162</sup>. Han et al. developed a decision tree model to discriminate 5-HT<sub>1A</sub> agonists or antagonists from high-throughput screening data<sup>163</sup>.

Most of the previous studies only focused on a small subset of the 5-HT<sub>1A</sub> ligands, and the predicting power and throughput of calculations were limited. For example, CoMFA requires 3D conformers to be pre-aligned and target compounds to possess identical or similar scaffolds<sup>154</sup>. Moreover, the structures of decision trees are usually developed based on heuristic algorithms, as learning an optimal tree structure is proven to be NP-complete. Therefore, decision trees are substantially influenced by sampled training datasets. Consequently, robust 2D and 3D linear/non-linear predictors were developed for the classification of the agonists and antagonists of the 5-HT<sub>1A</sub> receptor, by data mining 1697 diverse ligands from the GLIDA<sup>126</sup> database. First, the properties of the human 5-HT<sub>1A</sub> agonists and antagonists were analyzed and compared. Subsequently, intrinsic biological functions of the ligands were classified via support vector machine (SVM) algorithm, invented by Vapnik. SVM is a popular classification algorithm in many fields. As the parameter regularization is built into a constraint optimization, it generally tends not to overfit training datasets. In SVM, compound structures were mapped to high dimensional descriptor vectors, called molecular fingerprints. When 2D molecular fingerprints were used as descriptor vectors, polynomial kernel, RBF kernel (Radial Basis Function) and Tanimoto kernel were used to



develop non-linear classifiers. Furthermore, a novel Topomer kernel is proposed to incorporate the concept of 3D fragment similarity. Different from traditional fingerprint representation of compound structures, the Topomer kernel in the SVM does not require an explicit structure-descriptor mapping mechanism. Instead, the Topomer similarity score between any pair of compounds suffices to train a model, which is an application of “kernel trick”. The rationale of these algorithms is supported by the principle that an active compound usually behaves as an agonist if it structurally resembles agonists, and as an antagonist if it structurally resembles antagonists. Test calculations showed that the SVM yielded average 88.3% to 92.3% prediction accuracy with five types of molecular fingerprints and three non-linear kernels (Polynomial, RBF and Tanimoto kernels), and the Topomer kernel produced average 90.9% prediction accuracy.

## 5.2 METHODS, MATERIALS AND CALCULATION

### 5.2.1 Support Vector Machine

The motivation and concepts of Support Vector Machine (SVM) have been brought up in 4.3.2. Continue with the standard setting of linear SVM:

$$\text{Maximize } L_{\alpha} = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \quad (5-1)$$

$$\text{Subject to } \sum_{i=1}^n \alpha_i y_i = 0 \text{ with } 0 \leq \alpha_i \leq C \forall i, i = 1, \dots, n$$

Once we have  $\alpha_i$ , vector  $\mathbf{w}$  can be recovered by equation,  $\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i$ . Sometimes a linear classifier in the original feature space fails to identify the pattern of non-linear observations, e.g. using a linear classifier to learn an XOR function. This problem could be tackled by mapping the non-linear observations to some higher dimensional feature space:  $\mathbf{x} \rightarrow \varphi(\mathbf{x})$ , in which the data are linearly separable. The linear classifier in the higher dimensional space may correspond to a non-linear hypothesis in the original feature space. Nevertheless, it is rather difficult to identify an appropriate mapping function,  $\varphi$ . In equation (5-1), observations  $\mathbf{x}_i$  and  $\mathbf{x}_j$  only appear as inner product,  $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ . After the observations are mapped by function  $\varphi$ , the observations appear as the inner product in the new feature space,  $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \varphi(\mathbf{x}_i), \varphi(\mathbf{x}_j) \rangle$ . Therefore, the mapping function  $\varphi$  does not have to be identified explicitly, as long as the inner product in the higher dimensional space,  $K(\mathbf{x}_i, \mathbf{x}_j)$ , can be solved. This

technique is named as “kernel trick”, and function  $K(\mathbf{x}_i, \mathbf{x}_j)$  is called “kernel” function. This important character enables us to derive a SVM model only based on the relationship between any pair of training samples without knowing their vector representation. In this case, the prediction becomes  $\text{sign}(\sum_{i=1}^n \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_{\text{test}}) + b)$  for a given testing case  $\mathbf{x}_{\text{test}}$ , where  $\mathbf{x}_i$  is the  $i^{\text{th}}$  training sample.

In this study, four types of popular kernel functions were considered. A standard support vector machine employs the linear kernel (eq. 5-2). Polynomial kernel (eq. 5-4) and RBF kernel (eq.5-5) have been applied widely, delivering decent performance in many applications. Thus, the polynomial and RBF kernel were tested along with other kernel functions. The Tanimoto kernel function (eq. 5-6), which has been used intensively in cheminformatics, incorporates prior knowledge on how to calculate chemical similarity.

Besides these four popular kernel functions, we propose a novel kernel function, Topomer kernel, to integrate the concept of 3D similarity into SVM. Sybyl Topomer Search directly generates 3D similarity scores, which can be plugged into the Topomer kernel function (eq. 5-3) for model training. Conversely, it is difficult for non-kernel machines, e.g. naive Bayes classifier, to make use of this information. “A topomer is a specific alignment or pose of a molecular fragment, prescribing both its conformation and position”<sup>164</sup>. In the Topomer similarity algorithm, the molecule is split into two or three fragments and the similarity of two molecules is evaluated by the similarity between the topomers of these fragments. Generally, the Topomer similarity considers both overall steric and pharmacophoric features, which enables it to function as a powerful 3D similarity tool in both virtual screening and QSAR studies<sup>165-166</sup>.

$$K_{\text{linear}}(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle \quad (5-2)$$

$$K_{\text{Topomer}}(\mathbf{x}_i, \mathbf{x}_j) = \left(1 - \frac{\text{Topomer}(\mathbf{x}_i, \mathbf{x}_j)}{\max_{i,j} \text{Topomer}(\mathbf{x}_i, \mathbf{x}_j)}\right)^d \quad (5-3)$$

Note: for consistent notation, assume Topomer distance and Topomer kernel are calculated from virtual vector  $\mathbf{x}_i$ .

$$K_{\text{polynomial}}(\mathbf{x}_i, \mathbf{x}_j) = (\langle \mathbf{x}_i, \mathbf{x}_j \rangle + 1)^d \quad (5-4)$$

$$K_{\text{RBF}}(\mathbf{x}_i, \mathbf{x}_j) = \exp(-g|\mathbf{x}_i - \mathbf{x}_j|^2) \quad (5-5)$$

$$K_{\text{Tanimoto}}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\langle \mathbf{x}_i, \mathbf{x}_j \rangle}{\langle \mathbf{x}_i, \mathbf{x}_i \rangle + \langle \mathbf{x}_j, \mathbf{x}_j \rangle - \langle \mathbf{x}_i, \mathbf{x}_j \rangle} \quad (5-6)$$

## 5.2.2 Materials and Calculations

A collection of 1102 5-HT<sub>1A</sub> agonists and 595 5-HT<sub>1A</sub> antagonists was retrieved from the GLIDA database<sup>125-126</sup>. The whole 5-HT<sub>1A</sub> data set was randomly split into 10 different training and testing sets, with 75% agonists and antagonists representing the training sets and the remaining forming the testing sets. MACCS, Unity, FP2, Molprint 2D and PubChem fingerprints were generated for all the training and testing compounds. MACCS and PubChem fingerprints are dictionary-based fingerprints, each dimension of which indicates the presence or absence of a set of predefined fragments<sup>167</sup>. Unity and FP2 fingerprints encode atom-path patterns with variable length. Molprint 2D fingerprint<sup>168-169</sup> is a circular atom environment fingerprint, which depicts all the neighboring atoms around each central heavy atom. Tripos Sybyl was used to calculate the Topomer distance,  $\text{Topomer}(\mathbf{x}_i, \mathbf{x}_j)$ , between any pair of compounds.

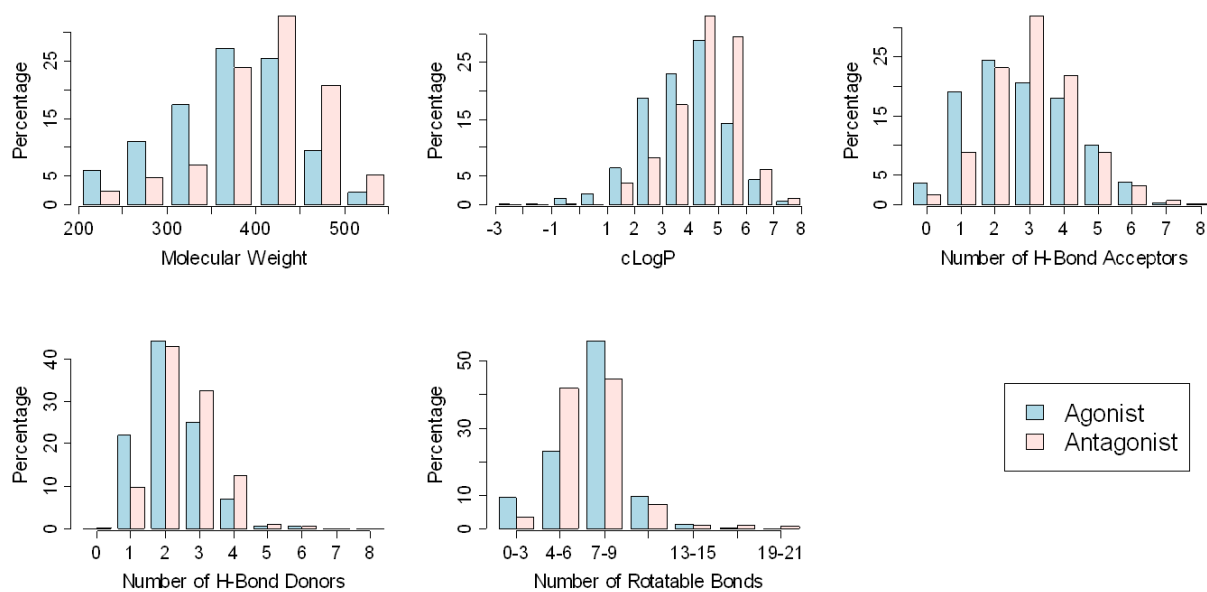
The performance of linear, polynomial, RBF and Tanimoto kernel functions was measured together with five types of fingerprints, resulting in 20 different models (descriptor  $\times$  kernel). Another SVM model was only specified by Topomer kernel, as Topomer distance was a special descriptor. For a given SVM model, 10 rounds of training were carried out by 10 different training sets, and the prediction accuracy was evaluated through the corresponding testing data sets. Predictive SVM model creation and evaluation were carried out with the svmpath software package<sup>170</sup>. The svmpath implementation discovers the entire path of the SVM solution and provides an interface for user-defined kernel functions. The regularization and kernel parameters were determined by 10-fold cross-validation.

## 5.3 RESULTS AND DISCUSSION

The focus of this section is on the classification of human 5-HT<sub>1A</sub> ligands. Firstly, we demonstrate that some general molecular properties, such as properties mentioned in Lipinski's Rule-of-Five and the number of rotatable bonds, exhibit limited discrimination power, as expected. Interestingly, the significant difference between intra-class and inter-class similarity measured by molecular fingerprints suggests the possibility of building a predictive agonist-antagonist classifier using these fingerprints. Next, the performance of different combinations of kernels and fingerprints is retrospectively investigated and analyzed through cross-validation.

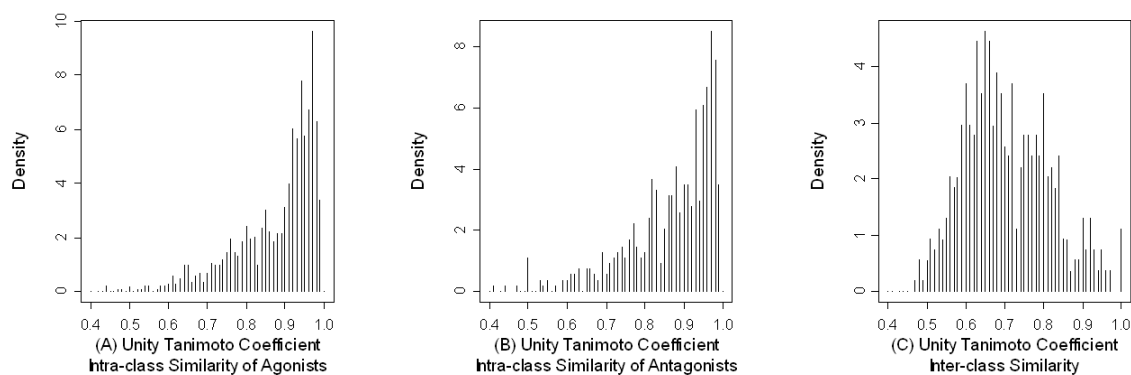
### 5.3.1 Molecular Properties of Agonists and Antagonists

The distribution of five molecular descriptors of 5-HT<sub>1A</sub> agonists and antagonists is plotted in Figure 5-2, including the properties mentioned in Lipinski's Rule-of-Five and the number of rotatable bonds. The average molecular weights for agonists and antagonists are 373.75 and 414.05 Dalton, respectively. Accordingly, agonists possess on average 2.78 H-bond acceptors, while antagonists have on average 3.07 H-bond acceptors. The analysis of rotatable bonds reveals that antagonists are relatively more rigid compared to agonists. The results are congruent with the TOGGLE model provided by Schwartz et al<sup>171</sup>. The flexibility of agonists is suggested through induction of a common molecular activation mechanism involving TM-V, TM-VI and TM-VII of the GPCR receptor. A recent publication on the human A<sub>2A</sub> adenosine receptor validates this hypothesis<sup>172</sup>. According to the TOGGLE model, agonists should be more flexible in order to adjust their conformation with the receptor's conformational changes. Antagonists, however, are rigid and can hinder the movement of the receptor helixes upon binding<sup>162</sup>. The subtle differences in these descriptors can hardly distinguish agonists from antagonists.



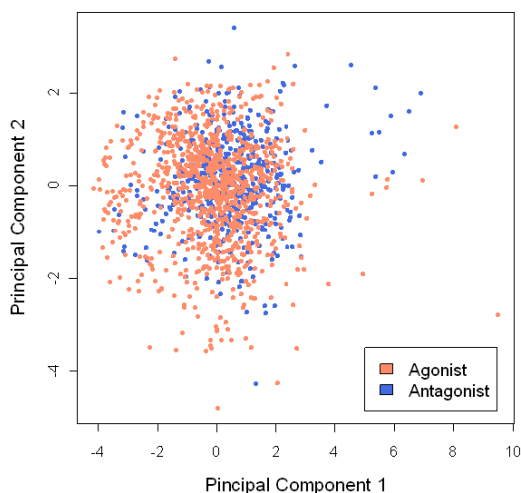
**Figure 5-2: The distribution of molecular properties 5-HT<sub>1A</sub> agonists and antagonists**

The properties include molecular weight, cLogP, the number of H-bond acceptors, the number of H-bond donors and the number of rotatable bonds.



**Figure 5-3: Intra-class and inter-class similarity of 5-HT<sub>1A</sub> agonists and antagonists.**

Principal component analysis is a popular technique for dimension reduction and visualization. The principal components are linear combinations of original features and capture the maximum variance of high-dimensional samples. N-dimensional data usually produces N components, the variance of which is in descending order. Figure 5-4 plots the first two principal components of the reported five molecular properties. Agonists and antagonists are colored differently in this transformed chemistry space. The standard deviations of the first two principal components are 1.52 and 1.10, and the standard deviations of omitted components are 0.89, 0.70, and 0.44. Agonists and antagonists have significant amounts of overlap in the new coordinate system, suggesting that performing data mining on these traditional molecular properties may only yield limited accuracy.



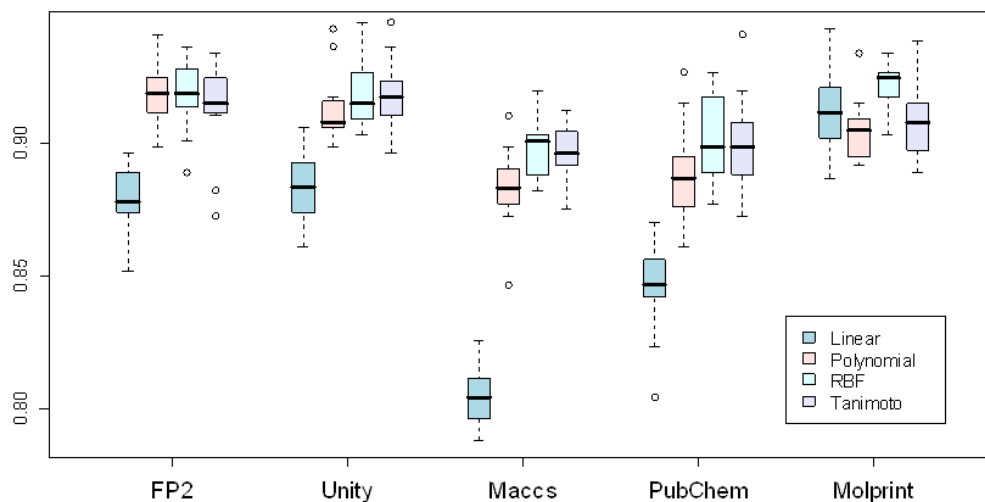
**Figure 5-4: Principal component analysis of five molecular descriptors.**

The scatter plot of the first two components from molecular weight, cLogP, number of H-bond donors, acceptors and rotatable bonds.

Even though agonists share certain similarity with antagonists, the examination of intra-class and inter-class similarities using compound library comparison<sup>173</sup>. Figure 5-3 suggests that they retain distinct molecular fragments (The distribution was generated by Tripos Sybyl Selector. Intra-class similarity is defined as the pairwise Tanimoto coefficient calculation within agonist or antagonist compound collection; inter-class similarity refers to the pairwise compound comparison between the agonist and antagonist datasets.). The average intra-class Tanimoto similarity of agonists and antagonists are 0.88 and 0.87, with a standard deviation of 0.11 and 0.12, respectively. On the other hand, the average inter-class Tanimoto score is 0.71 with a standard deviation of 0.11. The difference between intra- and inter-class similarities suggests the possibility of building a predictive agonist-antagonist classifier by data mining the structural patterns that define such similarity.

### 5.3.2 Classification Performance Using Fingerprints

The results of systematic comparisons of different molecular fingerprints and kernel functions are summarized in Figure 5-5 and Table 5-1. The average prediction accuracy of SVM models ranges from 80.4% to 92.3%. As shown in Figure 5-5, non-linear classifiers that are derived from polynomial, RBF and Tanimoto kernel functions generally outperform linear classifiers.



**Figure 5-5: The performance of kernel and fingerprint**

The boxplot showing the distribution of prediction accuracy with combinations of molecular fingerprints and kernel functions.

**Table 5-1: Average prediction accuracy of fingerprint/kernel combinations.**

	Linear	Polynomial	RBF	Tanimoto
FP2	0.878	0.919	0.919	0.912
Unity	0.883	0.913	0.919	0.918
MACCS	0.804	0.883	0.897	0.896
PubChem	0.846	0.888	0.903	0.900
Molprint 2D	0.912	0.905	0.923	0.909

The results are calculated as the ratio between correctly predicted compounds and the total number of testing compounds out of ten rounds of validation.

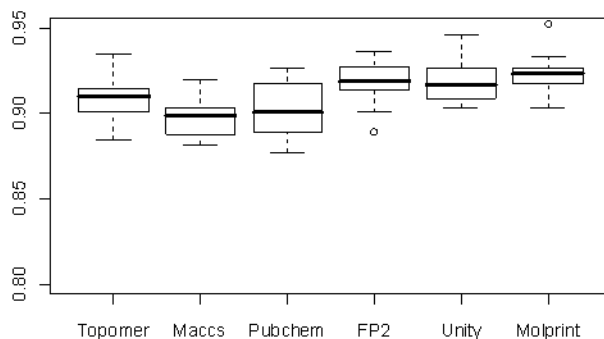
Out of 10 rounds of test calculations using the MACCS fingerprint as descriptor, the highest prediction accuracy of linear SVM models, which could be regarded as default approach for SVM, is still lower than the worst performance of any non-linear SVM models (Figure 5-5). The average performance of FP2, Unity and PubChem fingerprints with linear kernel is 87.8%, 88.3% and 84.6% respectively, which are approximately  $\Delta$  3% - 6% lower compared to non-linear kernels (Table 5-1). This outcome suggests that compound fragments may not make straightforward additive contributions to ligand-receptor interactions, i.e. considering the simultaneous influence of two or more functional groups could improve the quality of models. In this case, a non-linear model is a possible solution to capture the interaction among fragments. Although non-linear kernels outperform standard linear functions for these four types of fingerprints, Figure 5-5 shows that the Molprint 2D fingerprint consistently yields rigorous predictions regardless of kernel functions, with an average accuracy ranging from 90.5% to 92.3% (Table 5-1). This could be explained by the high sparsity and high dimensionality of the Molprint 2D fingerprint. In Molprint 2D, each heavy central atom and its surrounding heavy atoms uniquely define a pattern. In this study, the entire agonist and antagonist compound collection generates 6839 Molprint 2D patterns, while the lengths of FP2, Unity and MACCS fingerprints are 1024, 992 and 166, respectively. Out of 6839 Molprint 2D features, the number of features that a compound possesses is actually equal to the number of its heavy atoms, resulting in high sparsity. The purpose of kernel functions is to map original features to a higher, possibly infinite dimensional space so that a better linear classifier could be obtained in the new space. As Molprint 2D has already defined numerous features, a linear classifier is adequate to solve the problem and a solution in another high dimensional space does not necessarily lead to a significant improvement.

Among non-linear kernels, RBF kernel and Tanimoto kernel are generally superior to the polynomial kernel although their performance is quite similar. Furthermore, Figure 5-5 and Table 5-1 show that the

RBF kernel outmatches the other three kernels no matter which type of fingerprint is used. Molprint 2D is the most favorable fingerprint because of its rich information, consistency and SVM kernel independency. Following Molprint 2D, FP2 and Unity fingerprints produce similar results, roughly 88% accuracy for linear kernel and 91% - 92% for non-linear kernels. PubChem and MACCS fingerprints could be ranked as last, given their relatively lower performance compared to FP2 and Unity (Table 5-1). As previously mentioned, Molprint 2D, FP2 and Unity fingerprints specify a set of rules for generating structural patterns, instead of predefining structural fragments or patterns. Thus, the interpretation of these fingerprints depends on the presented compound collection. On the other hand, PubChem and MACCS fingerprint preset a look-up table or structure dictionary. In this case, primal functional groups pertaining to distinct pharmacological or physico-chemical properties may not be readily defined in MACCS and PubChem fingerprints, but they may be well traced in FP2, Unity and Molprint 2D descriptors. To summarize, the best 2D SVM model is derived from the combination of Molprint 2D fingerprint and a RBF (or Gaussian) kernel, which exhibits 92.3% average accuracy on ten testing datasets. This conclusion is also consistent with Wasserman et al.'s findings<sup>174</sup>.

