博士論文

A Study on Chemical Structure Generation Based on Inverse Quantitative Structure-Property Relationship/Quantitative Structure-Activity Relationship

(定量的構造物性相関/定量的構造活性相関モデルの逆解析を利用した化学構造創出に関する研究)

宮尾　知幸

(Tomoyuki Miyao)

To Chika and my parents.

ABSTRACT

In silico molecular design is becoming attractive because of increased computational power and accessibility. It ranges from first principles calculations to database screening. In the upper stream of molecular design, quantitative structure–property relationship/quantitative structure–activity relationship (QSPR/QSAR) analyses are used, which have been studied for more than 50 years. QSPR/QSAR modeling aims to establish a quantitative relationship between structural features of compounds (**x**) and their corresponding property or activity (y) using statistical approaches ($y = f(\mathbf{x})$). Their advantage of being able to treat millions of molecules in a reasonable time scale supports their positions in a molecular design workflow. Unfortunately, irrespective of how QSPR/QSAR methodologies are improved, the proposed molecules depend on the virtual libraries used for screening. In contrast to the screening approach, proposing chemical structures by inversely analyzing QSPR/QSAR models (inverse QSPR/QSAR) is logically possible. However, there has been little research conducted related to this topic because of the difficulty of both acquiring **x** information from y and retrieving chemical structures from that **x** information.

In this thesis, methodologies for chemical structure generation based on inverse QSPR/QSAR analysis have been studied. The goal of this study is to propose chemical structures exhibiting a specific property or activity based on a QSPR/QSAR model. This task was divided into two parts: retrieval of **x** information from y through the QSPR/QSAR model, and chemical structure generation based on **x** constraints. For the first part, cluster-wise multiple linear regression is proposed to capture a nonlinear relationship between **x** and y. This methodology also enables the applicability domain of a regression model to be taken into account. For the second part, a modified canonical construction-path method is

developed to generate chemical structures satisfying **x** constraints. Finally, these two elements are integrated into a chemical structure generation system based on inverse QSPR/QSAR. The proposed system may build chemical structures based on inverse QSPR/QSAR *de novo*, leading to unexpected discoveries in molecular design and providing inspiration for chemists.

ACKNOWLEDGMENTS

My first and deepest gratitude goes to Prof. Dr. Kimito Funatsu, my supervisor. Without his support and guidance, this study would not have been completed. Despite the fact that I once parted from the scientific path into the financial world, he welcomed me to joining his group and led me to the profound world of curiosity. I thank Dr. Hiromasa Kaneko, the assistant professor of our group, for our weekly scientific discussions and for giving me a number of valuable comments and suggestions to this study. I am also grateful to Dr. Matheus de Souza Escobar, a promising young scientist, also a good friend of mine, for proofreading my papers many times. His proofreading gave the necessary credibility this thesis required. Also I thank to all the members in our group. Their suggestion and advices at our monthly laboratory meeting inspired me to research profoundly. Apart from Tokyo, I would like to thank Prof. Dr. Gisbert Schneider at ETH Zurich, who provided me with an opportunity to join his group from May 2014 to March 2015. Not only he guided me in the right path in my research, he helped me tremendously during my first long stay in a foreign country. In his group, I am deeply grateful to Dr. Petra Schneider, Dr. Jan A. Hiss, Dr. Daniel Reker, and Dr. Jens Kunze, for supporting me and for giving valuable advice about my research. My sincere thank goes to Dr. Francesca Grisoni, a promising environmental scientist, also a good friend of mine. Bouldering with her, Mr. Aral C. Kaymaz, Daniel Reker, and Jens Kunze was one of my most enjoyable moments in Zurich. Moreover, immeasurable gratitude for the help and support during this journey are extended to following ex-colleagues of mine: Mr. Yuichi Murase, Mr. Kenichi Kasuga, Mr. Hiroki Nagasawa, and Mr. Masahiro Komatsu.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

LIST OF ABBREVIATIONS

AD	Applicability domain
ADMET	Absorption, distribution, metabolism, excretion, toxicity
AIC	Akaike information criterion
aR	Number of aromatic rings
BIC	Bayesian information criterion
CATS	Chemically advanced template search
CIC	Number of rings
CMC	Comprehensive medicinal chemistry
EM	Expectation-maximization
FDA	Food and Drug Administration
FOG	Fragment optimized growth
GA	Genetic algorithm
GCM	Group contribution method
GMM	Gaussian mixture model
GMMs/cMLR	Gaussian mixture models and cluster-wise multiple linear regression
GTM	Generative topographic mapping
MACCS	Molecular access system
MCD	Monotonous changing descriptors
MLR	Multiple linear regression
MTI	Molecular Topological Index
MW	Molecular weight
NMR	Nuclear magnetic resonance
nBR	Number of rotatable bonds
nHBA	Number of hydrogen bond acceptors
nHBD	Number of hydrogen bond donors
OLS	Ordinary least square
PAINS	Pan assay interference compounds
PCA	Principal component analysis
PLS	Partial least square
PPP	Potential pharmacophoric point
R^2	R-squared
QSAR	Quantitative structure-activity relationship
QSPR	Quantitative structure-activity relationships
RMSE	Root mean square error
RR	STDPs between aromatic rings
SMARTS	SMILES arbitrary target specification
SMILES	Simplified molecular-input line-entry system
TPSA	Topological polar surface area
VS	Virtual screening