### 5.3.3 Topomer Distance Kernel

As previously mentioned, Topomer distance reported by Sybyl is an efficient 3D similarity metric and somewhat better than the traditional approach of 2D Tanimoto similarity<sup>175</sup>. In this section, Topomer kernel combined with Topomer distance is compared to the top-performing kernel, Radial Basis kernel, and 2D fingerprints, and the results are plotted in Figure 5-6.



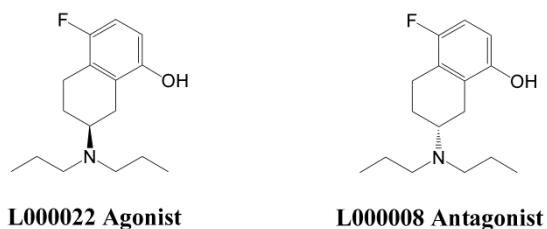
**Figure 5-6: The comparison of Topomer kernel and RBF kernel**

The boxplot shows SVM prediction accuracy with Topomer distance and Topomer kernel, compared with traditional 2D fingerprints and RBF kernel.



Overall, using Topomer distance in SVM shows decent classification power with an average 90.9% accuracy. This is slightly better than the RBF kernel with MACCS and PubChem fingerprints. Nevertheless, this approach does not necessarily outperform the RBF kernel with other fingerprints, since the “gap” is statistically insignificant. In many virtual screening processes, 2D fingerprints perform better than 3D fingerprints or pharmacophore features<sup>176</sup>. The conclusion from our study is not an exception. Heuristically, 3D descriptors and 3D similarity metrics are supposed to be superior to traditional 2D approaches, especially when a query compound has a different scaffold from currently known compounds. Other work suggests that 3D descriptors can provide competitive similarity searching results in some scaffold hopping-oriented virtual screening<sup>177</sup>.

Consider an extreme case shown in Figure 5-7. Both compounds (L000022 and L000008) cannot be correctly classified by any 2D approaches in this study, as they have identical 2D fingerprints. Their Topomer distance is 0.26, indicating remarkable pairwise similarity. Generally, compounds with Topomer distance below 185 are regarded as similar<sup>165-166</sup>, and identical compounds have zero Topomer distance. Unfortunately, the SVM model failed to identify L000008 as an antagonist using Topomer distance, when both compounds were present in the testing dataset. Recently, Guha et al. developed an index to detect the molecules that are very similar but have a large difference in activity. This phenomenon is named as “activity cliff”<sup>178</sup>. In our case, the functionality of L000022 and L000008 against the 5-HT<sub>1A</sub> receptor can be considered as another cliff, i.e. a functionality cliff. The functionality cliffs may provide some insights into the relationship between structure and functionality of GPCR ligands.

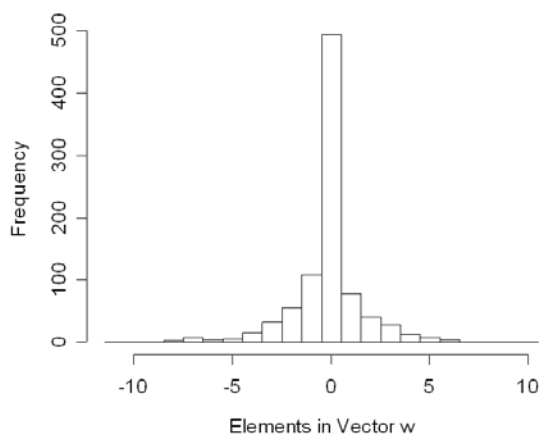


**Figure 5-7: Two stereoisomers that have distinct biological functionality**

### 5.3.4 Model Interpretation

In this study, linear classifiers and non-linear classifiers have been developed through SVM training algorithm. Linear classifiers make classifications based on a linear combination of structural fragments or features. As mentioned in the Methods section, the prediction of a testing sample is formulated as  $f_{w,b}(\mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x} + b)$  in vector notation, or  $f_{w,b}(\mathbf{x}) = \text{sign}(\sum w_i x_i + b)$ , where  $x_i \in \{0, 1\}$  is one dimension of molecular fingerprints and represents a specific structural pattern. Note that  $\mathbf{x}$  is descriptor vector;  $\mathbf{x}_i$  represents the descriptor of the  $i^{\text{th}}$  compound, and  $x_i$  is the  $i^{\text{th}}$  element of vector  $\mathbf{x}$ .

Accordingly, the corresponding coefficient  $w_i$  determines the contribution of the pattern to the final prediction. The pattern  $i$  that has a large associated  $|w_i|$  has more influence on the prediction than the patterns with smaller  $|w_i|$ . In other words, examining heavily weighted structural patterns may highlight the substructures associated with ligand functionality. The optimization of equation (1) yields a set of coefficients for support vectors, i.e.  $\alpha$ , and  $\mathbf{w}$  can be solved by equation  $\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i$ . In this section, one example is given to illustrate a linear SVM model that is developed on the whole agonist/antagonist dataset using PubChem fingerprint as descriptor.



**Figure 5-8: Histogram displaying the distribution of elements in vector  $w$ .**

Figure 5-8 shows the distribution of elements in vector  $\mathbf{w}$ , i.e.  $w_i$ . As shown, the majority of  $w_i$  reside in interval  $(-0.5, 0.5)$ , suggesting that most of predefined PubChem features are possibly irrelevant to ligand

functionality. Only 26 patterns possess an absolute weight larger than 5, some of which are shown in Table 5-2. Besides certain substructures, distinct ring systems are observed in agonists and antagonists. According to the model, agonists prefer saturated or heteroatom aromatic rings. In contrast, simple aromatic rings are frequently observed in antagonists. Thus, extra caution should be taken to correlate these features with chemical modifications in order to achieve the desired biological profile, which was pointed out by Wassermann et al<sup>174</sup>. The support vector machine aims at minimizing generalization error, not causal inference.

**Table 5-2: List of structural patterns emphasized by a linear SVM classifier.**

Index	Fingerprint Annotation <sup>a</sup>	Favored by	Weight <sup>b</sup>
187	>= 2 saturated or aromatic nitrogen-containing 6-member rings	Agonist	5.88
194	>= 3 saturated or aromatic nitrogen-containing 6-member rings	Agonist	6.41
650	O=C-N=C=O	Agonist	8.22
698	O=C-C-C-C-C-C-C-C	Agonist	9.12
860	CC1C(C)CCC1	Agonist	5.81
257	>= 2 aromatic rings	Antagonist	-9.40
261	>= 4 aromatic rings	Antagonist	-8.34
597	O=C-C-C-C	Antagonist	-11.44
647	O=C-N-C-N	Antagonist	-7.66
674	N-C-N-C	Antagonist	-9.63

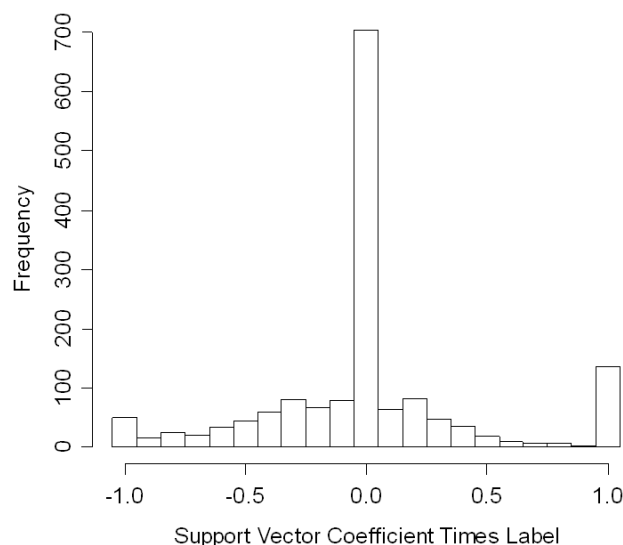
<sup>a</sup> The implementation of PubChem fingerprint comes from the CDK software package, so the fingerprint annotation may not completely conform to PubChem standards. The substructures are given in SMILES or SMARTS notation. "=" means double bond, "-" means single bond, ":" means aromatic bond, and "=:" means either double bond or aromatic bond.

<sup>b</sup> In this study, agonists are labelled as +1, while antagonists are labelled as -1. Thus, the presence of a pattern with positive weight "votes" for agonists or antagonists otherwise.

<sup>c</sup> The index is zero-based, and it corresponds to bit position in the PubChem fingerprint. The fingerprint annotation is the description of each bit represented in the fingerprint. The last two columns show which group favors the patterns and their weights ( $w_i$ ) in the prediction.

An interpretation of a non-linear classifier is more complicated, as the mapping mechanism original feature space  $\mathbf{x}$ ,  $\mathbf{x} \in \{0, 1\}^d$ , to Hilbert space  $\phi(\mathbf{x})$  is implicit. Therefore, it is impossible to explain the

model in the “feature level”. In the SVM methodology, the optimal decision boundary is only related to a few training observations, called support vectors. The presence or absence of other “non-support” vectors does not affect the model development. In the non-linear case, the prediction of a testing sample is given by the equation  $\text{sign}(\sum_{i=1}^n \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_{\text{test}}) + b)$ , which consists of two components, support vector coefficients  $\alpha$  and “similarity score” between training and testing compounds. The final prediction generally is contributed by training compound  $i$  that shows a high similarity to the testing compound ( $K(\mathbf{x}_i, \mathbf{x}_{\text{test}})$ ) and is heavily weighted by the support vector coefficient  $\alpha_i$ . As the decision boundary is determined by support vectors, we can examine the training compounds associated with a large value of  $\alpha_i$ . The SVM model developed on all labeled compounds with Topomer kernel is illustrated. The distribution of  $\alpha_i y_i$  is characterized in Figure 5-9.



**Figure 5-9: The distribution of support vector coefficient**

The X-axis shows the product of support vector coefficient  $\alpha_i$  and compound label  $y_i$ .

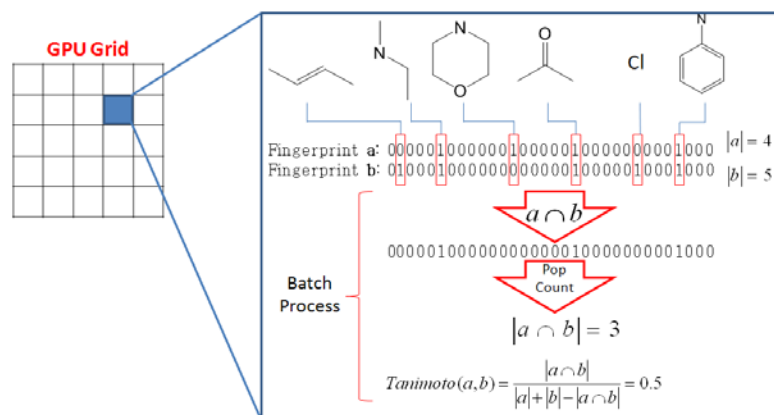
Similar to the appearance of Figure 5-8, the majority of training samples are not support vectors, except that the distribution of support vector coefficients has “thick tails” due to optimization constraint. The number of support vectors ( $\alpha_i > 0.1$ ) is 822, which is 48.4% of the whole training set. Mattera and Haykin<sup>179</sup> propose that a robust model may accommodate the condition that 50% of training data are

support vectors. Empirically, a large percentage of support vectors suggest overfitting the training data, while a lower percentage of support vectors may indicate lack-of-fit.

## 5.4 CONCLUSION

This chapter reports the investigation of linear and non-linear support vector machine (SVM) classifiers with the intention to distinguish agonists from antagonists of the human 5-HT<sub>1A</sub> receptor. The choice of molecular descriptors and kernel functions in SVM has been thoroughly discussed and analyzed. The test calculation shows that the Molprint 2D fingerprint, combined with RBF kernel function, yields the best prediction accuracy. Another innovation of the presented work is the integration of a 3D Topomer similarity distance into SVM through the proposed Topomer kernel function, without explicit vector representation of compounds. The mechanism of linear and non-linear SVM classifiers is illustrated by examining coefficient vector  $\mathbf{w}$  and support vector coefficient vector  $\mathbf{a}$ , respectively. Although only the classification of 5-HT<sub>1A</sub> agonists and antagonists is presented here, linear and non-linear SVM, as a generic classification tool, may be applied to address other challenges in computer-aided drug design.

## 6 GPU-ACCELERATED COMPOUND LIBRARY COMPARISON



Chemical similarity calculation plays an important role in compound library design, virtual screening, and “lead” optimization. This chapter presents a GPU-accelerated method for all-vs-all Tanimoto matrix calculation and nearest neighbor search. By taking advantage of multi-core GPU architecture and CUDA parallel programming technology, the algorithm is up to 39 times superior to the existing commercial software that runs on CPUs. Because of the utilization of intrinsic GPU instructions, this approach is nearly 10 times faster than existing GPU-accelerated sparse vector algorithm, when Unity fingerprints are used for Tanimoto calculation. The GPU program that implements this new method takes about 20 minutes to complete the calculation of Tanimoto coefficients between 32M PubChem compounds and 10K Active Probes compounds, *i.e.*, 324G Tanimoto coefficients, on a 128-CUDA-core GPU.

## 6.1 INTRODUCTION

Combinatorial chemistry generates a large number of compounds and boosts the growth of various screening libraries. Thus, analysis and data mining of the vast quantity of compounds significantly contribute to the success of both virtual screening and high-throughput screening.<sup>180</sup> Among the analysis schemes, chemical similarity calculation is a frequently used method in computer-aided drug design.<sup>181</sup> The concept of chemical similarity or molecular similarity is also heavily involved in molecular diversity analysis and combinatorial library design. Many methods have been established for this purpose, covering a wide range of molecular descriptors and data mining algorithms.<sup>66, 182-187</sup>

Various cheminformatics algorithms have been developed for chemical similarity measurement. The Tanimoto coefficient between molecular fingerprints is still the most popular similarity metric, because of its computational efficiency and its relevance to biological profile<sup>188-189</sup>. In addition, Database Comparison program in Tripos Sybyl<sup>98</sup> uses Tanimoto coefficient to characterize the degree of similarity or overlapping between two compound libraries.<sup>190</sup> Furthermore, Tanimoto calculation is also associated with modern machine learning algorithms, ranging from supervised kernel machine<sup>191</sup> to unsupervised compound clustering.<sup>192-193</sup> In these applications, an all-vs-all Tanimoto matrix, which contains Tanimoto coefficients between all pairs of compounds, needs to be calculated. This results in  $O(N^2)$  time complexity, *i.e.*, quadratic in the size of libraries. Despite the advance in computer hardware, exploring large chemical libraries, such as PubChem library, remains a substantial challenge.<sup>194-195</sup>

With the emphasis on “green high performance computing”, the development of modern graphics processing units (GPU) points out potential solutions to these challenges. GPUs are specialized microprocessors for graphic rendering. Modern GPUs feature higher memory bandwidth and computing throughput in terms of floating point operations per second (FLOPS), compared to CPUs. Recently, GPUs have been applied to quantum chemistry and molecular dynamics,<sup>196-197</sup> as well as Tanimoto calculation.<sup>198-200</sup> Haque *et al.*<sup>198</sup> and Liao *et al.*<sup>199</sup> reported sparse vector algorithm for all-vs-all Tanimoto matrix calculation on GPUs. The established sparse vector algorithm is proficient to process high-sparsity fingerprints.

This chapter introduces a new algorithm to calculate Tanimoto coefficients between pairs of compounds, and to perform compound library comparison on GPUs. Different from Haque’s and Liao’s work, this algorithm achieves better performance when processing low-sparsity molecular fingerprints. Furthermore, compound library comparison can be executed on GPUs after Tanimoto coefficients are solved. In the present studies, we start with algorithm design, and then compare its performance with existing GPU and CPU algorithms using three different chemical libraries and three CPU or GPU systems. The results show that the program that implements this novel approach runs up to 39 times faster than the

Sybyl Database Comparison program and nearly 10 times faster than the existing GPU-based sparse vector algorithm. Furthermore, the program completes the calculation of 324G Tanimoto coefficients in 20 minutes, for comparing 10K Active Probes compounds with PubChem library. The algorithm is implemented with graphical user interface (GUI) for ease of use. The binary, source code and user instruction are available at <http://www.cbligand.org/gpu>. The program can be customized to adapt any binary fingerprints and carry out kNN (k-nearest neighbor) search.

## 6.2 METHODS AND CALCULATIONS

### 6.2.1 Overview of Compound Library Comparison

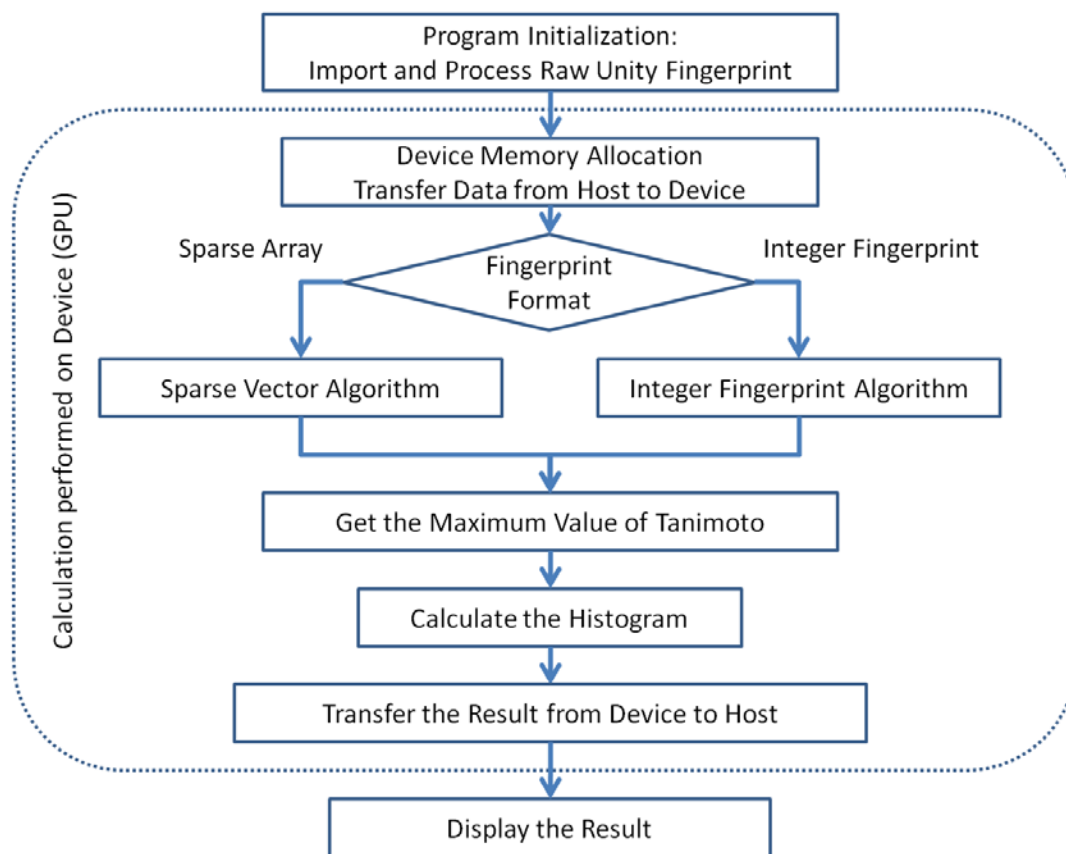
The concepts and details regarding the GPU-accelerated compound library comparison are described in this section. In cheminformatics, Tanimoto coefficient is one of the most popular chemical similarity indices, and is usually calculated based on binary molecular fingerprints. The Tanimoto coefficient between a pair of compounds that have molecular fingerprints **a** and **b** can be formulated as:

$$\text{Tanimoto}(a, b) = \frac{N_{ab}}{(N_a + N_b) - N_{ab}}$$

where  $N_{ab}$  is number of common “1” bits that occur in both fingerprint **a** and fingerprint **b**;  $N_a$  is number of “1” bits in fingerprint **a**;  $N_b$  is number of “1” bits in fingerprint **b**. In our calculation, Tanimoto coefficient is computed with Unity<sup>98</sup> fingerprint as illustration, whereas other fingerprints can be applied as well. Unity fingerprint is a 992-bit binary vector, which encodes the presence or absence of a set of predefined structural patterns.

The compound library to be compared to is named as reference library, while the other compound library is named as candidate library. The goal is to characterize how well the candidate library is represented in the reference library. Among established programs, Tripos Sybyl<sup>190</sup> examines the Tanimoto similarity between each candidate compound and its nearest neighboring compound (most similar compound) in a reference library. Let **X** denote an  $m \times n$  Tanimoto matrix, in which  $X_{i,j}$  is the Tanimoto coefficient between the  $i^{\text{th}}$  compound in candidate library and the  $j^{\text{th}}$  compound in reference library. Finally, the distribution of **y** indicates the degree of overlapping between the two libraries, where **y** is an  $m$ -element vector and  $y_i = \max_{1 \leq j \leq n} X_{i,j}$  for  $1 \leq i \leq m$ .





**Figure 6-1: Flowchart of similarity calculation on GPU**

The general workflow of compound library comparison on the computer hardware where a GPU is located

Figure 6-1 shows the general diagram for compound library comparison on GPUs. First, Unity fingerprints are indexed and transferred to allocated graphical memory. Next, depending on the format of the indexed fingerprints, “Sparse Vector” kernel or “Integer Fingerprint” kernel is launched on a GPU to calculate the Tanimoto matrix,  $X$ . After the Tanimoto matrix calculation, a parallel reduction kernel is executed to identify the maximum of each row of  $X$ , *i.e.*  $y_i$ . Finally, the distribution of  $y_i$  is examined through computing and displaying its histogram.

## 6.2.2 Integer Fingerprint Algorithm on GPUs

Tanimoto equation consists of three elements:  $N_a$ ,  $N_b$  and  $N_{ab}$ . In library comparison, the number of “1” bits of each compound,  $N_a$  or  $N_b$ , is used repeatedly. Thus, it is wise to pre-calculate  $N_a$  and  $N_b$  in the fingerprint indexing procedure. In this algorithm, binary molecular fingerprints, such as Unity fingerprint, are saved into 32-bit integers. This approach reduces time and space complexity for low-sparsity fingerprints. First, one integer is capable of representing 32 fingerprint bits, so only 124 bytes are required to save the whole Unity fingerprint of any compound (Unity fingerprint has 992 bits). Furthermore, the throughput of calculating  $N_{ab}$  can be guaranteed by efficient intrinsic “&” operator and population count (pop-count) instructions on GPUs. The “&” operator finds the common “1” bits between two 32-bit fingerprint fragments ( $A \cap B$ ), while the pop-count instruction returns the number of the common “1” bits. Therefore, the total number of common “1” bits between two fingerprints, *i.e.*  $N_{ab}$ , can be obtained through applying “&” and pop-count instructions on every 32-bit fingerprint fragment.

In GPU parallel programming, the  $i^{\text{th}}$  thread block (or virtual GPU core) is responsible for the  $i^{\text{th}}$  row of the Tanimoto matrix,  $X_{i,\bullet}$ , and every thread in the block calculates one or more elements of  $X_{i,\bullet}$ . Each thread block therefore compares a candidate compound to the whole reference library, and every thread in the block calculates the Tanimoto coefficients between the candidate compound and some reference compounds. For coalesced memory access, the reference and candidate library fingerprints are organized in column and row major 2D arrays, respectively. Figure 6-2A summarizes algorithm pseudo-code for a given thread block, and Figure 6-2B graphically illustrates the calculation procedure.

**Data:**  $N_a$  : the number of “1” bits of the  $i^{\text{th}}$  candidate fingerprint; vector  $\mathbf{a}$  contains the fingerprint fragments of the  $i^{\text{th}}$  candidate compound;  $N_b^j$  : the number of “1” bits of the  $j^{\text{th}}$  reference fingerprint; vector  $\mathbf{b}_j$  contains the fingerprint fragments of the  $j^{\text{th}}$  reference compound;  $M$ : the total number of compounds in the reference library; Dim: the number of fingerprint fragments or dimensions. Totally  $T$  threads are launched in a thread block ( $T = 256$  in our implementation).

**Begin**

$N_b^j$  and  $\mathbf{b}_j$  reside in global memory. Fetch  $N_a$  from constant memory to thread register; Fetch vector  $\mathbf{a}$  from global memory to low-latency shared memory.

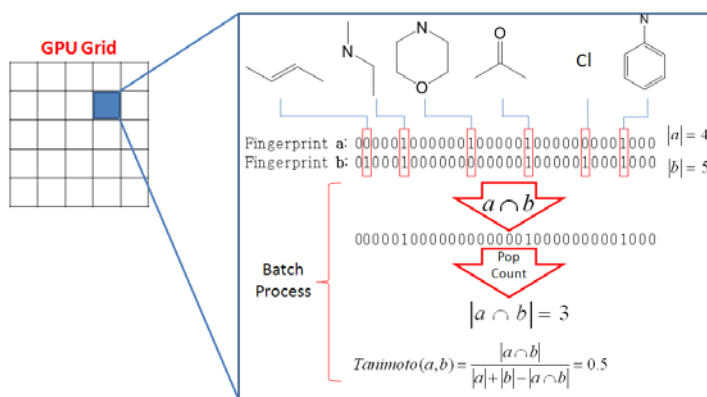
```

for each thread with index t
  for batch = 0 to ceil(M / T) - 1
    j = t + batch × T
    count = 0;
    for k = 0 to Dim - 1
      tmp =  $\mathbf{a}[k] \& \mathbf{b}_j[k]$ 
      count += popc(tmp) ‘intrinsic population count function
    end k
    Tanimoto( $\mathbf{a}, \mathbf{b}_j$ ) = count / ( $N_a + N_b^j - \text{count}$ )
  end batch
end thread t

```

**End**

(A)



(B)

**Figure 6-2: Similarity calculation based on dense-format fingerprint**

(A) The pseudo-code for a GPU thread block to calculate the Tanimoto coefficients between a candidate compound and a reference library using integer fingerprint format; (B) Graphical illustration of Tanimoto calculation on a CUDA core. The GPU algorithm is designed to count the number of common elements between two 32-bit fingerprint fragments in one single step.

(B)

### 6.2.3 Sparse Vector Algorithm on GPUs

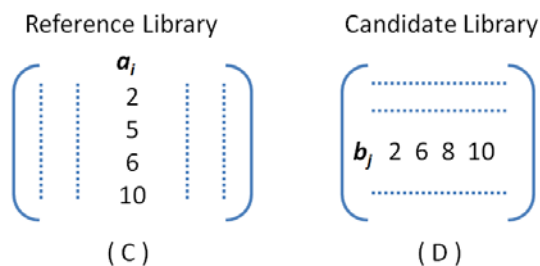
The calculation strategy of sparse vector algorithm is similar to the one of integer fingerprint algorithm previously mentioned: the  $i^{\text{th}}$  thread block (or one virtual core) is responsible for the  $i^{\text{th}}$  row of the Tanimoto matrix, and every thread in the block calculates one or more elements of that row. On the other hand, the sparse vector algorithm adapts an alternative approach to calculate Tanimoto coefficients between pairs of fingerprints, as reported by Haque *et al.* and Liao *et al.*<sup>198-199</sup>

$$\begin{array}{ll} i^{\text{th}} \text{ reference compound} & \mathbf{a}_i = [0\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 1] \\ j^{\text{th}} \text{ candidate compound} & \mathbf{b}_j = [0\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 1] \end{array}$$

(A)

$$\begin{array}{ll} i^{\text{th}} \text{ reference compound} & \mathbf{a}_i = [2, 5, 6, 10] \\ j^{\text{th}} \text{ candidate compound} & \mathbf{b}_j = [2, 6, 8, 10] \end{array}$$

(B)



**Figure 6-3: Illustration of data structure of sparse vectors.**

(A) original binary fingerprint format; (B) sparse vector format; (C) column major alignment for reference library; (D) row major alignment for candidate library

Figure 6-3 illustrates the representation of fingerprints in sparse vector format. Figure 6-3A displays the fingerprints of the  $i^{\text{th}}$  reference and  $j^{\text{th}}$  candidate compounds in original binary format. Instead of using the binary format, the fingerprints can be indexed into sparse vector format as well. As shown in Figure 6-3B, each element indicates the index, of which the bit is “1” in the binary format, and the indices are sorted in an increasing order. In Tanimoto matrix calculation, the fingerprints from reference library and candidate library are organized in a column-major (Figure 6-3C) and row-major (Figure 6-3D) layout in order to achieve coalesced memory access. Sparse vectors of a compound library are sorted according to

vector length with the intention of minimizing the possibility of divergent execution branches. For a specific pair of reference and candidate compounds, Figure 6-4 outlines the algorithm to calculate the Tanimoto coefficient between them.

**Data:** vector  $\mathbf{a}$  contains the sparse-vector fingerprint of a candidate compound,  $\mathbf{a}$  resides in GPU's shared memory;  $N_a$  : the length of vector  $\mathbf{a}$ ; vector  $\mathbf{b}$  contains the sparse-vector fingerprint of a reference compound;  $N_b$  : the length of vector  $\mathbf{b}$ ;

**Begin**

```

count = 0;
i = j = 0;
while i < Nb and j < Na
    buffer =  $\mathbf{b}$  [i]    'retrieve an element from global memory to thread register
    while j < Na and  $\mathbf{a}$ [j] < buffer
        j++
    end while
    if  $\mathbf{a}$ [j] == buffer
        count++
    end if
    i++
end while
Tanimoto( $\mathbf{a}$ ,  $\mathbf{b}$ ) = count / (Na + Nb - count)

```

**End**

**Figure 6-4: Tanimoto coefficient and sparse vector fingerprint**

The pseudo-code for calculating the Tanimoto coefficient between a specific pair of compounds that are represented in sparse vector fingerprints

#### 6.2.4 *Find Maximum Tanimoto and Create Histogram on GPUs*

A parallel divide-and-conquer algorithm is used to find the maximum values of Tanimoto coefficients. Every thread block is in charge of searching the maximum value from one row of the Tanimoto matrix. In

a block, threads independently find the local maximum values from disjoint sets of Tanimoto coefficients. Then, parallel reduction step determines the global maximum ( $y_i$ ) of each row.

Despite its simple idea, histogram creation is a non-trivial job on GPUs, because GPUs have a limited amount of memory allowing low-latency random access. In this case, GPU threads maintain conflict-free counters to create sub-histograms in parallel. Finally, a 100-bin histogram is generated by merging all the sub-histograms. This histogram displays the distribution of  $y_i$  and depicts the degree of similarity between two compound libraries.

## 6.2.5 Computation Protocols

The computing performance of the established GPU-accelerated algorithms was evaluated through comparing three compound libraries against each other. For the three libraries, 992-bit Unity fingerprints were generated by Tripos Sybyl software and formatted into the data structures required by sparse vector or integer fingerprint algorithms.

**Table 6-1: The number of compounds and coverage statistics for three testing compound libraries.**

Library ID	No. of Molecules	Lowest Coverage (%)	Average Coverage (%)	Highest Coverage (%)
A	9902	5.2	19.8	43.2
B	24755	5.3	18.7	44.4
C	56079	2.4	19.1	41.8

Coverage ratio is defined as the percentage of “1” bits in the 992-bit Unity fingerprint. Library A was generated from TimTec 10K-Active-Probes library (<http://www.timtec.net/actiprobe-10k.html>); Library B was generated from TimTec 25K-Active-Probes library (<http://www.timtec.net/actiprobe-25k.html>); Library C was generated from Maybridge Screening Collection.

Details regarding the testing compound libraries can be found in Figure 6-1, including the TimTec libraries (Library A and B) and Maybridge Screening Collection (Library C). Sometimes, a data entry in SDF files may contain two or more separate structures, *e.g.*, compound and organic solvent, and such entries have been removed from these libraries. The coverage statistics in Table 6-1 reflects the sparsity of Unity fingerprints, when applied on the sample libraries.

The GPU integer fingerprint and sparse vector algorithms were implemented in our CudaCLA program. The executable file, together with source code and documents, is available at <http://www.cbligand.org/gpu>. The program was compiled by Visual Studio C++ 2005 and CUDA 3.0 toolkit, and implemented with graphical user interface. For comparison studies, we selected three computer systems, including:

- Machine 1: CPU: Intel Xeon 5160 at 3.0 GHz
- Machine 2: CPU: AMD Athlon 3800+ at 2.0 GHz; GPU NVIDIA Quadro FX 580 (32 CUDA cores at 1.12 GHz); NVIDIA drivers: 197.13
- Machine 3: CPU: Intel Core i7 860 at 2.8 GHz; GPU NVIDIA Geforce GTS 250 (128 CUDA cores at 1.78 GHz); NVIDIA drivers: 197.13

The performance of our GPU program was compared to the commercial Sybyl Database Comparison<sup>190</sup> program. Sybyl Database Comparison program, as a CPU program, was tested on the three machines, while the performance of integer fingerprint and sparse vector algorithms were tested on GPUs of machine 2 and 3. It is worth pointing out that the Sybyl Database Comparison program is also based on Unity fingerprints.

### 6.3 RESULTS AND DISCUSSION

The GPU-accelerated compound library comparison is developed for the first time using integer fingerprint algorithm, and its performance is summarized in Table 6-2 and Table 6-3. For comparisons, the performance of sparse vector algorithm is listed in Table 6-4 and Table 6-5. In the tables, the calculation time includes the time spent on graphical memory allocation, data transfer, Tanimoto calculation using either algorithm, searching for the maximum values, and histogram creation (the procedures enclosed by dotted line in Figure 6-1). Tanimoto matrix calculation generally accounts for more than 95% of the GPU time. Therefore, kTanimotos/sec (kilo Tanimoto coefficients per second) is simply used as performance metric. The results from testing the Sybyl Database Comparison program on the three machines can be found in Table 6-6.

### 6.3.1 GPU-based Sparse Vector Algorithm Compared with Published Results

Our implementation of sparse vector algorithm achieved an average throughput of 28634 kTanimotos/sec on GTS 250, the graphics device of machine 3 (Table 6-5). Haque *et al.*<sup>198</sup> attained 60900 to 64230 kLINGOS/sec on the same GPU, GTS 250. Despite certain difference between Tanimoto and LINGO, the same underlying algorithms find the overlapping structural patterns. In addition, the sparse vector algorithm has a linear time complexity in the length of sparse vectors. The average length of sparse vectors in Haque's work ranged from 29.31 to 31.65 out of thousands of possible patterns. On the other hand, the average of sparse vectors in our sample libraries ranged from 185 to 196 out of a total of 992. Given the lower sparsity of the Unity fingerprints and the longer length of the sparse vectors in this study, the performance of our implementation of sparse vector algorithm was comparable to Haque's results. Similarly, Liao *et al.*<sup>199</sup> reported that all-vs-all Tanimoto matrix calculation for PB83 library took 17.1 seconds on device GTX 280 (240 CUDA cores). In their study, PB83 library contained 27674 molecules, and the maximum length of sparse vectors was 459. In our case, the library comparison of Library B to itself (Table 6-5) took 20.45 seconds on GTS 250 (128 CUDA cores). Library B contained 24755 molecules and the maximum length of sparse vectors was 440, which was similar to PB83. Note that GTX 280 had higher memory bandwidth and more cores than GTS 250. Although these experiments were carried out with different machines, fingerprints and datasets, the side-by-side analysis showed that the performance of our implemented sparse vector algorithm was at the same magnitude of published results.

**Table 6-2: The computation performance using integer fingerprint algorithm on machine 2**

Reference and Candidate Libraries	Time (sec)	Throughput (kTanimotos/sec)	Effective Bandwidth (GB/sec)
A vs A	2.16	45477	5.75
A vs B	5.30	46284	5.86
A vs C	11.97	46398	5.88
B vs A	5.25	46690	5.91
B vs B	13.30	46086	5.84
B vs C	29.61	46885	5.94
C vs A	11.83	46974	5.95
C vs B	29.58	46934	5.94
C vs C	68.09	46184	5.85



**Table 6-3: The computation performance using integer fingerprint algorithm on machine 3**

Reference and Candidate Libraries	Time (sec)	Throughput (kTanimotos/sec)	Effective Bandwidth (GB/sec)
A vs A	0.47	209507	26.43
A vs B	0.95	257483	32.68
A vs C	2.12	261684	33.18
B vs A	0.94	261884	33.03
B vs B	2.29	267136	33.90
B vs C	4.85	286175	36.26
C vs A	2.00	278064	35.17
C vs B	4.80	288914	36.63
C vs C	11.37	276519	35.03

**Table 6-4: The computation performance using sparse vector algorithm on machine 2**

Reference and Candidate Libraries	Time (sec)	Throughput (kTanimotos/sec)
A vs A	22.11	4434
A vs B	52.92	4631
A vs C	122.75	4523
B vs A	53.33	4596
B vs B	126.63	4839
B vs C	295.00	4705
C vs A	122.14	4546
C vs B	291.38	4764
C vs C	674.17	4664

**Table 6-5: The computation performance using sparse vector algorithm on machine 3**

Reference and Candidate Libraries	Time (sec)	Throughput (kTanimotos/sec)
A vs A	3.68	26636
A vs B	8.71	28158
A vs C	20.08	27658
B vs A	8.63	28413
B vs B	20.45	29963
B vs C	47.35	29320
C vs A	19.53	28429
C vs B	46.54	29831
C vs C	107.33	29301

### 6.3.2 Integer Fingerprint Algorithm versus Sparse Vector Algorithm

Sparse vector algorithm is designed to handle high-sparsity molecular fingerprints. Nevertheless, many of the popular molecular fingerprints are binary and have low sparsity, such as Unity, FP2, MACCS, and PubChem fingerprints. The integer fingerprint algorithm is capable of attaining higher computation efficiency for these types of fingerprints, because it can process many fingerprint “bits” in a single iteration. On the contrary, the sparse vector algorithm scans through every present structural pattern to search for the overlapping ones. Moreover, the “while” and “if-else” statements in the algorithm easily result in divergent execution paths on GPU, which reduces the degree of parallelism. The advantage of integer fingerprint algorithm on low-sparsity fingerprint is demonstrated by the experiment. As shown in Table 6-4 and Table 6-5, the throughput of sparse vector algorithm varies from 4434 kTanimotos/sec to 4839 kTanimotos/sec on the GPU of machine 2, and from 26636 kTanimotos/sec to 29963 kTanimotos/sec on the GPU of machine 3. Integer fingerprint algorithm delivers much higher throughput ranging from 45477 to 46974 on the GPU of machine 2 (Table 6-2), and from 209507 kTanimotos/sec to 288914 kTanimotos/sec on the GPU of machine 3. Among the nine cases, integer fingerprint algorithm is 9.5 to 10.3 times faster than sparse vector algorithm on machine 2, and 7.9 to 9.8 times faster than sparse vector algorithm on machine 3.

The performance of integer fingerprint algorithm is further assessed by effective memory bandwidth (the rate at which data is read from, and stored to graphical memory). The effective bandwidth of library comparison using integer fingerprint algorithm can be determined according to the following equation:

$$\frac{n \times (m + 1) \times 124 + n \times (m + 1) \times 4 + n \times m \times 2 \times 4 + n \times 2 \times 4}{\text{time in seconds}} \times 2^{-30} \text{GByte/sec}$$

Where  $n$  is the size of a candidate library, and  $m$  is the size of a reference library. Each virtual core reads a candidate fingerprint into cache and compares it to the whole reference library. As there are  $n$  candidate compounds and a Unity fingerprint occupies 124 bytes, the Tanimoto matrix calculation reads  $n \times (m + 1) \times 124$  bytes from graphical memory. The number of “1” bits of reference fingerprints and candidate fingerprints are accessed by each core, which brings  $n \times (m + 1) \times 4$  byte data. The Tanimoto matrix has  $n \times m$  float-type (32-bit) elements. The matrix is accessed twice (matrix generation and searching for maximum Tanimoto coefficients). GPU therefore reads and writes  $n \times m \times 2 \times 4$  bytes. Finally, the array containing the maximum of each row of the Tanimoto matrix is accessed, which leads to input and output of  $n \times 2 \times 4$  bytes. In fact, the effective memory bandwidth is somehow underestimated by this equation, because the total calculation time includes some other procedures, such as memory allocation and host-to-device data transfer.

As summarized in Table 6-2 and Table 6-3, the implemented integer fingerprint achieved average effective bandwidth of 5.88GB/sec on machine 2 and 33.59GB/sec on machine 3. The program from NVIDIA SDK<sup>17</sup> was used to test peak memory bandwidth, revealing that the GPU of machine 2 had 12.88GB/sec device-to-device copy bandwidth, and the GPU of machine 3 had 57.36GB/sec device-to-device copy bandwidth. The program therefore made good utilization of hardware resources.

Due to the use of intrinsic instructions (“&” operator and population count), integer fingerprint algorithm yielded much higher throughput than sparse vector algorithm regarding Tanimoto calculation. The analysis on effective memory bandwidth further depicted absolute computation performance. As a result, integer fingerprint algorithm is considerably superior to sparse vector algorithm when Tanimoto coefficient is measured on low-sparsity fingerprints, *e.g.*, Unity fingerprint.

### 6.3.3 Performance of GPU- and CPU-based Programs for Compound Library Comparison

**Table 6-6: The computation performance of Sybyl Database Comparison Program**

Reference and Candidate Libraries	Machine 1		Machine 2		Machine 3	
	Time (sec)	<sup>a</sup> Effective Throughput (kTanimotos/sec)	Time (sec)	<sup>a</sup> Effective Throughput (kTanimotos/sec)	Time (sec)	<sup>a</sup> Effective Throughput (kTanimotos/sec)
A vs A	8.38	11689	34.38	2852	10.58	9267
A vs B	23.60	10385	90.19	2718	29.38	8343
A vs C	78.77	7049	306.57	1811	98.02	5663
B vs A	21.96	11161	86.14	2845	27.78	8823
B vs B	52.98	11569	208.63	2937	66.17	9260
B vs C	190.91	7268	748.42	1854	235.45	5895
C vs A	73.34	7568	284.55	1951	91.59	6061
C vs B	176.48	7846	689.49	2013	225.55	6152
C vs C	393.43	7988	1535.34	2047	511.54	6143

<sup>a</sup> For easy comparison, the effective throughput is approximated assuming that the program finishes the whole Tanimoto matrix calculation, which is not necessarily required in practice. The throughput is in the unit of kTanimotos/sec.

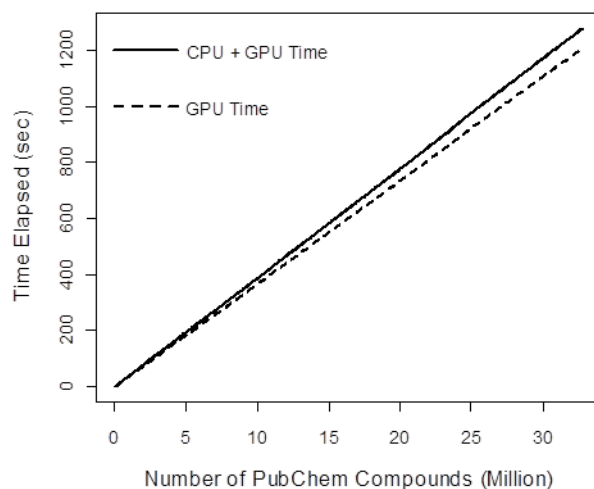
Table 6-6 shows that Sybyl Database Comparison, as a traditional CPU program, runs fastest on machine 1, with an average throughput of 9169kTanimotos/sec. Nevertheless, the program yields an average throughput of 2336 kTanimotos/sec on machine 2 and 7290 kTanimotos/sec on machine 3. Thus, the

performance of Database Comparison program on machine 1 is compared to the GPU program. The GPU implementation of compound library comparison using integer fingerprint algorithm shows an average 5.28 times speedup on the GPU of machine 2 (Table 6-2) in comparison with Sybyl Database Comparison program on machine 1 (Table 6-6). Further speedup can be seen on machine 3. Table 6-3 and Table 6-6 show that the GPU implementation is up to 39.47 times as fast as the CPU program, and 30.44 times speedup on average. FX 580 (the graphic device in machine 2) is considered as a low-end or entry-level device, as it only has 32 CUDA cores. Nevertheless, the GPU program still runs much faster than the commercial CPU program. This is mainly due to the multi-core parallel computing architecture of modern GPUs. In the GPU program, multiple threads are launched concurrently for Tanimoto calculation, searching for the maximums, histogram creation, and so on.

Additionally, the GPU program using sparse vector algorithm shows an average 3.26 times speedup relative to Sybyl Database Comparison program, when run on machine 3 (see Table 6-5 and Table 6-6). Nevertheless, the implementation using sparse vector algorithm is not necessarily as fast as the CPU program, when tested on machine 2. The integer fingerprint algorithm significantly outperforms the Database Comparison program even on low-end device, whereas the sparse vector algorithm only shows moderate speedup on machine 3. These results illustrate the fact that extra attention is required for GPU algorithm design and implementation in order to achieve optimal performance.

### **6.3.4 Validation on PubChem Database**

The integer fingerprint algorithm was further tested on a large compound database with the intention to examine its scalability and consistency. In this experiment, Library A (9902 compounds) was compared with 32.79M PubChem structures (by April 2011). The computation was performed on machine 3, and 324.69G Tanimoto coefficients were generated. Figure 6-5 plots the amount of aggregated time as a function of the number of PubChem structures. The whole process took 1279 seconds, and GPU time accounted for 94% of it, *i.e.*, 1207 seconds. Thus, the average throughput was 269041 kTanimotos/sec, which resembled the results summarized in Table 6-3. Figure 6-5 also reveals that the elapsed time is strictly linear to the quantity of processed structures, suggesting that the program generates stable throughput regardless of diverse structures.



**Figure 6-5: The plot of elapsed time versus processed PubChem compounds.**

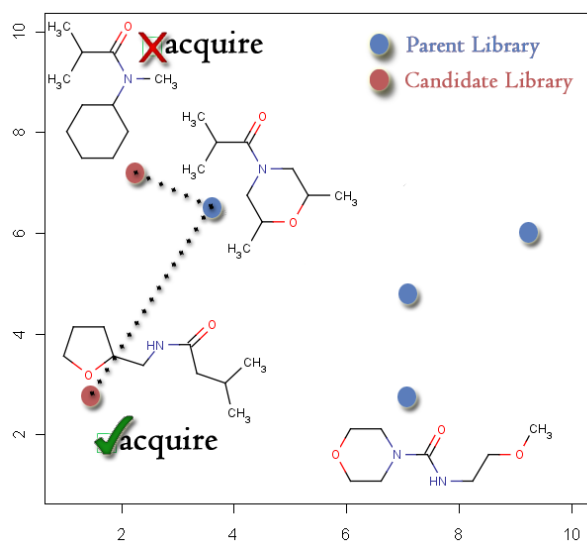
Dashed line shows the elapsed GPU time as a function of the number of compounds. The GPU time includes time allocated for host-device data transfer, Tanimoto matrix calculation, histogram creation, and so on (Figure 6-1). The solid line demonstrates the total calculation time. This is equal to GPU time plus CPU time. CPU time is for hard drive I/O, task scheduling, *etc.*

## 6.4 CONCLUSION

The ever-growing chemical libraries demand the development of efficient algorithms and programs for chemical similarity calculation that plays a fundamental role in cheminformatics. As a similarity index, Tanimoto coefficient is widely involved in data mining of small molecules, such as compound clustering, diversity analysis, and k-nearest-neighbor search (kNN). The quadratic time complexity in the number of compounds could be a problem for these techniques, particularly when large compound datasets are presented. Parallel Tanimoto calculation on modern GPUs points out a potential solution to these challenges. In this chapter, we report a GPU-accelerated integer fingerprint algorithm and its application for calculating Tanimoto coefficients. The test calculation shows that the integer fingerprint algorithm runs up to 10 times faster than the published sparse vector algorithm on GPUs, and up to 39 times faster than the CPU-based commercial program. The GPU-accelerated integer fingerprint algorithm produces high throughput for low-sparsity binary fingerprint. For example, more than 200 million Tanimoto

coefficients can be calculated every second on a decent device, such as GTS 250. With this speed, comparing a 9902 Active Probe compounds with PubChem library, that has totally 324G Tanimoto coefficients to calculate, can be completed in about 20 minutes. In this study, compound library comparison can be interpreted as 1-nearest-neighbor search. This approach could be easily upgraded to k-nearest-neighbor search by a minor modification (The source code is accessible at <http://www.cbligand.org/gpu>). Additionally, the all-vs-all Tanimoto matrix calculation in the GPU program can also be adapted by other algorithms. Currently, we are conducting the application of the GPU-based algorithm to modeling quantitative structure-activity relationship for large datasets.

## 7 COMPOUND ACQUISITION ALGORITHM



In this chapter, a compound acquisition and prioritization algorithm is reported for rational chemical library purchasing or compound synthesis in order to increase the diversity of an existing compound collection. This method was established based on chemistry-space calculation using BCUT (Burden CAS University of Texas) descriptors. In order to identify the acquisition of compounds from candidate collections into the existing collection, a derived distance-based selection rule was applied, and the results were well supported by pairwise similarity calculations and cell-partition statistics in chemistry space. The correlation between chemistry-space distance and Tanimoto similarity index was also studied to justify the compound acquisition strategy through weighted linear regression. Then, a case study is followed to demonstrate its application in real world chemical synthesis. As a rational approach for library design, the distance-based selection rule exhibits certain advantages in prioritizing compound selection to enhance the overall structural diversity of an existing in-house compound collection or virtual combinatorial library for in silico screening, diversity oriented synthesis and high-throughput screening.

## 7.1 INTRODUCTION

Although modern HTS technologies can screen millions of compounds more quickly and cheaply than ever before, it is still challenging for a small pharmaceutical company or an academic institution to cover the costs in the absence of significant funds. Moreover, interrogating a large number of compounds generates unmanageable false positives. Thus, it is particularly necessary and important to build high-quality compound screening sets for some bioassays that have low screening throughput capacity or are limited by the availability of key reagents (e.g., antibodies, primary cells, or whole organism systems). In contrast to a large combinatorial screening collection that targets structural variations for structure-activity relationship (SAR) studies, a high-quality screening compound set built by rational acquisition of structurally diverse compounds potentially improves the HTS/HCS hit rate while preserving resources.

To build a compound collection for virtual screening or high-throughput screening, an ideal strategy seeks balanced tradeoff between overall molecular diversity and the number of compounds. Molecular diversity may be assessed by the variety of molecular properties, which is encoded by molecular descriptors such as physicochemical properties, topology index, or fingerprints.<sup>201</sup> Enhancing molecular diversity or removing redundancy can be achieved by four categories of approaches: cluster-based method, dissimilarity-based method, cell-based method and optimization-based method.<sup>202</sup> A cluster-based method is implemented to assign compounds into groups so that compounds possess higher within-group similarity than between-group similarity.<sup>203</sup> Once compound similarity is solved, a hierarchy-clustering algorithm, such as neighbor joining, or non-hierarchy algorithm, such as K-means, can be carried out for clustering. The motivation for applying a chemical dissimilarity-based method is to maximize the total dissimilarity between each pair of nearest neighboring compounds.<sup>204</sup> Relying on some linear or non-linear binning procedure, a cell based method aims to cover more cells with a minimal number of compounds, categorizing compounds in the same cell as similar.<sup>60</sup> An optimization-based approach enhances the diversity by optimizing the object function that may incorporate a set of descriptors to measure the molecular diversity in different criteria.<sup>205-206</sup> Although the approaches involving molecular diversity are frequently mentioned, there is still no widely accepted quantitative procedure for the prioritization and acquisition of new compounds to increase the structural diversity of an existing compound collection.

Among various molecular descriptors, BCUT descriptors<sup>58-61</sup> incorporate comprehensive information regarding molecular structure, atom property and more into decimal numbers. A general description of BCUT descriptors is given in section 2.2.3 Topological Descriptor. Creating BCUT descriptors is one of the most popular approaches to construct low-dimensional chemistry space and perform diversity analyses. The performance of BCUT descriptors has been validated through previous QSAR studies<sup>62-64</sup>



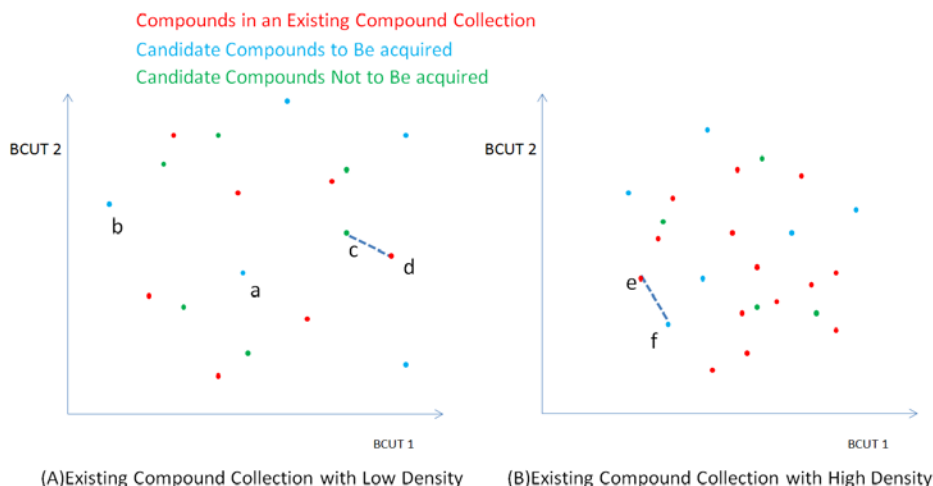
and successful applications in library design.<sup>207-208</sup> While BCUT descriptors demonstrate the relevance of generating a representative PubChem library<sup>65</sup> and diversity analysis,<sup>66</sup> I am expanding their use for the acquisition of new candidate compounds from external compound collections, in order to optimize an existing in-house screening set and increase its overall diversity.

A compound acquisition and prioritization algorithm is established on the Euclidean distance in BCUT chemistry space. This method is validated using weighted linear regression between the Euclidean distance and similarity index. Results from two case studies demonstrate that the selected subsets of external candidate compound collections enhanced the overall chemical diversity of an existing in-house screening collection, according to chemistry-space cell partition statistics and similarity index. Discussions are also presented on distance cutoff value and disagreement between the chemistry-space distance and similarity index. The algorithm provides useful information to facilitate decision-making for acquiring new candidate compounds and prioritizing compound syntheses.

## **7.2 ALGORITHM DESIGN AND EXPERIMENTAL PROTOCOLS**

### ***7.2.1 BCUT Chemistry Space and Compound Acquisition Protocol***

The established compound acquisition and prioritization algorithm is based on BCUT chemistry-space calculation using the protocol reported.<sup>65</sup> Briefly, BCUT descriptors<sup>60</sup> are defined by combining atomic descriptors for each atom and description of the nominal bond-types for adjacent and nonadjacent atoms into BCUT matrices. The value of each chemistry-space coordinate is specified as the highest or lowest eigen-value of BCUT matrix. In this project, the Diverse Solutions program (Tripos Sybyl 8.0)<sup>5</sup> was used to generate a set of default 2D BCUT descriptors that covered different scaling factors and atomic properties, including H-bond donor, H-bond acceptor, partial charge and polarity. The optimal combination of descriptors was selected automatically by the program to construct BCUT chemistry space, with the restriction that the correlation coefficient between any pair of BCUT descriptors was less than 0.25.



**Figure 7-1: Motivation of compound acquisition protocol**

A graphic representation of BCUT chemistry space to illustrate the concept and effect of the density of an existing compound collection. The compound collection with low density (A) sparsely covers the BCUT chemistry space, while the one with high density (B) exhausts the chemistry space more specifically. The choice of distance cutoff value depends on the density of the existing compound collection.

The computational protocol of the compound acquisition and prioritization algorithm using chemistry-space distance calculation is summarized below:

1. Initialization: define BCUT chemistry space and specify a distance cutoff value,  $c$  based on **Distance Threshold** calculated below.
2. Iteration: for each compound,  $j$ , in the candidate compound collection,
  - a. Calculate its distance to the nearest neighbor from the current compound collection,  $S$ :
 
$$D_j = \min_i |y_j - x_i|$$

$y_j$  is the descriptor vector of candidate compound  $j$ , and  $x_i$  is the descriptor vector of compound  $i$  in the current compound collection,  $S$ .
  - b. If the distance to the nearest neighbor  $D_j > c$ , then add the compound  $j$  into the current compound set:  $S \leftarrow S + \text{candidate compound } j$ .
  - c. Go to step 2 to analyze next candidate compound.

This method is rationally justified through the correlation studies between Euclidean distance in the BCUT chemistry space and Tanimoto coefficient from MACCS key fingerprints. The results are given later.

### 7.2.2 Distance Threshold

By default, the distance cutoff value,  $c$ , is defined as the estimated density of the existing compound collection according to the equation,

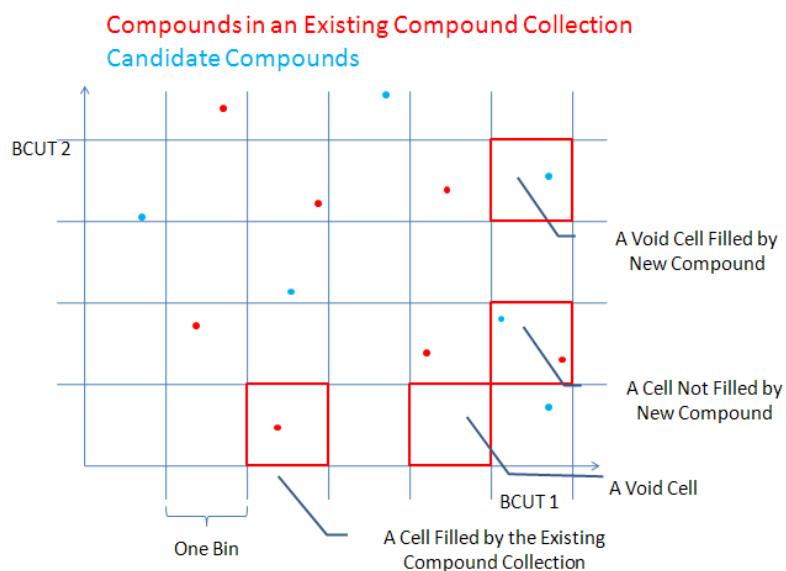
$$c = \frac{1}{N} \sum_{j=1}^N \min_{i, i \neq j} |x_i - x_j|$$

where  $i, j$  are the compound indices for the existing collection. The density indicates how well the chemistry space was explored or exhausted in the previous experiment. Thus, the new candidate compounds are also expected to cover the chemistry space in a similar pattern. Figure 7-1 illustrates how the density of an existing compound collection affects the choice of acquired compounds. Candidate compounds with large distance to their nearest neighbors in the existing collection are considered dissimilar to the compounds in the existing collection, and such candidate compounds are recommended for acquisition (like points “a” and “b” in Figure 7-1A). On the other hand, the candidate compound, “c” in Figure 7-1A, is excluded from the acquisition list due to its short distance to its nearest neighbor, “d”. However, the compound, “f” in Figure 7-1B, is still to be acquired, although the distance to its nearest neighbor “e” is almost the same as the distance between “c” and “d” (Figure 7-1A). The different acquisition decisions for similar circumstance can be explained by the density of two existing compound collections. In Figure 7-1A, the established compound dataset may be primarily designed to search the chemistry space sparsely. The high-density dataset in Figure 7-1B may explore the chemistry space more thoroughly. Therefore, the decision-making relies on the profile of the existing compound collection. In this method, the default distance threshold is equal to the density of the existing compound collection.

### 7.2.3 Molecular Diversity Analyses

A structurally diverse compound collection is expected to cover well-defined chemistry space uniformly. The chemical diversity of a compound dataset may be measured in a binning procedure<sup>209</sup>. The binning procedure is used to generate “cells” in a multi-dimensional descriptor space. Each dimension is divided uniformly into a finite number of “bins”. The bin-definition defines multi-dimensional “cells”, which cover the entire space. The chemical diversity could be accessed by counting the number of filled cells. As illustrated in Figure 7-2, the concepts regarding a bin and a filled/void cell are given in a hypothetical plot of two-dimensional BCUT chemistry space. As shown in the plot, the acquired compound filling a

void cell is believed to increase the overall structural diversity. On the other hand, the new compound in the cell that already has compounds (red dots) from the existing compound collection does not contribute to increase structural diversity and is not recommended to be acquired or purchased.<sup>60</sup>



**Figure 7-2: Two-dimensional chemistry space and filled/void cells**

In this study, four-dimensional instead of two-dimensional chemistry-space was constructed. The entire space was partitioned into  $100^4$  cells with the same volume by dividing each axis into 100 bins equally. Each cell was indexed by  $(I_1, I_2, I_3, I_4)$ . Indices  $I_k$  were integers ranging from 0 to 99. A cell indexed by  $(I_1, I_2, I_3, I_4)$  represented a subspace  $R(I_1, I_2, I_3, I_4) = \{ (x_1, x_2, x_3, x_4): I_k \times 0.1 \leq x_k < I_k \times 0.1 + 0.1, k=1, 2, 3, 4 \}$ . Finally the number of filled void cells by candidate compounds was sorted out to describe diversity increment.

Candidate compounds could also be compared to an established compound collection to characterize the degree of similarity between two compound datasets, according to molecular fingerprint. This approach measures how closely the candidate compounds are represented in the existing compound collection by Tanimoto coefficient.<sup>14</sup> The degree of similarity between candidate compounds and the existing compound collection was evaluated by Database Comparison program (Tripos Sybyl) based on UNITY fingerprint, as described below.

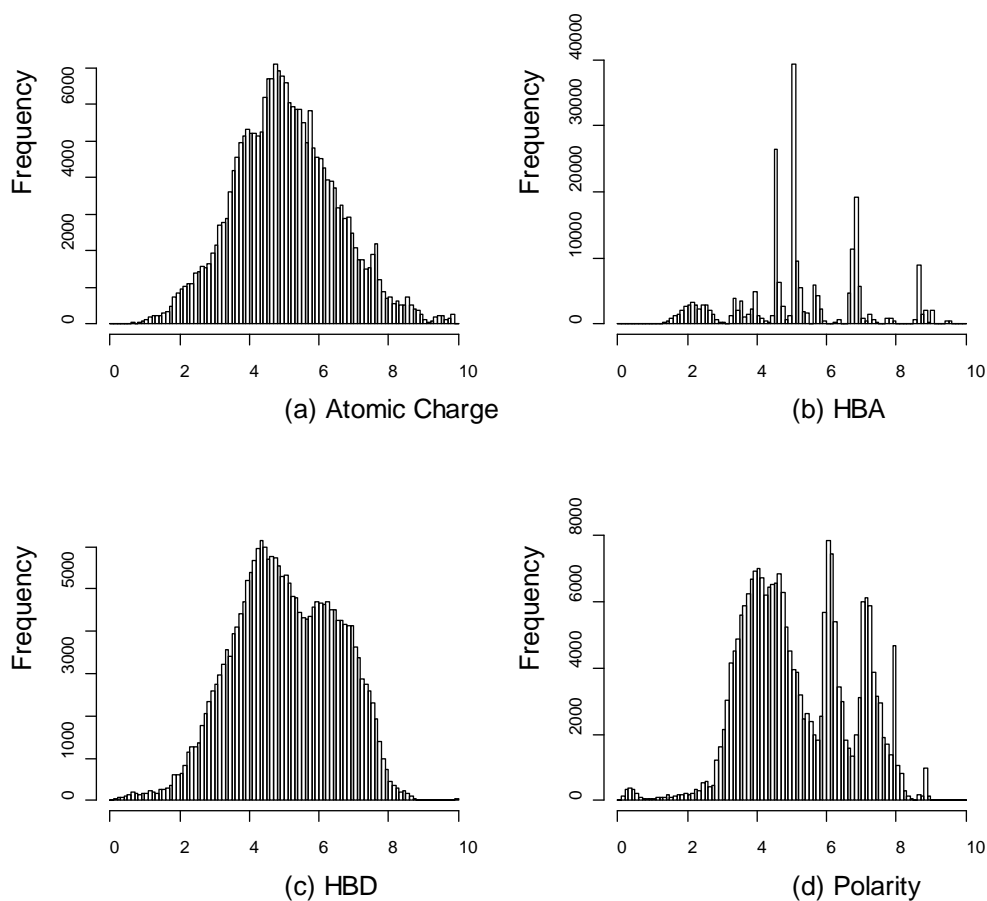
### 7.3 RESULTS AND DISCUSSION

Similar to other drug screening centers or institutes, the University of Pittsburgh Drug Discovery Institute (UPDDI) faces the issue of building a high-quality chemical library in terms of library size and structural diversity associated with the cost of purchasing and storage. In this section, the rationality of the compound acquisition and prioritization algorithm, together with its application, is presented through guiding candidate compound acquisition in order to increase diversity of the current PMLSC screening set that contains 230k compounds from the Pittsburgh Molecular Libraries Screening Center (PMLSC, [pmlsc.pitt.edu](http://pmlsc.pitt.edu)). For this illustration, two commercial libraries, TimTec 3k Natural Derivatives Library (NDL)<sup>210-211</sup> and TimTec 2k Active Probes Library (APL) were selected as candidate compound collections, from which compounds were prioritized and selectively deposited into the PMLSC screening set.

**Table 7-1: The specifications of BCUT descriptors for constructing four-dimensional chemistry space**

Diagonal Element	Off-diagonal Element	Scaling factor	Remove(R) or keep(K) hydrogen	Use lowest(L) or highest(H) Eigen value
GasTchrg (Atomic Charge)	Burden	0.1	R	H
Haccept (HBA)	Burden	0.9	R	H
Hdonor (HBD)	Burden	0.75	R	H
Tabpolar (polarity)	Burden	0.5	R	H

Four atom properties (partial charge, polarity, H-bond donor, and H-bond acceptor in diagonal elements) were considered to calculate BCUT. According to PMLSC screening set, the best combination of scaling factor and the choice of eigen-value were selected to construct chemistry space. The value of each BCUT descriptor was scaled to range from 0 to 10. The distribution of each BCUT descriptor of the PMLSC screening set is shown in Figure 7-3, and the specifications of BCUT descriptors are listed in Table 7-1. The correlation coefficient,  $r^2$ , between any pair of dimensions was less than 0.11, suggesting that every dimension independently described different aspects of molecular properties.

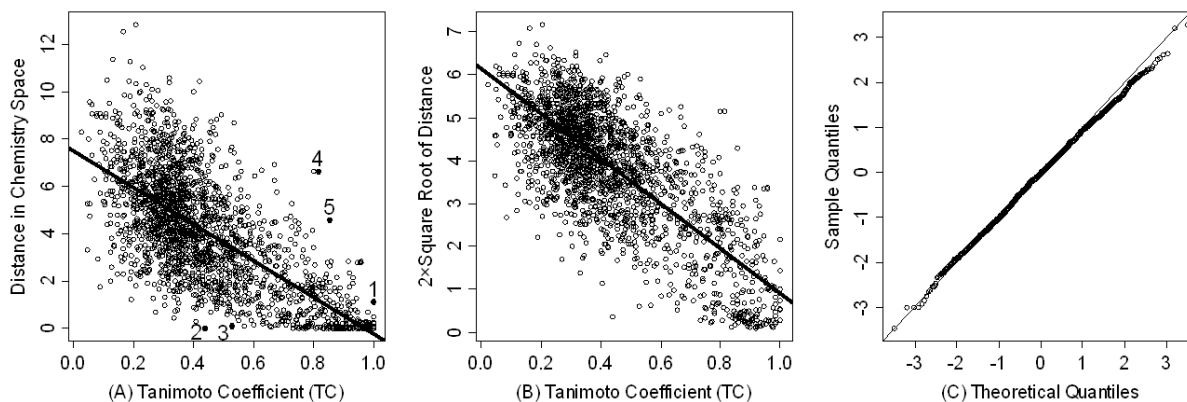


**Figure 7-3: The distribution of four chemistry-space descriptors for the PMLSC screening set**

(a) the histograms of atomic partial charge descriptor, (b) H-bond acceptor descriptor, (c) H-bond donor descriptor, and (d) polarity descriptor.

In the compound selection or prioritization algorithm, high acquisition priority was assigned to the candidate compounds that had large chemistry-space distances to their nearest neighbors in the existing compound collection. For the validation of this method, 1991 pairs of compounds were selected sequentially from the Active Probes Library (APL) in order to study the correlation between Tanimoto coefficient (Tc) and chemistry-space distance through weighted linear regression. MACCS<sup>9</sup> key molecular fingerprints were then generated to calculate Tanimoto coefficient for these compounds pairs, and their chemistry-space distances were evaluated in the chemistry space defined by PMLSC screening set. Figure 7-4A displays the scatter plot of the raw Euclidean distance in chemistry space and the

calculated Tanimoto coefficient (Tc) similarity score of 1991 pairs of compounds in APL. The Tc values of 1991 compound pairs range from 0.023 to 1.000 and their distances range from 0.002 to 12.824.



**Figure 7-4: Correlation between Tanimoto Coefficient and Euclidean distance in BCUT chemistry space**

(A) The scatter plot between the Euclidean distance in BCUT chemistry space and Tanimoto coefficient (Tc) of 1991 pairs of compounds in Active Probes Library (APL). The fitted regression line and five labeled outliers are also shown. The Tanimoto coefficients are calculated according to MACCS fingerprint; (B) The scatter plot of Tanimoto coefficient (Tc) and transformed Euclidean distance for the 1991 pairs of APL compounds with the weighted regression line. The weight for each point is its Tc value.

For a correlation study, the distance in chemistry space was transformed to normalize its variance as a function of Tc (Figure 7-4B). As fingerprints were developed to measure compound similarity instead of dissimilarity,<sup>60</sup> weighted regression was performed to emphasize the significance of high Tc values. Figure 7-4B shows the scatter plot of  $2 \times \sqrt{D}$  ( $D$ : chemistry-space distance along y-axis) and Tanimoto coefficient (Tc along x-axis) of 1991 pairs of APL compounds together with the fitted regression line. The regression equation was then solved as:

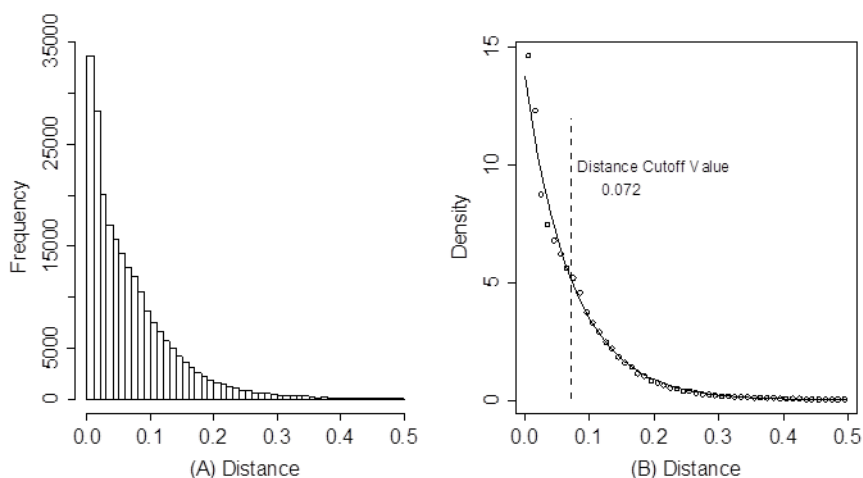
$$2 \times \sqrt{D} = \alpha + \beta \times \text{Tc}$$

where  $\alpha = 6.13$ ,  $\beta = -5.23$ ; and the correlation coefficient,  $r^2$ , was 0.61.

The corresponding normal Q-Q plot of regression residuals is shown in Figure 7-4C. Q-Q plot is an effective technique to examine the distributions of two sets of samples by plotting quantiles against each other. According to the Q-Q plot, the distribution of regression residuals that were the difference between fitted values and corresponding observed values was close to standard normal distribution, allowing for hypothesis testing to examine the correlation of those two variables. Based on Figure 7-4B, hypothesis

testing resulted in a two-sided p-value  $< 0.0001$ , which was strongly against null hypothesis  $\beta = 0$  and favored alternative hypothesis  $\beta \neq 0$ . This statistical result suggested a fine negative correlation between the chemistry-space distance and Tanimoto coefficient calculated by MACCS fingerprint. Therefore, candidate compounds with large distances to their nearest neighbors were expected to be dissimilar to the compounds in the existing compound collection, and acquiring such compounds would efficiently enhance the overall chemical diversity.

Despite favorable correlations, discrepancies still existed between chemistry-space distance and Tc, as illustrated by five pairs of labeled outliers in Figure 7-4A. The structures of the compound pairs are listed in Table 7-2. MACCS fingerprint based similarity Tc calculation did not detect the structural difference for the compound pair 1 (AP-49 and AP-50), showing a Tc value of 1.0 (Table 7-2). However, the distance between them was considered to be relatively large (distance = 1.16) in the BCUT chemistry space, which could reflect different  $\pi$ -conjugated systems between two compounds. The subtle feature is sometimes important for biological activities. On the other hand, the compound pair 2 (AP-526/AP-527) and pair 3 (AP-230/AP-231) were quite similar with reported distance of 0.031 and 0.095 respectively, while the Tc value was less than 0.85, indicating structural difference between them. The large distances between compound pair 4 and pair 5 (distance = 6.65 and 4.61 respectively) were essentially due to the BCUT polarity descriptor.

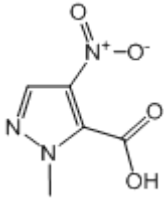
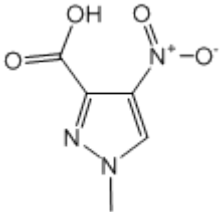
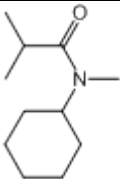
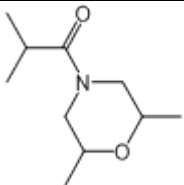
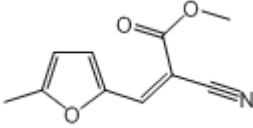
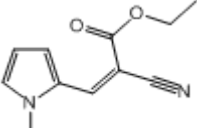
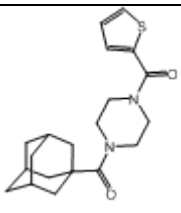
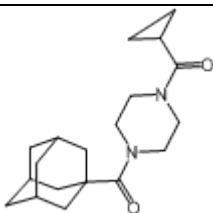
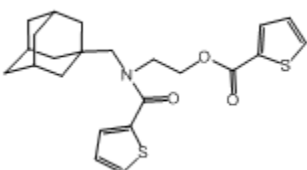
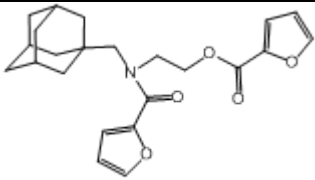


**Figure 7-5: Histogram and probability density**

(A) The histogram of the distances between nearest neighboring compounds in the existing screening collection. (B) The normalized histogram with fitted exponential probability density function (PDF). The default distance cutoff value is 0.072.



**Table 7-2: Five pairs of compounds illustrate some outliers in Figure 7-4A.**

Compound Pair		Distance	Tc
 <p>AP-49</p>	 <p>AP-50</p>	1.16	1.0
 <p>AP-526</p>	 <p>AP-527</p>	0.031	0.44
 <p>AP-230</p>	 <p>AP-231</p>	0.095	0.53
 <p>AP-1685</p>	 <p>AP-1686</p>	6.65	0.81
 <p>AP-1665</p>	 <p>AP-1666</p>	4.61	0.85

For example, the calculated electric dipole of compound AP-1665 was 2.61 Debye, while the dipole of AP-1666 was 5.34 Debye (according to original structure and Gasteiger–Hückel charge). Thus, BCUT descriptors characterize structural topology together with atom properties and possess certain advantages for constructing low-dimensional chemistry space, compared to molecular fingerprint.

The distribution of distances between all pairs of nearest-neighboring compounds in the PMLSC screening set is shown in Figure 7-5A. The probability density function (Figure 7-5B) of exponential distribution was fit to the normalized histogram:  $f(x) = \frac{1}{\lambda} \exp(-\frac{x}{\lambda}); \lambda = 0.072589, x \geq 0$ .

Thus, the expectation of distance between one pair of nearest-neighboring compounds was 0.072. As shown Figure 7-5B,  $\lambda$  could be regarded as the density of an existing compound collection, so  $\lambda$  was the default threshold for compound selection. In the present case, the distance threshold value,  $c$ , was equal to 0.072.

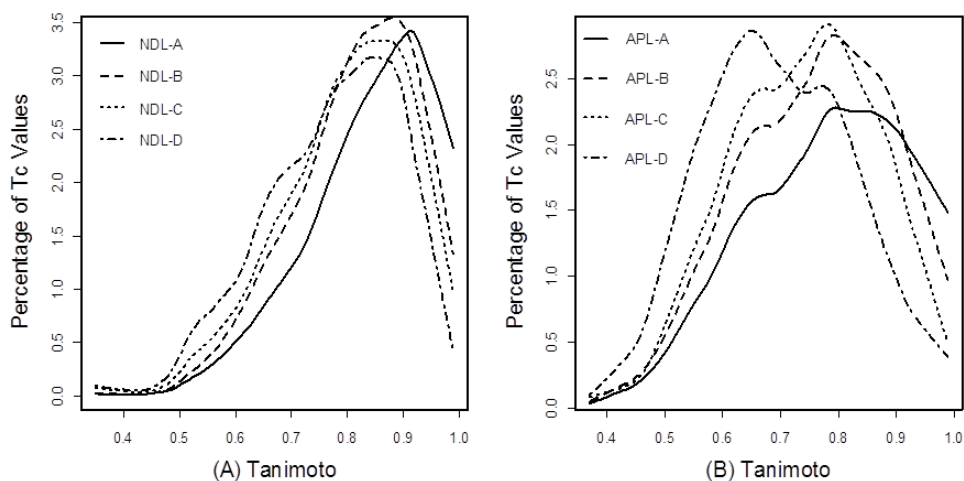
For comparison, different subsets of commercial compound collections (NDL and APL) were generated and compared to the current PMLSC screening collection, with the intention to justify the compound acquisition method. As shown in Table 7-3, 1648 compounds from NDL (NDL-B) and 1096 compounds from APL (APL-C) were selected according to the acquisition protocol, using distance threshold 0.072. Alternatively, the top 1000 and 500 compounds were selected to create another two subsets, NDL-C and NDL-D respectively, after ranking NDL compounds descendingly according to their distances to the nearest neighbors from the PMLSC screening set. The same strategy was also applied to select 1500 and 500 APL compounds (APL-B and APL-D).

**Table 7-3: The average and standard deviation of Tc for different NDL and APL compound subsets, compared to the PMLSC screening collection.**

Natural Derivatives Library			Active Probes Library		
Subset	Size	Mean/Stdev Tanimoto	Subset	Size	Mean/Stdev Tanimoto
NDL-A	3000	0.8593±0.11	APL-A	2000	0.8134±0.15
NDL-B	1648	0.8211±0.11	APL-B	1500	0.7636±0.13
NDL-C	1000	0.8071±0.11	APL-C	1096	0.7460±0.13
NDL-D	500	0.7851±0.12	APL-D	500	0.7004±0.13

The Tanimoto coefficients are calculated by Database Comparison program that is based on UNITY fingerprint.

To investigate the correlation between chemistry space distance and Tc in a larger scale, the whole NDL, APL and their subsets were compared to the PMLSC screening set using Database Comparison program. It is worth pointing out that Database Comparison program characterizes the degree of overlapping between two compound collections using UNITY fingerprint and Tanimoto coefficient. As subsets NDL-C, NDL-D, APL-B and APL-D were not created by compound acquisition protocol, they might possess high intra-subset similarity. Database Comparison program was used to examine the between-collection similarity, i.e. comparing NDL or APL subsets to the PMLSC screening set, to show the effect of chemistry space distance on Tanimoto similarity index. Table 7-3 summarizes the sample mean and standard deviation of Tc values for different NDL and APL subsets, when compared to the PMLSC screening set. As shown in the table, the subset NDL-A, the whole NDL, possesses average Tc 0.8593 in comparison to the PMLSC screening set, whereas the subsets, NDL-B, NDL-C and NDL-D, have average Tc values of 0.8211, 0.8071 and 0.7851, when the number of acquired compounds is 1648, 1000 and 500 respectively. A similar trend is also observed with the APL, which possesses an average Tc value of 0.8134 to the PMLSC collection. The average Tc values between the APL subsets and the PMLSC collection decrease from 0.8134 to 0.7004, as the size of acquired compounds is reduced from 2000 to 500.



**Figure 7-6: Distribution of Tanimoto Coefficients from Database Comparison**

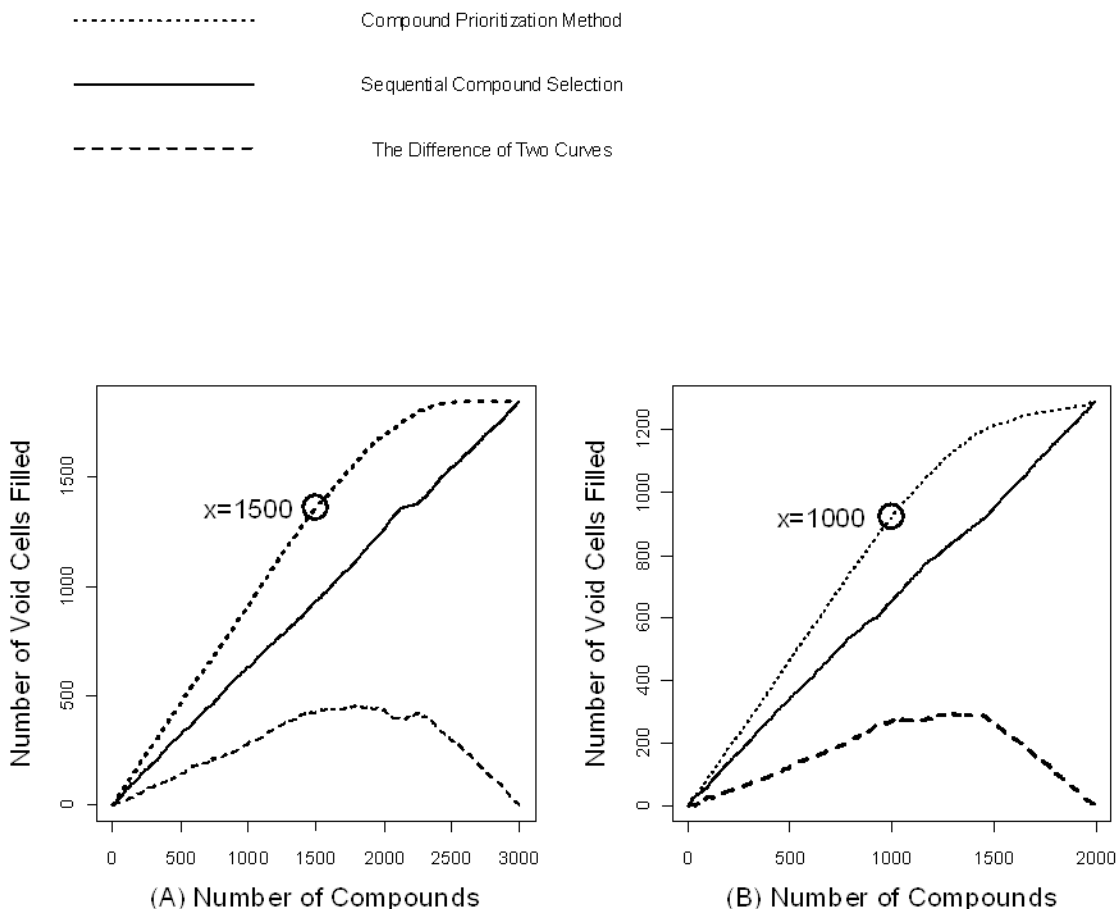
- (A) The distribution of Tanimoto coefficient (Tc) values for four NDL subsets: NDL-A with 3000 compounds, NDL-B with 1648 compounds, NDL-C with 1000 compounds, NDL-D with 500 compounds;  
 (B) The distribution of Tanimoto coefficient (Tc) values for four APL subsets: APL-A with 2000 compounds, APL-B with 1500 compounds, APL-C with 1096 compounds, APL-D with 500 compounds.  
 The Y-axis is the percentage of Tanimoto values that fall into every 0.01 interval.

The Database Comparison program calculates the Tanimoto between all candidate compounds and their nearest neighboring compounds in the PMLSC screening set. Thus, a set of Tanimoto coefficients were reported after comparing NDL or APL to the PMLSC screening set, and a histogram was created to examine the distribution of these Tc values. The distributions of Tc values are plotted in Figure 7-6A for NDL subsets and Figure 7-6B for APL subsets. In Figure 7-6A, the solid line represents the density profile of Tc values between all the NDL compounds and their most similar counterparts in the PMLSC compound collection in 0.01 intervals. While the data shows 12% NDL compounds with Tc=1.0 to the PMLSC screening set (data point not shown in Figure 7-6A), the peak of the curve is around Tc = 0.91, indicating a relatively large portion of NDL compounds with Tc = 0.91. The distributions of Tc values from NDL subsets with size 1648, 1000 and 500 are represented by a dashed line, a dotted line and a dash-dotted line, respectively. Any of the three subsets contains less than 1% of compounds that possess Tc = 1.0 to the PMLSC collection. As the size of NDL subsets decreases from 3000 to 500, the distribution shifts to the lower Tc value, indicating that smaller subsets tend to be increasingly dissimilar to the PMLSC screening set.

Figure 7-6B reveals a similar pattern for APL. 15% of APL compounds have Tc = 1.0 to their most similar counterpart in PMLSC collection, while none of APL-B, APL-C and APL-D possesses more than 1% of Tc that is 1.0 (data point not shown in the Figure). The peaks also shift towards the lower value of Tc as the size of APL subsets decreases. In general, the correlation between subset size and Tc distribution can be explained by the regression study as shown in Figure 7-4 above. Figure 7-4 reveals the negative correlation between Tc values and chemistry-space distance. In other words, the Tc value between a pair of compounds tends to decrease as their chemistry-space distance increases. As the NDL or APL candidate compounds were selected according to the distances to their nearest neighbors, smaller subset had larger average distance to the PMLSC screening set. Thus, the smaller subset tended to be dissimilar to the PMLSC compound collection, even if the similarity score was calculated by Database Comparison program based on UNITY fingerprint (Figure 7-6 and Table 3). While any novel candidate compounds would add certain structural diversity to an existing compound collection, the amount of to-be-acquired candidate compounds should be carefully determined to balance the quality and quantity through the choice of distance cutoff value. The density of an established compound collection is the recommended distance cutoff value, because it reflects how the compound dataset explores the chemistry space. Furthermore, the chemistry-space distance between identical compounds is zero, because identical compounds have the same coordinate values. Import of any duplicate candidate compounds into the PMLSC screening set is avoided by applying an appropriate distance cutoff value. For example, 12% of compounds in NDL were duplicates to the PMLSC screening set and removed from the wish list.

Conversely, in the case studies, less than 1% of compounds present in NDL-B, NDL-C and NDL-D possessed  $T_c = 1.0$  to the PMLSC screening set. This is attributed to the fact that a pair of compounds possessing  $T_c = 1.0$ , such as the compound pair 1 (AP49/AP50) from Table 7-2, may not necessarily be identical.

The similarity assessment from Database Comparison program described the degree of overlapping between two compound collections, yet it was unable to provide a quantitative measure of the overall diversity increment. Therefore, a “binning” procedure described in Methods section was applied to the BCUT chemistry space for diversity assessment. For the binning procedure, the bin size was required to determine the volume of “cells” in chemistry space. A large bin size would reduce the sensitivity of diversity measurement, where as a small bin size would trap most candidate compounds in void cells and make counting the filled void cells meaningless. A reasonable bin size could be determined in consideration of the size and density of an existing compound collection, or the regression analysis illustrated in Figure 7-4. For this study, the size of one bin was set to 0.1, which was at the magnitude of the density of the PMLSC compound collection.



**Figure 7-7: The number of filled void cell versus the number of acquired compounds**

The plots of the number of filled void cells as a function of the number of candidate compounds that are selected sequentially or acquired by the compound acquisition method. Plot (a) is for NDLC compounds and plot (b) is for APL compounds.

Figure 7-7 visualizes the number of filled void cells by applying the established compound acquisition and prioritization algorithm and gradually relaxing the threshold distance value,  $c$ , until all the NDLC and APL compounds were deposited into the PMLSC screening set. The X-axis denotes the number of the deposited compounds, while the Y-axis denotes the number of void cells filled by the corresponding compounds. Figure 7-7 shows an approximate linear growth of the number of filled cells (dotted lines), when less than 1500 NDLC compounds or less than 1000 APL compounds are deposited into the PMLSC screening set (data points circled in Figure 7-7). At the early stage, the deposited candidate compounds surely filled a void cell due to the large distance to their nearest neighbors in the PMLSC compound

collection. As the number of acquired compounds increased, newly acquired ones tended to be closer to their nearest-neighbors, and some of them might be located in the same cells where some PMLSC compounds were already present. As the circled points marked in Figure 7-7, the derivative of the number of filled cells began to decrease after acquiring more than 1500 NDL compounds or 1000 APL compounds, respectively. Subsequently, fewer and fewer void cells were filled as more candidate compounds were acquired. Finally, the number of filled cells reached a plateau after depositing approximately 2500 NDL compounds with 1844 filled void cells, and 1500 APL compounds with 1290 filled void cells. For comparison, candidate compounds from NDL and APL were sequentially merged into the PMLSC compound collection. Because the candidate libraries and the PMLSC screening set were prepared independently for the calculation, there was an equal probability to fill a void cell by any NDL or APL compound.

A close analysis of plots in Figure 7-7 also reveals that the solid lines, representing the number of filled cells under the sequential compound acquisition, demonstrate nearly linear growth with the number of candidate compounds. The dashed lines in Figure 7-7 represent the difference in the number of filled cells between the established compound acquisition method and sequential compound selection. The dashed lines reached the plateau when approximately 1700 NDL compounds and 1200 APL compounds were acquired. After the plateau of the dashed lines, the diversity analysis showed that the low priority compounds did not significantly fill the void cells or increase the diversity of the PMLSC screening set. Consequently, the plateau indicated the optimal number of compounds to be acquired under the current chemistry space binning procedure. This conclusion was also supported by the number of acquired compounds with the default distance threshold, which instructed us to acquire 1648 compounds from NDL and 1096 compounds from APL.

## 7.4 CONCLUSION

Through the application of BCUT descriptors, I have constructed multiple dimensional chemistry space for compound acquisition and prioritization. As pointed out above, high-quality diverse compound collections play a significant role in virtual screening and HTS/HCS campaigns. In general, a structurally diverse library, or representative subset, is constructed directly or indirectly from compound collections in order to minimize the experimental bioassay costs, but this may result in a failure to identify active compounds or promising “leads”, namely false negatives. Thus, thoughtfully expanding the screening sets

and testing these newly acquired compounds provide opportunities to cover more structural chemistry space, while avoiding duplicating the testing of structurally similar compounds. However, the acquisition of candidate compounds should be performed in carefully designed chemistry space that is within a biological meaningful context, because the interpretation of “diversity” is directly determined by chemistry-space coordinates. Cautions should be taken that solely blinded pursue of structural dissimilarity may bring in irrelevant compounds and impair the outcome of virtual screening or high-throughput screening.

In the compound acquisition protocol, candidate compounds are acquired or deposited into an existing compound collection according to Euclidean distance in BCUT chemistry space. In order to rationalize this approach, a regression analysis was carried out to model the correlation between chemistry distance and Tanimoto coefficient based on MACCS key. Statistical results indicated negative correlation between the two variables, supporting the conclusion that a pair of compounds tended to be dissimilar if the chemistry distance between them was large. Different sizes of NDL and APL subsets were then generated and compared to the PMLSC screening set in order to show the correlation between Tanimoto similarity index and chemistry space distance in a compound collection scale. Next, the diversity assessment was implemented to demonstrate how the number of filled void cells grew along with the number of acquired candidate compounds using either sequential selection or the compound acquisition protocol. We also wanted to point out that the choice of bin size would affect the diversity assessment as discussed above. The result illustrated the diversity increment by importing candidate compounds and helped to determine the optimal number of acquired compounds in a specific binning procedure.

Taken together, the compound acquisition and prioritization algorithm using BCUT descriptors is capable of retrieving compounds from candidate compound collections to increase structural diversity of an existing compound dataset. Currently, this method is being used for prioritizing to-be-synthesized combinatorial libraries in order to enhance the diversity-oriented library design and synthesis; however, it could also be view as a necessary complement to the existing techniques for building quality chemical libraries for HTS/HCS and virtual screening.

## 7.5 CASE STUDY

This section describes the application of BCUT descriptors and derived chemistry-space-oriented selection rule, in order to determine whether the newly synthesized compound library contributes any



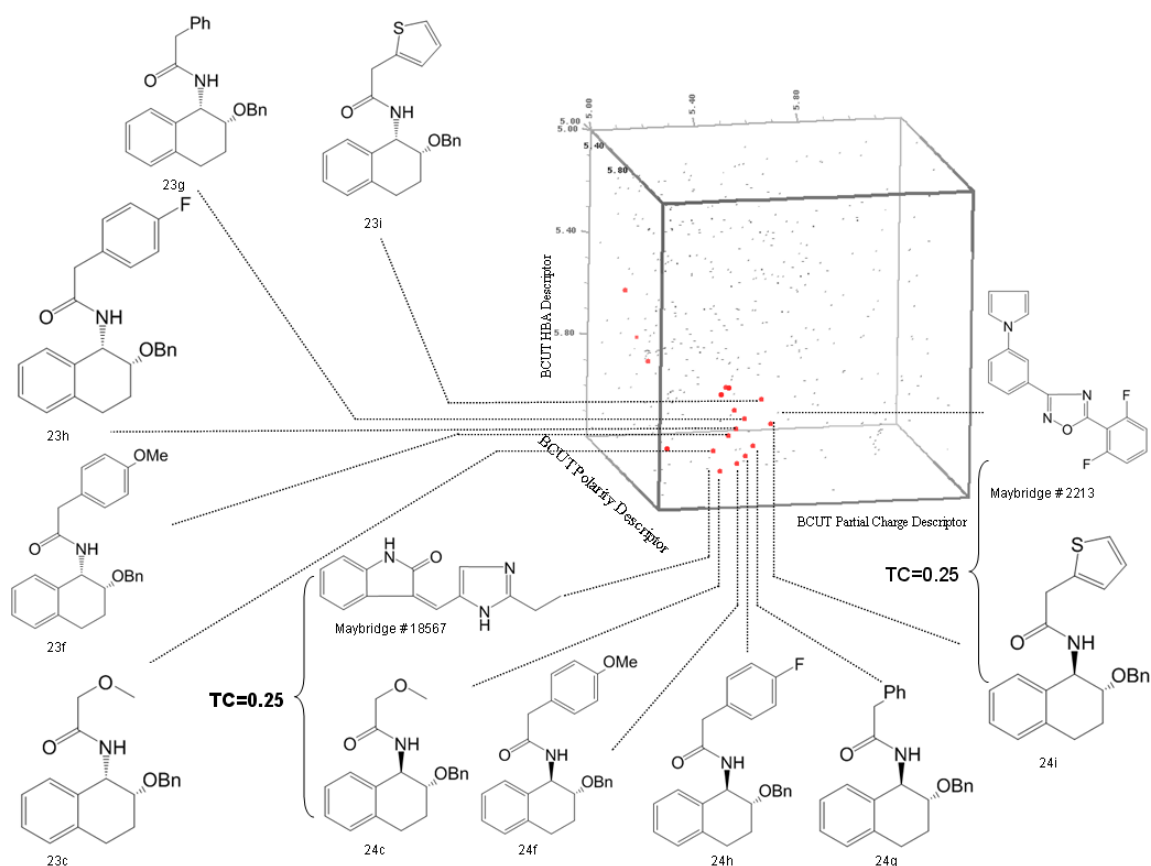
chemical diversity value to the existing NIH small molecular repository (SMR). The chemical structures of the synthesized compounds and synthetic route can be found in the following publication:

Construction of a Bicyclic  $\beta$ -Benzyloxy and  $\beta$ -Hydroxy Amide Library through a Multicomponent Cyclization Reaction Li Zhang, Qing Xiao, **Chao Ma**, Xiang-Qun Xie, Paul E. Floreancig *Journal of Combinatorial Chemistry* **2009** *11* (4), 640-644

The established 3D chemistry-space BCUT metrics calculation and 2D fingerprint similarity calculation approaches are applied to analyze structural diversity of 43 compounds. All computational works were performed using a Linux PC/dual-core dual-CPU Xeon-based HPCC 30-processor Dell cluster, loaded with Tripos Sybyl molecular modeling package (version 8.0). 330K SMR compound library was downloaded from NIH Small Molecular Repository (SMR) <sup>2</sup>. 57K Maybridge screening collection was downloaded from Maybridge website <sup>3</sup>. As all the 43 synthesized compounds were chiral molecules, molecular conformation was considered as one of the key molecular features for diversity analysis. Tripos CONCORD <sup>1</sup> was used to generate 3D conformation for each library.

Multi-dimensional chemistry space coordinates were calculated for each compound in the target library according to four main classes of atomic properties including atomic Gasteiger-Huckel charge, polarity, H-bond donor (HBD) and H-bond acceptor (HBA) attributes. BCUT descriptors were generated for both SMR and Maybridge libraries using DiverseSolutions <sup>212</sup>. For visualization and dimension reduction purpose, the chemistry space was defined by the best three BCUT descriptors and raw descriptor values were rescaled to range from 0 to 10. A diversity analysis was performed by cell statistics under the chemistry space.

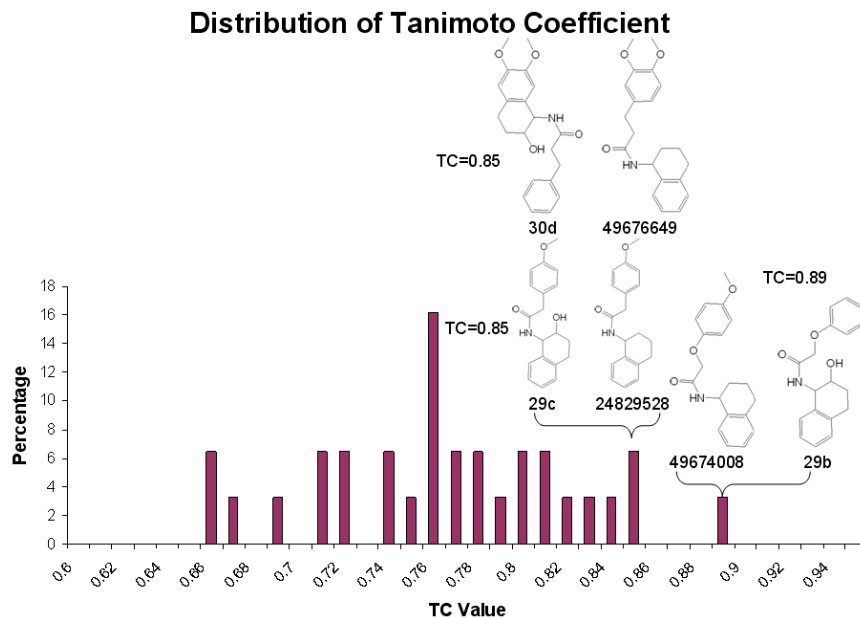
To further evaluate the similarity properties of the new 43 sets compound library in comparing with the NIH SMR library, pair-wise Tanimoto coefficient (Tc) calculation was carried out based on 2D molecular fingerprint. The comparison of 43 sets versus SMR (330K) was done using the Tripos SELECTOR program. Molecular fingerprints were calculated using the standard Tripos UNITY 2D fingerprints and the data processes were carried out using Sybyl Molecular Spreadsheet <sup>213</sup>.



**Figure 7-8: 3D chemistry-space plots**

Plotting 43 synthesized compounds (red dots) in comparison with Maybridge chemical database (57K compounds, grey dots). Three axes were defined by atomic partial charge, H-Bond acceptor and polarity BCUT descriptors.

Figure 7-8 shows a 3D chemistry-space plot of the newly synthesized 43 compounds versus a Maybridge library. As shown in Figure 7-8, eight points representing 23c, 24c, 23f, 24f, 23h, 24h, 23g and 24g filled a void cell ranging approximately from 5.31 to 5.47 along the GH-charge axis, from 5.97 to 6.17 along the HBA axis and from 5.69 to 5.79 along the Polarity axis. To approximate the density of Maybridge library locally, the number of compounds was counted in a unit cubic cell which was centered at (5.39, 6.07, 5.74). The volume of void cell was 0.0033 and the local density of Maybridge library was 214, so  $214 \times 0.0033 \approx 0.7$  compound was expected to be found in the cell. None of Maybridge structures were observed in that cell and eight synthesized compounds filled the void region instead. This calculation showed that the size of the void cell filled by 8 new compounds was reasonable and the diversity increased by filling the void was also significant.



**Figure 7-9: A database similarity comparison of 43 compounds with SMR library**

Interestingly, Figure 7-8 also revealed that the chemistry-space coordinates of compound 23c, 23f, 23h, 23g and 23i followed almost the same pattern as those of 24c, 24f, 24h, 24g and 24i did. They were arranged in the same order and distributed over the space in a similar manner. The difference between scaffold 23 and 24 was the chirality. This indicated that the molecular conformation generated by Concord and BCUT 3D descriptors reflected the property of those compounds.

Figure 7-9 shows a distribution of Tanimoto coefficient ( $T_c$ ) for each of 43 compounds in comparison with its most similar compound in SMR database, showing a mean  $T_c$  values of 0.77 and a standard deviation of 0.06. As 2D molecular fingerprints could not reflect the chirality, the compounds 23a and 24a were indistinguishable under 2D fingerprint algorithm. Thus, there were totally 31 distinct 2D structures considered. It is generally accepted that Tanimoto coefficient value 0.85<sup>214</sup> is a threshold for UNITY 2D fingerprints. If the  $T_c$  value between a pair of compound is higher than 0.85, they are considered similar. 2D molecular fingerprint similarity calculations of two chemical libraries revealed that none of the compounds except three cases between two libraries possess a pair wise  $T_c$  value higher

than 0.85. Four newly synthesized compounds and their nearest neighbors by  $T_C$  were also illustrated in Figure 7-9. These results further confirmed that the newly synthesized compound library improved the molecular diversity of existing libraries. The combined 3D chemistry-space and 2D fingerprint pair-wised  $T_C$  similarity calculation approaches conclude that the newly synthesized compound library adds certain structural diversity value to the existing chemical libraries.

## 8 CONCLUSIONS AND FUTURE DIRECTIONS

The following conclusion may be drawn from the plot studies in this thesis:

**It may be feasible to derive quantitative structure-activity relationship (QSAR) or structure-property relationship (QSPR) from modern machine learning techniques and appropriate molecular descriptors.**

The novel ligand classification technique, LiCABEDS reported in Chapter 4, was successfully applied to the prediction of ligand functionality, selectivity and blood-brain-barrier passage. Its performance was validated and compared with other data mining techniques, including Naive Bayes classifier, Tree and support vector machine. LiCABEDS provides an alternative option to the mainstream QSAR/QSPR methods. For example, CoMFA was reported to study the structure requirement for 5-HT<sub>1A</sub> agonists and antagonists using a small set of compounds. On the other hand, the LiCABEDS models were derived from hundreds of compounds in cheminformatics database, and the algorithm design guaranteed a prediction throughput of a few thousand compounds per second. The implementation of LiCABEDS further demonstrated that QSAR/QSPR modeling could be as straightforward as similarity calculation.

**Molecular fingerprints exhibit decent performance in QSAR/QSPR studies.**

This dissertation reviews molecular descriptors in many categories, including frequently referred descriptors, for example, the number of H-bond donors. Many descriptors have been proven effective in predicting physicochemical properties, but variable selection and multivariate analysis are generally complicated in QSAR/QSPR modeling. Hybrid descriptors, including a diverse set of descriptors, seem to be the favored choice in publications. In this dissertation, MACCS, FP2, Unity, PubChem and Molprint 2D fingerprints revealed their relevance to molecular property and bioactivity prediction. According to scientific papers and experiments described in this thesis, molecular fingerprints are good starting point for QSAR/QSPR modeling even if they may be not the best.

**Robustness, interpretability, and simple parameter tuning are major advantages of LiCABEDS.**

The valuable attributes of LiCABEDS were exemplified through case studies. The limited variance in its consisted components suggested the robustness of LiCABEDS for prospective predictions. Furthermore,

LiCABEDS models were interpreted by examining highly weighted decision stumps. Last but not least, the simple parameterization was shown to be another advantage of LiCABEDS. It was free of structure alignment and time-consuming parameter optimization. A large number of iteration steps produced models that were close to the optimal. Modeling using LiCABEDS did not involve much subjectiveness.

**Non-linear classifiers with appropriate parameters may outperform the linear classifiers. The distance or similarity between any pair of queries may be sufficient to develop a robust classification model, without explicit representation of explanatory variables.**

The use of non-linear kernel functions in support vector machine (Chapter 5) outperformed default linear kernel according to benchmarked dataset. This result indicated that the contribution of compound fragments to biological properties was not linear. However, the choice of kernel function and parameters was computationally expensive, but quite critical to model robustness. Tanimoto kernel and RBF kernel were generally the default non-linear choice, supported by recent publications. Tanimoto kernel is free of kernel parameter, while RBF kernel function is not.

**GPUs show significant performance advantage over CPUs regarding computationally expensive cheminformatics tasks, but extra attention must be paid to algorithm design and implementation.**

Computation power still remains as limitation for virtual screening. The GPU-accelerated chemical similarity calculation proposed a very efficient algorithm to calculate Tanimoto coefficients based on molecular fingerprints represented in dense array formats. Nowadays, famous molecular mechanics software, CHARMM, has already integrated GPU implementation in order to accelerate molecular dynamics simulation, which levels up molecular modeling to a new time frame. In the near future, it is expected to have more molecular modeling software targeting at GPU platform.

Despite the novel approaches proposed in this thesis, mapping chemical structures to bioactivity space still remains as a significant challenge to both information science and artificial intelligence. First, extensive variability in bioassays and frequent false positives in HTS experiments demand robust and specific learning algorithms to catch meaningful information and predictive features from assorted data sources. Furthermore, learning algorithms are supposed to be capable of modeling numerous molecular properties and pharmacological properties, with the intention to insure the wide applicability in drug development projects, which may target at enzymes, GPCRs or other receptors. Finally, the developed predictive models are expected to be interpretable and chemically suggestive to bridge the gap between computer science and pharmaceutical science. Accordingly, the future directions of this thesis may include but not limited to:

**Developing effective ensemble learning algorithms and hypothesis selection methods for the predictions of ligand bioactivity and pharmacological properties**

Linear and non-linear ensemble methods will be developed for classification and regression purposes in order to predict various ligand properties. Medicinal chemists refine the structures of “lead” compounds according to a set of empirical hypotheses drawn from observations in most cases. This approach is frequently referred as SAR (structure-activity relationship). In SAR studies, researchers usually attempt to derive certain heuristic trends or assumptions from experimental data. However, these hypotheses are usually put into practice without systematic validation and quantification. The motivation of this aim is to model the hypothesis-driven SAR: each hypothesis  $i$ ,  $H_i$ ,  $H_i \in \mathcal{H}$ , is quantitatively correlated with certain property of interest,  $y$ , through function  $f$ , i.e.  $y = f(H_i(x))$ .  $\mathcal{H}$  represents the possible hypothesis space and  $x$  represents chemical structure. However, in ensemble-learning framework the property ( $y$ ) of any novel structure ( $x$ ) is determined by weighted hypothesis ensemble:  $y = f(\sum_i w_i H_i(x))$ .

### **Designing active learning strategies for iterative model refinement, together with retrospective and prospective model validation**

The dependence on training data restricts the applicability of supervised learning in many application domains. At the beginning stage of many drug design projects, the knowledge or annotated compounds may be not sufficient to develop supervised learning models. In this case, we propose active learning strategy to selectively acquire additional compound annotation through bioassays, and iteratively improve established prediction models. Compared to random compound annotation, this experimental design ensures early model convergence, but yields equivalently robust models. Most importantly, models are validated prospectively, which is considered as the most reliable way to assess generalization error.

### **Implementing the proposed algorithms targeting at high-performance computing architecture and providing intuitive graphical interface for medicinal chemists**

Millions of chemical structures are deposited into public cheminformatics database annually. Data mining the vast database and making predictions on enormous virtual compounds are sometimes questioned due to the limitation of computation power. The advent of CUDA, NVIDIA’s solution to GPGPU (General-purpose computing on graphics processing units), suggests an alternative path to scientific computing. Once implemented efficiently, arithmetic-intensive calculations can be accelerated up to 30 folds by GPUs, typically.

*The overall goal of this thesis is to present an objective high-through computational methodology for computer-aided drug design in order to enhance the productivity of the pharmaceutical industry. As a relatively new research field, QSAR/QSPR modeling using modern machine learning techniques is still not widely accepted by medicinal chemists and pharmacologists. Nevertheless, statistical inference and machine learning are proven to be helpful in many fields, and the pharmaceutical industry should not be an exception. Finally, the author would like to end this thesis by quoting prestigious statistician, George E. P. Box: "All Models Are Wrong But Some Are Useful." (George E. P. Box, 1979)*



## 9 BIBLIOGRAPHY

1. Adams, C. P.; Brantner, V. V. Estimating the cost of new drug development: is it really 802 million dollars? *Health Aff (Millwood)* **2006**, *25* (2), 420-8.
2. DiMasi, J. A. The value of improving the productivity of the drug development process: faster times and better decisions. *Pharmacoeconomics* **2002**, *20 Suppl 3*, 1-10.
3. DiMasi, J. A.; Hansen, R. W.; Grabowski, H. G. The price of innovation: new estimates of drug development costs. *J Health Econ* **2003**, *22* (2), 151-85.
4. Arrowsmith, J. Trial watch: Phase II failures: 2008–2010. *Nat Rev Drug Discov* **2011**, *10* (5), 328-329.
5. Arrowsmith, J. Trial watch: Phase III and submission failures: 2007–2010. *Nat Rev Drug Discov* **2011**, *10* (2), 87-87.
6. Hann, M. M.; Oprea, T. I. Pursuing the leadlikeness concept in pharmaceutical research. *Current Opinion in Chemical Biology* **2004**, *8* (3), 255-263.
7. O'Driscoll, C., A Virtual Space Odyssey. In *The 4th Horizon Symposium*  
<http://www.nature.com/horizon/chemicalspace/background/pdf/odyssey.pdf> U.S.A., 2004.
8. McInnes, C. Virtual screening strategies in drug discovery. *Current Opinion in Chemical Biology* **2007**, *11* (5), 494-502.
9. Sheridan, R. P.; Kearsley, S. K. Why do we need so many chemical similarity search methods? *Drug Discovery Today* **2002**, *7* (17), 903.
10. Wilton, D. J.; Harrison, R. F.; Willett, P.; Delaney, J.; Lawson, K.; Mullier, G. Virtual Screening Using Binary Kernel Discrimination: Analysis of Pesticide Data. *Journal of Chemical Information and Modeling* **2006**, *46* (2), 471.
11. X-Q, X. Exploiting PubChem for Virtual Screening. *Expert Opinion for Drug Discovery (in press)* **2010**, *5*, 1-16.
12. Clark, D. E.; Pickett, S. D. Computational methods for the prediction of "drug-likeness". *Drug Discovery Today* **2000**, *5* (2), 49.
13. Cramer, R. D.; Patterson, D. E.; Bunce, J. D. Comparative molecular field analysis (CoMFA). 1. Effect of shape on binding of steroids to carrier proteins. *Journal of the American Chemical Society* **1988**, *110* (18), 5959.
14. Martin, Y. C., 3D QSAR: Current State, Scope, and Limitations. In *3D QSAR in Drug Design*, 2002; p 3.
15. Mitchell, T. M. *Machine Learning*. McGraw-Hill Science Engineering: 1997.
16. Baker, J. A.; Hirst, J. D. Molecular Dynamics Simulations Using Graphics Processing Units. *Molecular Informatics* **2011**, *30* (6-7), 498-504.

17. NVIDIA CUDA ZONE. [http://www.nvidia.com/object/cuda\\_home\\_new.html](http://www.nvidia.com/object/cuda_home_new.html) (accessed Dec 10, 2010).
18. Johnson, S. R. The Trouble with QSAR (or How I Learned To Stop Worrying and Embrace Fallacy). *Journal of Chemical Information and Modeling* **2007**, *48* (1), 25-26.
19. Dudek, A. Z.; Arodz, T.; Galvez, J. Computational methods in developing quantitative structure-activity relationships (QSAR): a review. *Comb Chem High Throughput Screen* **2006**, *9* (3), 213-28.
20. Bishop, C. M. *Pattern Recognition and Machine Learning*. 1st ed. 2006 ed.; Springer: 2006.
21. Hastie, T.; Rosset, S.; Tibshirani, R.; Zhu, J. The Entire Regularization Path for the Support Vector Machine. *J. Mach. Learn. Res.* **2004**, *5*, 1391-1415.
22. Wold, S.; Eriksson, L.; Clementi, S., Statistical Validation of QSAR Results. In *Chemometric Methods in Molecular Design*, Wiley-VCH Verlag GmbH: 2008; pp 309-338.
23. Wasserman, L. Lecture Notes for 36-707 Linear Regression. *Unpublished* **2008**.
24. Thaimattam, R.; Daga, P.; Rajjak, S. A.; Banerjee, R.; Iqbal, J. 3D-QSAR CoMFA, CoMSIA studies on substituted ureas as Raf-1 kinase inhibitors and its confirmation with structure-based studies. *Bioorganic & Medicinal Chemistry* **2004**, *12* (24), 6415-6425.
25. Ravichandran, V.; Agrawal, R. K. Predicting anti-HIV activity of PETT derivatives: CoMFA approach. *Bioorganic & Medicinal Chemistry Letters* **2007**, *17* (8), 2197-2202.
26. Zernov, V. V.; Balakin, K. V.; Ivaschenko, A. A.; Savchuk, N. P.; Pletnev, I. V. Drug Discovery Using Support Vector Machines. The Case Studies of Drug-likeness, Agrochemical-likeness, and Enzyme Inhibition Predictions. *Journal of Chemical Information and Computer Sciences* **2003**, *43* (6), 2048-2056.
27. Cross, J. B.; Thompson, D. C.; Rai, B. K.; Baber, J. C.; Fan, K. Y.; Hu, Y.; Humblet, C. Comparison of Several Molecular Docking Programs: Pose Prediction and Virtual Screening Accuracy. *Journal of Chemical Information and Modeling* **2009**, *49* (6), 1455-1474.
28. Truchon, J.-F.; Bayly, C. I. Evaluating Virtual Screening Methods: Good and Bad Metrics for the "Early Recognition" Problem. *Journal of Chemical Information and Modeling* **2007**, *47* (2), 488-508.
29. Todeschini, R.; Consonni, V. *Handbook of Molecular Descriptors*. Wiley-VCH, Weinheim (Germany): 2000.
30. Schultz, H. P.; Schultz, E. B.; Schultz, T. P. Topological organic chemistry. 4. Graph theory, matrix permanents, and topological indices of alkanes. *Journal of Chemical Information and Computer Sciences* **1992**, *32* (1), 69-72.
31. Ruecker, G.; Ruecker, C. Counts of all walks as atomic and molecular descriptors. *Journal of Chemical Information and Computer Sciences* **1993**, *33* (5), 683-695.
32. Burden, F. R. Molecular identification number for substructure searches. *Journal of Chemical Information and Computer Sciences* **1989**, *29* (3), 225-227.
33. Gasteiger, J.; Sadowski, J.; Schuur, J.; Selzer, P.; Steinhauer, L.; Steinhauer, V. Chemical Information in 3D Space. *Journal of Chemical Information and Computer Sciences* **1996**, *36* (5), 1030-1037.
34. Cruciani, G.; Crivori, P.; Carrupt, P. A.; Testa, B. Molecular fields in quantitative structure-permeation relationships: the VolSurf approach. *Journal of Molecular Structure: THEOCHEM* **2000**, *503* (1-2), 17-30.
35. Senese, C. L.; Duca, J.; Pan, D.; Hopfinger, A. J.; Tseng, Y. J. 4D-Fingerprints, Universal QSAR and QSPR Descriptors. *Journal of Chemical Information and Computer Sciences* **2004**, *44* (5), 1526-1539.

36. Waterbeemd, H. v. d.; Testa, B. *Drug Bioavailability: Estimation of Solubility, Permeability, Absorption and Bioavailability*. Wiley-VCH: December 2008.
37. Wong, O.; McKeown, R. H. Substituent effects on partition coefficients of barbituric acids. *Journal of Pharmaceutical Sciences* **1988**, *77* (11), 926-932.
38. Luan, F.; Ma, W.; Zhang, H.; Zhang, X.; Liu, M.; Hu, Z.; Fan, B. Prediction of pKa for Neutral and Basic Drugs Based on Radial Basis Function Neural Networks and the Heuristic Method *Pharmaceutical Research* **2005**, *22* (9), 1454-1460.
39. Eroglu, E. Some QSAR Studies for a Group of Sulfonamide Schiff Base as Carbonic Anhydrase CA II Inhibitors. *International Journal of Molecular Sciences* **2008**, *9* (2), 181-197.
40. Gharagheizi, F. QSPR Studies for Solubility Parameter by Means of Genetic Algorithm-Based Multivariate Linear Regression and Generalized Regression Neural Network. *QSAR & Combinatorial Science* **2008**, *27* (2), 165-170.
41. Garkani-Nejad, Z.; Karlovits, M.; Demuth, W.; Stimpfl, T.; Vycudilik, W.; Jalali-Heravi, M.; Varmuza, K. Prediction of gas chromatographic retention indices of a diverse set of toxicologically relevant compounds. *Journal of Chromatography A* **2004**, *1028* (2), 287-295.
42. Yao, X.; Liu, M.; Zhang, X.; Hu, Z.; Fan, B. Radial basis function network-based quantitative structure–property relationship for the prediction of Henry’s law constant. *Analytica Chimica Acta* **2002**, *462* (1), 101-117.
43. Liu, H. X.; Xue, C. X.; Zhang, R. S.; Yao, X. J.; Liu, M. C.; Hu, Z. D.; Fan, B. T. Quantitative prediction of logk of peptides in high-performance liquid chromatography based on molecular descriptors by using the heuristic method and support vector machine. *J Chem Inf Comput Sci* **2004**, *44* (6), 1979-86.
44. Ivanciuc, O.; Ivanciuc, T.; Balaban, A. T. Quantitative structure-property relationship study of normal boiling points for halogen-/ oxygen-/ sulfur-containing organic compounds using the CODESSA program. *Tetrahedron* **1998**, *54* (31), 9129-9142.
45. Li, X.; Zhang, G.; Dong, J.; Zhou, X.; Yan, X.; Luo, M. Estimation of critical micelle concentration of anionic surfactants with QSPR approach. *Journal of Molecular Structure: THEOCHEM* **2004**, *710* (1-3), 119-126.
46. Li, J.; Liu, H.; Yao, X.; Liu, M.; Hu, Z.; Fan, B. Structure–activity relationship study of oxindole-based inhibitors of cyclin-dependent kinases based on least-squares support vector machines. *Analytica Chimica Acta* **2007**, *581* (2), 333-342.
47. Hou, T. J.; Zhang, W.; Xia, K.; Qiao, X. B.; Xu, X. J. ADME Evaluation in Drug Discovery. 5. Correlation of Caco-2 Permeation with Simple Molecular Properties. *Journal of Chemical Information and Computer Sciences* **2004**, *44* (5), 1585-1600.
48. Hemmateenejad, B.; Safarpour, M. A.; Miri, R.; Nesari, N. Toward an Optimal Procedure for PC-ANN Model Building: Prediction of the Carcinogenic Activity of a Large Set of Drugs. *Journal of Chemical Information and Modeling* **2004**, *45* (1), 190-199.
49. Talevi, A.; Goodarzi, M.; Ortiz, E. V.; Duchowicz, P. R.; Bellera, C. L.; Pesce, G.; Castro, E. A.; Bruno-Blanch, L. E. Prediction of drug intestinal absorption by new linear and non-linear QSPR. *European Journal of Medicinal Chemistry* **2011**, *46* (1), 218-228.
50. Waterbeemd, H. v. d. *Chemometric Methods in Molecular Design (Methods and Principles in Medicinal Chemistry)* Wiley-VCH 1995; p 359.
51. Katritzky, A. R.; Mu, L.; Lobanov, V. S.; Karelson, M. Correlation of Boiling Points with Molecular Structure. 1. A Training Set of 298 Diverse Organics and a Test Set of 9 Simple Inorganics. *The Journal of Physical Chemistry* **1996**, *100* (24), 10400-10407.

52. de Bruijn, J.; Hermens, J. Relationships Between Octanol/Water Partition Coefficients and Total Molecular Surface Area and Total Molecular Volume of Hydrophobic Organic Chemicals. *Quantitative Structure-Activity Relationships* **1990**, *9* (1), 11-21.
53. Silverman, B. D.; Platt, D. E. Comparative Molecular Moment Analysis (CoMMA): 3D-QSAR without Molecular Superposition. *Journal of Medicinal Chemistry* **1996**, *39* (11), 2129-2140.
54. Mattioni, B. E.; Jurs, P. C. Development of Quantitative Structure-Activity Relationship and Classification Models for a Set of Carbonic Anhydrase Inhibitors. *Journal of Chemical Information and Computer Sciences* **2001**, *42* (1), 94-102.
55. Todeschini, R.; Vighi, M.; Provenzani, R.; Finizio, A.; Gramatica, P. Modeling and prediction by using whim descriptors in QSAR studies: toxicity of heterogeneous chemicals on *Daphnia magna*. *Chemosphere* **1996**, *32* (8), 1527-1545.
56. Bravi, G.; Wikel, J. H. Application of MS-WHIM Descriptors: 3. Prediction of Molecular Properties. *Quantitative Structure-Activity Relationships* **2000**, *19* (1), 39-49.
57. Randic, M.; Zupan, J. On interpretation of well-known topological indices. *J Chem Inf Comput Sci* **2001**, *41* (3), 550-60.
58. Pearlman, R. S. Diverse Solutions User's Manual. **1997**, 1-44.
59. Pearlman, R. S.; Smith, K. M. Metric validation and the receptor-relevant subspace concept. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 28-35.
60. Pearlman, R. S.; Smith, K. M. Novel Software Tools for Chemical Diversity. *Perspect. Drug Discovery Des.* **1998**, *9-11*, 339-353.
61. Burden, F. R. Molecular Identification Number for Substructure Searches. *J. Chem. Inf. Comput. Sci* **1989**, *29*, 225-227.
62. Stanton, D. T. Evaluation and Use of BCUT Descriptors in QSAR and QSPR Studies. *J. Chem. Inf. Comput. Sci.* **1999**, *39* (1), 11-20.
63. Pirard, B.; Pickett, S. D. Classification of Kinase Inhibitors Using BCUT Descriptors. *J. Chem. Inf. Comput. Sci.* **2000**, *40* (6), 1431-1440.
64. Gao, H. Application of BCUT metrics and genetic algorithm in binary QSAR analysis. *J. Chem. Inf. Comput. Sci.* **2001**, *41* (2), 402-7.
65. Xie, X. Q.; Chen, J. Z. Data mining a small molecule drug screening representative subset from NIH PubChem. *J. Chem. Inf. Model.* **2008**, *48* (3), 465-75.
66. Zhang, L.; Xiao, Q.; Ma, C.; Xie, X.-Q.; Floreancig, P. E. Construction of a Bicyclic  $\beta$ -Benzyloxy and  $\beta$ -Hydroxy Amide Library through a Multicomponent Cyclization Reaction. *J. Comb. Chem.* **2009**, *11* (4), 640-644.
67. Albaugh, D. R.; Hall, L. M.; Hill, D. W.; Kertesz, T. M.; Parham, M.; Hall, L. H.; Grant, D. F. Prediction of HPLC Retention Index Using Artificial Neural Networks and IGroup E-State Indices. *Journal of Chemical Information and Modeling* **2009**, *49* (4), 788-799.
68. Lukovits, I. Decomposition of the Wiener topological index. Application to drug-receptor interactions. *Journal of the Chemical Society, Perkin Transactions 2* **1988**, (9).
69. Estrada, E.; Ivanciuc, O.; Gutman, I.; Gutierrez, A.; Rodriguez, L. Extended Wiener indices. A new set of descriptors for quantitative structure-property studies. *New Journal of Chemistry* **1998**, *22* (8).
70. Basak, S. C.; Mills, D. R.; Balaban, A. T.; Gute, B. D. Prediction of Mutagenicity of Aromatic and Heteroaromatic Amines from Structure: A Hierarchical QSAR Approach. *Journal of Chemical Information and Computer Sciences* **2000**, *41* (3), 671-678.

71. Estrada, E.; Rodríguez, L. Edge-Connectivity Indices in QSPR/QSAR Studies. 1. Comparison to Other Topological Indices in QSPR Studies. *Journal of Chemical Information and Computer Sciences* **1999**, *39* (6), 1037-1041.
72. Karelson, M.; Lobanov, V. S.; Katritzky, A. R. Quantum-Chemical Descriptors in QSAR/QSPR Studies. *Chem Rev* **1996**, *96* (3), 1027-1044.
73. Iyer, M.; Tseng, Y. J.; Senese, C. L.; Liu, J.; Hopfinger, A. J. Prediction and Mechanistic Interpretation of Human Oral Drug Absorption Using MI-QSAR Analysis. *Molecular Pharmaceutics* **2006**, *4* (2), 218-231.
74. Cocchi, M.; Menziani, M. C.; Fanelli, F.; de Benedetti, P. G. Theoretical quantitative structure-activity relationship analysis of congeneric and non-congeneric  $\alpha$ 1-adrenoceptor antagonists: a chemometric study. *Journal of Molecular Structure: THEOCHEM* **1995**, *331* (1-2), 79-93.
75. Tuppurainen, K.; Lötjönen, S.; Laatikainen, R.; Vartiainen, T.; Maran, U.; Strandberg, M.; Tamm, T. About the mutagenicity of chlorine-substituted furanones and halopropenals. A QSAR study using molecular orbital indices. *Mutation Research/Fundamental and Molecular Mechanisms of Mutagenesis* **1991**, *247* (1), 97-102.
76. Clare, B. W.; Supuran, C. T. Carbonic anhydrase activators. 3: Structure-activity correlations for a series of isozyme II activators. *Journal of Pharmaceutical Sciences* **1994**, *83* (6), 768-773.
77. Bender, A.; Mussa, H. Y.; Glen, R. C.; Reiling, S. Molecular Similarity Searching Using Atom Environments, Information-Based Feature Selection, and a Naïve Bayesian Classifier. *Journal of Chemical Information and Computer Sciences* **2003**, *44* (1), 170-178.
78. McGregor, M. J.; Muskal, S. M. Pharmacophore Fingerprinting. 1. Application to QSAR and Focused Library Design. *Journal of Chemical Information and Computer Sciences* **1999**, *39* (3), 569-574.
79. Heikamp, K.; Bajorath, J. r. Large-Scale Similarity Search Profiling of ChEMBL Compound Data Sets. *Journal of Chemical Information and Modeling* **2011**, *51* (8), 1831-1839.
80. Lee, A. C.; Shedden, K.; Rosania, G. R.; Crippen, G. M. Data Mining the NCI60 to Predict Generalized Cytotoxicity. *Journal of Chemical Information and Modeling* **2008**, *48* (7), 1379-1388.
81. Takaoka, Y.; Endo, Y.; Yamanobe, S.; Kakinuma, H.; Okubo, T.; Shimazaki, Y.; Ota, T.; Sumiya, S.; Yoshikawa, K. Development of a Method for Evaluating Drug-Likeness and Ease of Synthesis Using a Data Set in Which Compounds Are Assigned Scores Based on Chemists' Intuition. *Journal of Chemical Information and Computer Sciences* **2003**, *43* (4), 1269-1275.
82. Shen, J.; Cheng, F.; Xu, Y.; Li, W.; Tang, Y. Estimation of ADME Properties with Substructure Pattern Recognition. *Journal of Chemical Information and Modeling* **2010**, *50* (6), 1034-1041.
83. Geppert, H.; Horváth, T.; Gärtner, T.; Wrobel, S.; Bajorath, J. Support-Vector-Machine-Based Ranking Significantly Improves the Effectiveness of Similarity Searching Using 2D Fingerprints and Multiple Reference Compounds. *Journal of Chemical Information and Modeling* **2008**, *48* (4), 742-746.
84. Hert, J.; Keiser, M. J.; Irwin, J. J.; Oprea, T. I.; Shoichet, B. K. Quantifying the Relationships among Drug Classes. *Journal of Chemical Information and Modeling* **2008**, *48* (4), 755-765.
85. Brown, R. D.; Martin, Y. C. Use of Structure-Activity Data to Compare Structure-Based Clustering Methods and Descriptors for Use in Compound Selection. *Journal of Chemical Information and Computer Sciences* **1996**, *36* (3), 572-584.

86. Xia, X.; Maliski, E. G.; Gallant, P.; Rogers, D. Classification of Kinase Inhibitors Using a Bayesian Model. *Journal of Medicinal Chemistry* **2004**, *47* (18), 4463-4470.
87. Gillet, V. J.; Willett, P.; Bradshaw, J. The Effectiveness of Reactant Pools for Generating Structurally-Diverse Combinatorial Libraries. *Journal of Chemical Information and Computer Sciences* **1997**, *37* (4), 731-740.
88. Liu, R.; Zhou, D. Using Molecular Fingerprint as Descriptors in the QSPR Study of Lipophilicity. *Journal of Chemical Information and Modeling* **2008**, *48* (3), 542-549.
89. Zhou, D.; Alelyunas, Y.; Liu, R. Scores of Extended Connectivity Fingerprint as Descriptors in QSPR Study of Melting Point and Aqueous Solubility. *Journal of Chemical Information and Modeling* **2008**, *48* (5), 981-987.
90. Schattel, V.; Hinselmann, G.; Jahn, A.; Zell, A.; Laufer, S. Modeling and Benchmark Data Set for the Inhibition of c-Jun N-terminal Kinase-3. *Journal of Chemical Information and Modeling* **2011**, *51* (3), 670-679.
91. Nidhi; Glick, M.; Davies, J. W.; Jenkins, J. L. Prediction of Biological Targets for Compounds Using Multiple-Category Bayesian Models Trained on Chemogenomics Databases. *Journal of Chemical Information and Modeling* **2006**, *46* (3), 1124-1133.
92. Askjaer, S.; Langgard, M. Combining Pharmacophore Fingerprints and PLS-Discriminant Analysis for Virtual Screening and SAR Elucidation. *Journal of Chemical Information and Modeling* **2008**, *48* (3), 476-488.
93. Free, S. M.; Wilson, J. W. A Mathematical Contribution to Structure-Activity Studies. *Journal of Medicinal Chemistry* **1964**, *7* (4), 395-399.
94. Cronin, M. T. D.; Aptula, A. O.; Dearden, J. C.; Duffy, J. C.; Netzeva, T. I.; Patel, H.; Rowe, P. H.; Schultz, T. W.; Worth, A. P.; Voutzoulidis, K.; Schüürmann, G. Structure-Based Classification of Antibacterial Activity. *Journal of Chemical Information and Computer Sciences* **2002**, *42* (4), 869-878.
95. Embrechts, K. P. B. a. M. J. An Optimization Perspective on Kernel Partial Least Squares Regression.
96. Bender, A.; Mussa, H. Y.; Glen, R. C.; Reiling, S. Similarity Searching of Chemical Databases Using Atom Environment Descriptors (MOLPRINT 2D): Evaluation of Performance. *Journal of Chemical Information and Computer Sciences* **2004**, *44* (5), 1708-1718.
97. Sharma, A. K.; Sheikh, S.; Pelczer, I.; Levy, G. C. Classification and Clustering: Using Neural Networks. *Journal of Chemical Information and Computer Sciences* **1994**, *34* (5), 1130-1139.
98. Aoyama, T.; Suzuki, Y.; Ichikawa, H. Neural networks applied to quantitative structure-activity relationship analysis. *J Med Chem* **1990**, *33* (9), 2583-90.
99. Plewczynski, D.; Spieser, S. A.; Koch, U. Assessing different classification methods for virtual screening. *J Chem Inf Model* **2006**, *46* (3), 1098-106.
100. Walters, W. P.; Murcko, M. A. Prediction of [']drug-likeness'. *Advanced Drug Delivery Reviews* **2002**, *54* (3), 255.
101. Deconinck, E.; Zhang, M. H.; Coomans, D.; Vander Heyden, Y. Classification tree models for the prediction of blood-brain barrier passage of drugs. *J Chem Inf Model* **2006**, *46* (3), 1410-9.
102. Burbidge, R.; Trotter, M.; Buxton, B.; Holden, S. Drug design by machine learning: support vector machines for pharmaceutical data analysis. *Comput Chem* **2001**, *26* (1), 5-14.
103. Jorissen, R. N.; Gilson, M. K. Virtual screening of molecular databases using a support vector machine. *J Chem Inf Model* **2005**, *45* (3), 549-61.

104. Geppert, H.; Humrich, J.; Stumpfe, D.; Gartner, T.; Bajorath, J. Ligand prediction from protein sequence and small molecule information using support vector machines and fingerprint descriptors. *J Chem Inf Model* **2009**, *49* (4), 767-79.
105. Wassermann, A. M.; Geppert, H.; Bajorath, J. Searching for target-selective compounds using different combinations of multiclass support vector machine ranking methods, kernel functions, and fingerprint descriptors. *J Chem Inf Model* **2009**, *49* (3), 582-92.
106. Warmuth, M. K.; Liao, J.; Ratsch, G.; Mathieson, M.; Putta, S.; Lemmen, C. Active learning with support vector machines in the drug discovery process. *J Chem Inf Comput Sci* **2003**, *43* (2), 667-73.
107. Rathke, F.; Hansen, K.; Brefeld, U.; Müller, K.-R. StructRank: A New Approach for Ligand-Based Virtual Screening. *Journal of Chemical Information and Modeling* **2010**, *51* (1), 83-92.
108. Hsieh, J.-H.; Yin, S.; Wang, X. S.; Liu, S.; Dokholyan, N. V.; Tropsha, A. Cheminformatics Meets Molecular Mechanics: A Combined Application of Knowledge-Based Pose Scoring and Physical Force Field-Based Hit Scoring Functions Improves the Accuracy of Structure-Based Virtual Screening. *Journal of Chemical Information and Modeling* **2011**, *52* (1), 16-28.
109. Papa, E.; Pilutti, P.; Gramatica, P. Prediction of PAH mutagenicity in human cells by QSAR classification. *SAR and QSAR in Environmental Research* **2008**, *19* (1), 115 - 127.
110. Grover, I. I.; Singh, I. I.; Bakshi, I. I. Quantitative structure-property relationships in pharmaceutical research - Part 2. *Pharm Sci Technolo Today* **2000**, *3* (2), 50-57.
111. Grover, M.; Singh, B.; Bakshi, M.; Singh, S. Quantitative structure-property relationships in pharmaceutical research - Part 1. *Pharmaceutical Science & Technology Today* **2000**, *3* (1), 28.
112. Wassermann, A. M.; Geppert, H.; Bajorath, J. Searching for Target-Selective Compounds Using Different Combinations of Multiclass Support Vector Machine Ranking Methods, Kernel Functions, and Fingerprint Descriptors. *Journal of Chemical Information and Modeling* **2009**, *49* (3), 582.
113. Liu, R.; Matheson, L. E. Comparative molecular field analysis combined with physicochemical parameters for prediction of polydimethylsiloxane membrane flux in isopropanol. *Pharm Res* **1994**, *11* (2), 257-66.
114. Van der Graaf, P. H. N., J. Van Schaick, E. A. Danhof, M. Multivariate quantitative structure-pharmacokinetic relationships (QSPKR) analysis of adenosine A1 receptor agonists in rat. *J Pharm Sci* **1999**, *88* (3), 306-12.
115. Chen, J. Z.; Han, X. W.; Liu, Q.; Makriyannis, A.; Wang, J.; Xie, X. Q. 3D-QSAR studies of arylpyrazole antagonists of cannabinoid receptor subtypes CB1 and CB2. A combined NMR and CoMFA approach. *J Med Chem* **2006**, *49* (2), 625-36.
116. Agarwal, A.; Taylor, E. W. 3-D QSAR for intrinsic activity of 5-HT1A receptor ligands by the method of comparative molecular field analysis. *Journal of Computational Chemistry* **1993**, *14* (2), 237-245.
117. Stumpfe, D.; Geppert, H.; Bajorath, J. Methods for Computer-Aided Chemical Biology. Part 3: Analysis of Structure Selectivity Relationships through Single- or Dual-Step Selectivity Searching and Bayesian Classification. *Chemical Biology & Drug Design* **2008**, *71* (6), 518-528.
118. Freund, Y.; Schapire, R. E. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *journal of computer and system sciences* **1997**, *55*, 119-139.
119. Freund, Y.; Schapire, R. E. A Short Introduction to Boosting. *Journal of Japanese Society for Artificial Intelligence* **1999**, *14* (5), 771.

120. Breiman, L.; Friedman, J. H.; Olshen, R. A.; Stone, C. J. Classification and Regression Trees, Wadsworth, Monterey **1984**.
121. Mitchell, T. M., Machine Learning. New York: McGraw-Hill: 1997.
122. Bishop, C. M., *Pattern Recognition and Machine Learning*. Springer: 2006.
123. Walters, W. P.; Murcko, M. A. Prediction of 'drug-likeness'. *Advanced Drug Delivery Reviews* **2002**, *54* (3), 255.
124. Bender, A.; Mussa, H. Y.; Glen, R. C.; Reiling, S. Similarity Searching of Chemical Databases Using Atom Environment Descriptors (MOLPRINT 2D): Evaluation of Performance. *Journal of Chemical Information and Computer Sciences* **2004**, *44* (5), 1708.
125. Okuno, Y.; Yang, J.; Taneishi, K.; Yabuuchi, H.; Tsujimoto, G. GLIDA: GPCR-ligand database for chemical genomic drug discovery. *Nucleic Acids Res* **2006**, *34* (Database issue), D673-7.
126. Okuno, Y.; Tamon, A.; Yabuuchi, H.; Nijima, S.; Minowa, Y.; Tonomura, K.; Kunimoto, R.; Feng, C. GLIDA: GPCR--ligand database for chemical genomics drug discovery--database and tools update. *Nucleic Acids Res* **2008**, *36* (Database issue), D907-12.
127. Hutchison, G. Open Babel: File Translation for Computational Chemistry and Nanoscience. *NNIN/CNF Fall Workshop* **2005**.
128. Steinbeck, C.; Han, Y.; Kuhn, S.; Horlacher, O.; Luttmann, E.; Willighagen, E. The Chemistry Development Kit (CDK): An Open-Source Java Library for Chemo- and Bioinformatics. *Journal of Chemical Information and Computer Sciences* **2003**, *43* (2), 493.
129. Steinbeck, C.; Hoppe, C.; Kuhn, S.; Floris, M.; Guha, R.; Willighagen, E. L. Recent developments of the chemistry development kit (CDK) - an open-source java library for chemo- and bioinformatics. *Curr Pharm Des* **2006**, *12* (17), 2111-20.
130. Ripley, B. D., Pattern Recognition and Neural Networks Chapter 7. Cambridge University Press: Cambridge, 1996.
131. Tripos, Sybyl Biopolymer [http://tripos.com/index.php?family=modules,SimplePage,sybyl\\_biopolymer](http://tripos.com/index.php?family=modules,SimplePage,sybyl_biopolymer) Sybyl version 7.1. Tripos, Inc. 2006.
132. Altschul, S. F.; Gish, W.; Miller, W.; Myers, E. W.; Lipman, D. J. Basic local alignment search tool. *J. Mol. Biol.* **1990**, *215*, 403-410.
133. Spring, D. R. Chemical genetics to chemical genomics: small molecules offer big insights. *Chem Soc Rev* **2005**, *34* (6), 472-82.
134. Williams, R.; Brown, K. Chemical genetics and genomics and drug discovery. Highlights from the Society for Medicines Research symposium held Thursday March 10, 2005, in London, United Kingdom. *Drug News Perspect* **2005**, *18* (4), 285-8.
135. Devane, W. A.; Dysarz, F. A., 3rd; Johnson, M. R.; Melvin, L. S.; Howlett, A. C. Determination and characterization of a cannabinoid receptor in rat brain. *Mol Pharmacol* **1988**, *34* (5), 605-13.
136. Munro, S.; Thomas, K. L.; Abu-Shaar, M. Molecular characterization of a peripheral receptor for cannabinoids. *Nature* **1993**, *365* (6441), 61-5.
137. Fernandez-Ruiz, J.; Pazos, M. R.; Garcia-Arencibia, M.; Sagredo, O.; Ramos, J. A. Role of CB2 receptors in neuroprotective effects of cannabinoids. *Mol Cell Endocrinol* **2008**, *286* (1-2 Suppl 1), S91-6.
138. McKallip, R. J.; Lombard, C.; Fisher, M.; Martin, B. R.; Ryu, S.; Grant, S.; Nagarkatti, P. S.; Nagarkatti, M. Targeting CB2 cannabinoid receptors as a novel therapy to treat malignant lymphoblastic disease. *Blood* **2002**, *100* (2), 627-34.



139. Ofek, O.; Karsak, M.; Leclerc, N.; Fogel, M.; Frenkel, B.; Wright, K.; Tam, J.; Attar-Namdar, M.; Kram, V.; Shohami, E.; Mechoulam, R.; Zimmer, A.; Bab, I. Peripheral cannabinoid receptor, CB2, regulates bone mass. *Proc Natl Acad Sci U S A* **2006**, *103* (3), 696-701.
140. Christensen, R.; Kristensen, P. K.; Bartels, E. M.; Bliddal, H.; Astrup, A. Efficacy and safety of the weight-loss drug rimonabant: a meta-analysis of randomised trials. *The Lancet* **370** (9600), 1706-1713.
141. Felder, C. C.; Joyce, K. E.; Briley, E. M.; Mansouri, J.; Mackie, K.; Blond, O.; Lai, Y.; Ma, A. L.; Mitchell, R. L. Comparison of the pharmacology and signal transduction of the human cannabinoid CB1 and CB2 receptors. *Mol Pharmacol* **1995**, *48* (3), 443-50.
142. Huffman, J. W. The search for selective ligands for the CB2 receptor. *Curr Pharm Des* **2000**, *6* (13), 1323-37.
143. Ashton, J. C.; Wright, J. L.; McPartland, J. M.; Tyndall, J. D. Cannabinoid CB1 and CB2 receptor ligand specificity and the development of CB2-selective agonists. *Curr Med Chem* **2008**, *15* (14), 1428-43.
144. Ma, C.; Wang, L.; Xie, X. Q. Ligand Classifier of Adaptively Boosting Ensemble Decision Stumps (LiCABEDS) and its application on modeling ligand functionality for 5HT-subtype GPCR families. *J Chem Inf Model* **2011**, *51* (3), 521-31.
145. Reggio, P. H.; Wang, T.; Brown, A. E.; Fleming, D. N.; Seltzman, H. H.; Griffin, G.; Pertwee, R. G.; Compton, D. R.; Abood, M. E.; Martin, B. R. Importance of the C-1 Substituent in Classical Cannabinoids to CB2 Receptor Selectivity: Synthesis and Characterization of a Series of O,2-Propano- $\Delta$ 8-tetrahydrocannabinol Analogs. *Journal of Medicinal Chemistry* **1997**, *40* (20), 3312-3318.
146. Menon, P.; Yin, G.; Smolock, E. M.; Zuscik, M. J.; Yan, C.; Berk, B. C. GPCR kinase 2 interacting protein 1 (GIT1) regulates osteoclast function and bone mass. *J Cell Physiol* **2010**, *225* (3), 777-85.
147. Chang, C.-C.; Lin, C.-J. LIBSVM : a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* **2011**, *2* (3), 27:1-27:27.
148. Bemis, G. W.; Murcko, M. A. The properties of known drugs. 1. Molecular frameworks. *J. Med. Chem* **1996**, *39* (15), 2887-2893.
149. Wan, J. P.; Chai, Y. F.; Wu, J. M.; Pan, Y. J. N,N'-(Phenylmethylene)diacetamide Analogues as Economical and Efficient Ligands in Copper-Catalyzed Arylation of Aromatic Nitrogen-Containing Heterocycles. *Synlett* **2008**, 3068-3072.
150. Bayer, A.; Maier, M. E. Synthesis of Enamides from Aldehydes and Amides. *ChemInform* **2004**, *35* (48), no-no.
151. Wang, L.; Ma, C.; Wipf, P.; Xie, X.-Q. Linear and Nonlinear Support Vector Machine for the Classification of Human 5-HT1A Ligand Functionality. *Molecular Informatics* **2012**, *31* (1), 85-95.
152. Zhao, Y. H.; Abraham, M. H.; Ibrahim, A.; Fish, P. V.; Cole, S.; Lewis, M. L.; de Groot, M. J.; Reynolds, D. P. Predicting Penetration Across the Blood-Brain Barrier from Simple Descriptors and Fragmentation Schemes. *Journal of Chemical Information and Modeling* **2006**, *47* (1), 170-175.
153. Kenakin, T. Inverse, protean, and ligand-selective agonism: matters of receptor conformation. *The FASEB Journal* **2001**, *15* (3), 598.

154. Chen, J. Z.; Han, X. W.; Liu, Q.; Makriyannis, A.; Wang, J.; Xie, X. Q. 3D-QSAR studies of arylpyrazole antagonists of cannabinoid receptor subtypes CB1 and CB2. A combined NMR and CoMFA approach. *J. Med. Chem* **2006**, *49* (2), 625-636.
155. Xie, X.-Q. S. Exploiting PubChem for virtual screening. *Expert Opinion on Drug Discovery* **2010**, *5* (12), 1205-1220.
156. Ito, H.; Halldin, C.; Farde, L. Localization of 5-HT1A receptors in the living human brain using [carbonyl-11C]WAY-100635: PET with anatomic standardization technique. *J Nucl Med* **1999**, *40* (1), 102-9.
157. Mewshaw, R. E.; Zhou, D.; Zhou, P.; Shi, X.; Hornby, G.; Spangler, T.; Scerni, R.; Smith, D.; Schechter, L. E.; Andree, T. H. Studies toward the discovery of the next generation of antidepressants. 3. Dual 5-HT1A and serotonin transporter affinity within a class of N-aryloxyethylindolylalkylamines. *J Med Chem* **2004**, *47* (15), 3823-42.
158. Parks, C. L.; Robinson, P. S.; Sibille, E.; Shenk, T.; Toth, M. Increased anxiety of mice lacking the serotonin1A receptor. *Proc Natl Acad Sci U S A* **1998**, *95* (18), 10734-9.
159. Kennett, G. A.; Dourish, C. T.; Curzon, G. Antidepressant-like action of 5-HT1A agonists and conventional antidepressants in an animal model of depression. *Eur J Pharmacol* **1987**, *134* (3), 265-74.
160. Yadav, V. K.; Ryu, J. H.; Suda, N.; Tanaka, K. F.; Gingrich, J. A.; Schutz, G.; Glorieux, F. H.; Chiang, C. Y.; Zajac, J. D.; Insogna, K. L.; Mann, J. J.; Hen, R.; Ducy, P.; Karsenty, G. Lrp5 controls bone formation by inhibiting serotonin synthesis in the duodenum. *Cell* **2008**, *135* (5), 825-37.
161. Schechter, L. E.; Smith, D. L.; Rosenzweig-Lipson, S.; Sukoff, S. J.; Dawson, L. A.; Marquis, K.; Jones, D.; Piesla, M.; Andree, T.; Nawoschik, S.; Harder, J. A.; Womack, M. D.; Buccafusco, J.; Terry, A. V.; Hoebel, B.; Rada, P.; Kelly, M.; Abou-Gharbia, M.; Barrett, J. E.; Childers, W. Lecozotan (SRA-333): a selective serotonin 1A receptor antagonist that enhances the stimulated release of glutamate and acetylcholine in the hippocampus and possesses cognitive-enhancing properties. *J Pharmacol Exp Ther* **2005**, *314* (3), 1274-89.
162. Sylte, I.; Bronowska, A.; Dahl, S. G. Ligand induced conformational states of the 5-HT(1A) receptor. *Eur J Pharmacol* **2001**, *416* (1-2), 33-41.
163. Han, L.; Wang, Y.; Bryant, S. H. Developing and validating predictive decision tree models from mining chemical structural fingerprints and high-throughput screening data in PubChem. *BMC Bioinformatics* **2008**, *9*, 401.
164. Cramer, R. D.; Clark, R. D.; Patterson, D. E.; Ferguson, A. M. Bioisosterism as a molecular diversity descriptor: steric fields of single "topomeric" conformers. *Journal of medicinal chemistry* **1996**, *39* (16), 3060-3069.
165. Cramer, R. D. Topomer CoMFA: a design methodology for rapid lead optimization. *Journal of medicinal chemistry* **2003**, *46* (3), 374-388.
166. Cramer, R. D.; Jilek, R. J.; Guessregen, S.; Clark, S. J.; Wendt, B.; Clark, R. D. "Lead Hopping". Validation of Topomer Similarity as a Superior Predictor of Similar Biological Activities. *Journal of medicinal chemistry* **2004**, *47* (27), 6777-6791.
167. Durant, J. L.; Leland, B. A.; Henry, D. R.; Nourse, J. G. Reoptimization of MDL keys for use in drug discovery. *Journal of chemical information and computer sciences* **2002**, *42* (6), 1273-1280.
168. Bender, A.; Mussa, H. Y.; Glen, R. C.; Reiling, S. Molecular similarity searching using atom environments, information-based feature selection, and a naive bayesian classifier. *J. Chem. Inf. Comput. Sci* **2004**, *44* (1), 170-178.

169. Bender, A.; Mussa, H. Y.; Glen, R. C.; Reiling, S. Similarity searching of chemical databases using atom environment descriptors: evaluation of performance. *J. Chem. Inf. Comput. Sci* **2004**, *44*, 1708-1718.
170. Hastie, T.; Rosset, S.; Tibshirani, R.; Zhu, J. The entire regularization path for the support vector machine. *The Journal of Machine Learning Research* **2004**, *5*, 1391-1415.
171. Schwartz, T. W.; Frimurer, T. M.; Holst, B.; Rosenkilde, M. M.; Elling, C. E. Molecular mechanism of 7TM receptor activation—a global toggle switch model. **2006**.
172. Xu, F.; Wu, H.; Katritch, V.; Han, G. W.; Jacobson, K. A.; Gao, Z. G.; Cherezov, V.; Stevens, R. C. Structure of an Agonist-Bound Human A2A Adenosine Receptor. *Science* **2011**, *332* (6027), 322.
173. Ma, C.; Wang, L.; Xie, X. Q. GPU Accelerated Chemical Similarity Calculation for Compound Library Comparison. *J. Chem. Inf. Model.* **2011**, *51* (7), 1521–1527.
174. Wassermann, A. M.; Geppert, H.; Bajorath, J. Searching for Target-Selective Compounds Using Different Combinations of Multiclass Support Vector Machine Ranking Methods, Kernel Functions, and Fingerprint Descriptors. *Journal of Chemical Information and Modeling* **2009**, *49* (3), 582-592.
175. Cramer, R. D.; Jilek, R. J.; Andrews, K. M. Dbtop: topomer similarity searching of conventional structure databases. *Journal of Molecular Graphics and Modelling* **2002**, *20* (6), 447-462.
176. Brown, R. D.; Martin, Y. C. The information content of 2D and 3D structural descriptors relevant to ligand-receptor binding. *J. Chem. Inf. Comput. Sci* **1997**, *37* (1), 1-9.
177. Andrews, K. M.; Cramer, R. D. Toward general methods of targeted library design: topomer shape similarity searching with diverse structures as queries. *J Med Chem* **2000**, *43* (9), 1723-40.
178. Guha, R.; Van Drie, J. H. Structure- Activity Landscape Index: Identifying and Quantifying Activity Cliffs. *J. Chem. Inf. Model* **2008**, *48* (3), 646-658.
179. Mattera, D.; Haykin, S. In *Support vector machines for dynamic reconstruction of a chaotic system*, MIT Press: 1999; pp 211-241.
180. Harper, G.; Pickett, S. D. Methods for mining HTS data. *Drug Discov Today* **2006**, *11* (15-16), 694-9.
181. Nikolova, N.; Jaworska, J. Approaches to Measure Chemical Similarity – a Review. *QSAR Comb. Sci.* **2003**, *22* (9-10), 1006-1026.
182. Nilakantan, R.; Bauman, N.; Haraki, K. S. Database diversity assessment: New ideas, concepts, and tools. *J. Comput.-Aided Mol. Des.* **1997**, *11* (5), 447-452.
183. Shemetulskis, N. E.; Weininger, D.; Blankley, C. J.; Yang, J. J.; Humblet, C. Stigmata: An Algorithm To Determine Structural Commonalities in Diverse Datasets. *J. Chem. Inf. Comput. Sci.* **1996**, *36* (4), 862-871.
184. Boyd, S. M.; Beverley, M.; Norskov, L.; Hubbard, R. E. Characterising the geometric diversity of functional groups in chemical databases. *J. Comput.-Aided Mol. Des.* **1995**, *9* (5), 417-424.
185. Martin, E. J.; Blaney, J. M.; Siani, M. A.; Spellmeyer, D. C.; Wong, A. K.; Moos, W. H. Measuring Diversity: Experimental Design of Combinatorial Libraries for Drug Discovery. *J. Med. Chem.* **1995**, *38* (9), 1431-1436.
186. Cummins, D. J.; Andrews, C. W.; Bentley, J. A.; Cory, M. Molecular Diversity in Chemical Databases: Comparison of Medicinal Chemistry Knowledge Bases and Databases of Commercially Available Compounds. *J. Chem. Inf. Comput. Sci.* **1996**, *36* (4), 750-763.

187. Singh, N.; Guha, R.; Giulianotti, M. A.; Pinilla, C.; Houghten, R. A.; Medina-Franco, J. L. Chemoinformatic Analysis of Combinatorial Libraries, Drugs, Natural Products, and Molecular Libraries Small Molecule Repository. *J. Chem. Inf. Comput. Sci.* **2009**, *49* (4), 1010-1024.
188. Bajorath, J. Integration of virtual and high-throughput screening. *Nat. Rev. Drug Discovery* **2002**, *1* (11), 882-94.
189. Martin, Y. C.; Kofron, J. L.; Traphagen, L. M. Do structurally similar molecules have similar biological activity? *J. Med. Chem.* **2002**, *45* (19), 4350-8.
190. *Tripos Bookshelf 8.0*, Tripos International: 1699 South Hanley Rd., St. Louis, Missouri, 63144, USA: 2007.
191. Wassermann, A. M.; Geppert, H.; Bajorath, J. Searching for Target-Selective Compounds Using Different Combinations of Multiclass Support Vector Machine Ranking Methods, Kernel Functions, and Fingerprint Descriptors. *J. Chem. Inf. Model.* **2009**, *49* (3), 582-592.
192. McGregor, M. J.; Pallai, P. V. Clustering of Large Databases of Compounds: Using the MDL "Keys" as Structural Descriptors. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 443-448.
193. Butina, D. Unsupervised Data Base Clustering Based on Daylight's Fingerprint and Tanimoto Similarity: A Fast and Automated Way To Cluster Small and Large Data Sets. *J. Chem. Inf. Comput. Sci.* **1999**, *39* (4), 747-750.
194. Wang, Y.; Xiao, J.; Suzek, T. O.; Zhang, J.; Wang, J.; Bryant, S. H. PubChem: a public information system for analyzing bioactivities of small molecules. *Nucleic Acids Res.* **2009**, *37* (Web Server issue), W623-33.
195. Xie, X.-Q. Exploiting PubChem for virtual screening. *Expert Opin. Drug Discovery* **2010**, *5* (12), 1205-1220.
196. Götz, A. W.; Wölfle, T.; Walker, R. C., Quantum Chemistry on Graphics Processing Units. In *Annu. Rep. Comput. Chem.*, Ralph, A. W., Ed. Elsevier: 2010; Vol. Volume 6, pp 21-35.
197. Xu, D.; Williamson, M. J.; Walker, R. C., Advancements in Molecular Dynamics Simulations of Biomolecules on Graphical Processing Units. In *Annu. Rep. Comput. Chem.*, Ralph, A. W., Ed. Elsevier: 2010; Vol. Volume 6, pp 2-19.
198. Haque, I. S.; Pande, V. S.; Walters, W. P. SIML: A Fast SIMD Algorithm for Calculating LINGO Chemical Similarities on GPUs and CPUs. *J. Chem. Inf. Model.* **2010**, *50* (4), 560-564.
199. Liao, Q.; Wang, J.; Webster, Y.; Watson, I. A. GPU accelerated support vector machines for mining high-throughput screening data. *J. Chem. Inf. Model.* **2009**, *49* (12), 2718-25.
200. Sachdeva, V.; Freimuth, D.; Mueller, C., Evaluating the Jaccard-Tanimoto Index on Multi-core Architectures. In *Computational Science – ICCS 2009*, Allen, G.; Nabrzyski, J.; Seidel, E.; van Albada, G.; Dongarra, J.; Sloot, P., Eds. Springer Berlin / Heidelberg: 2009; Vol. 5544, pp 944-953.
201. Gillet, V. J. New directions in library design and analysis. *Curr. Opin. Chem. Biol.* **2008**, *12* (3), 372-8.
202. Gorse, A. D. Diversity in medicinal chemistry space. *Curr. Top. Med. Chem.* **2006**, *6* (1), 3-18.
203. Downs, G. M.; Barnard, J. M., Clustering Methods and Their Uses in Computational Chemistry. In *Rev. Comput. Chem.*, Vol 18, Lipkowitz, K. B.; Boyd, D. B., Eds. John Wiley & Sons, Inc.: 2003; pp 1-40.
204. Lajiness, M. S. Dissimilarity-based compound selection techniques. *Perspect. Drug Discovery Des.* **1997**, *7-8*, 65-84.
205. Brown, R. D.; Martin, Y. C. Designing combinatorial library mixtures using a genetic algorithm. *J Med Chem* **1997**, *40* (15), 2304-13.

206. Chen, H.; Borjesson, U.; Engkvist, O.; Kogej, T.; Svensson, M. A.; Blomberg, N.; Weigelt, D.; Burrows, J. N.; Lange, T. ProSAR: a new methodology for combinatorial library design. *J Chem Inf Model* **2009**, *49* (3), 603-14.
207. Menard, P. R.; Mason, J. S.; Morize, I.; Bauerschmidt, S. Chemistry Space Metrics in Diversity Analysis, Library Design, and Compound Selection. *J. Chem. Inf. Comput. Sci.* **1998**, *38* (6), 1204-1213.
208. Schnur, D. Design and Diversity Analysis of Large Combinatorial Libraries Using Cell-Based Methods. *J. Chem. Inf. Comput. Sci.* **1999**, *39* (1), 36-45.
209. Stahura, F. L.; Bajorath, J. Partitioning methods for the identification of active molecules. *Curr. Med. Chem.* **2003**, *10* (8), 707-15.
210. Newman, D. J. Natural Products as Leads to Potential Drugs: An Old Process or the New Hope for Drug Discovery? *J. Med. Chem.* **2008**, *51* (9), 2589-2599.
211. Rishton, G. M. Natural products as a robust source of new drugs and drug leads: past successes and present day issues. *Am. J. Cardiol.* **2008**, *101* (10A), 43D-49D.
212. Paul R. Menard, J. S. M., Isabelle Morize, and Susanne Bauerschmidt Chemistry Space Metrics in Diversity Analysis Library Design, and Compound Selection. *J. Chem. Inf. Comput. Sci* **1998**, *38* (6), 1204-1213.
213. Tripos-vn.7.6 *Tripos Sybyl (version 7.2) molecular modeling software packages*. [www.tripos.com](http://www.tripos.com), TRIPOS, Associates, Inc.: St.Louis, MI63144.
214. Matter, H. Selecting optimally diverse compounds from structure databases: a validation study of two-dimensional and three-dimensional molecular descriptors. *J Med Chem* **1997**, *40* (8), 1219-29.