CHAPTER 1 General Introduction

1-1 Molecular Design with Quantitative Structure-Property Relationship/Quantitative Structure-Activity Relationship

Quantitative structure-property relationship (QSPR) or quantitative structure-activity relationship (QSAR) is a way to find a quantitative relationship between compounds and their corresponding property or activity in a statistical manner. This property or activity is usually numerical and, therefore, can be represented as an objective variable: y. To treat compounds numerically, they are usually transformed into a set of descriptors (**x**), which are the abstract representation of a molecule. Once molecules are converted into descriptors, the remaining task is to make a regression model. Hence, latest machine learning techniques as well as classical statistical methodologies are employed for constructing mathematical correlations between **x** and y[1,2].

A simple way of making use of a QSPR/QSAR model for molecular design is to apply a QSPR/QSAR model to the molecules designed by chemists. Output from the model is examined in order to assess whether or not they exhibit the desired y. If they do not, chemists modify the structures. The modified structures are repeatedly tested by the model until they exhibit satisfactory y. Chemists may try to interpret the model so as to seek a way to improve designed molecules when it is impossible to obtain the desired y for them. This is usually conducted by statistical approaches, such as checking regression coefficients in a linear regression model, and sensitivity analysis in a non-linear regression model. Those heuristic approaches to obtain molecules exhibiting better y have been widely studied for practical molecular design[3,4].

Historically, group contribution methods (GCMs) for estimating physicochemical properties are the origin of today's sophisticated QSPR/QSAR methodologies, in particular QSPR analysis. A basic formula of a GCM is

$$\mathbf{y} = \sum_{i=1}^{n} \boldsymbol{c_i x_i} \qquad (1.1)$$

where y represents the property value of a query structure, x_i is a descriptor—the occurrence of group *i*—c_i is the corresponding contribution, and n is the total number of groups. The property value can be predicted by summing up each group contribution in a molecule. Many researches have been conducted and successfully applied for estimating physicochemical properties using linear GCMs, such as normal boiling point, normal freezing point, heat capacity at ideal gas conditions, dynamic viscosity at a given temperature, and so on[5,6]. Furthermore, linear GCMs are frequently used by medicinal chemists for estimating physicochemical properties in the field of drug design: aqueous solubility[7,8,9],

water/octanol partition coefficient[10,11,12], etc. One of the biggest advantages of linear GCMs compared to complicated machine learning techniques with fancy descriptors is that chemists can easily make use of feedback from a GCM model for improving the property of their designed molecules. When improving aqueous solubility of a molecule to a desired level, for example, one can consider inserting one group having positive contribution into that molecule.

Although designing chemical structures with linear GCMs is straightforward, they are not frequently used for practical applications of molecular design. There are two reasons for this: inadequate predictability and uncertainness of the model's applicability domain (AD)[13]. The former is obvious. GCMs usually show poorer predictability than that obtained with complicated regression models using various descriptors—due to its linear representation and the number of substructures as descriptors. The latter is a limitation of its usage when making prediction of novel compounds. AD limits the chemical space in the way that, only inside AD, predicted values produced by a regression model can be trusted. To make a larger AD, groups employed in a GCM should be thorough, otherwise the GCM may conceive a new structure that does not contain even a single group in the model. Consequently, it poorly predicts the desired property. It is, however, usually difficult to prepare a diversified training dataset for constructing GCMs. Furthermore, even when groups in the GCM are thorough, it should not be applied to a novel compound that is too different from any compounds in the training dataset used for model construction.

Therefore, it is important to consider AD when applying GCMs for predicting y values. In other words, training data for constructing GCMs should influence the way models predict y values. This situation is also applicable to any QSPR/QSAR models, not only to GCMs. AD depends on descriptors, regression methodologies and training data. Thus, these factors should be considered when evaluating AD of a model. The dataset used for model construction cannot be discarded once QSPR/QSAR model construction has been completed. Without considering AD, molecular design based on QSPR/QSAR might result in proposing compounds annotated with desired predictive values despite the fact that these values cannot be reliable.

Recent applications regarding small organic molecular design with QSPR/QSAR are biased toward the field of drug design. This is partly because when designing functional small organic molecules, drug is one of the most challenging and interesting targets that require multiple functions. drug design has to consider not only preferable activity to a target macromolecule, but also to exhibit drug-like properties (absorption, distribution, metabolism, excretion, and toxicity (ADMET))[14]. Consequently, many QSPR/QSAR analyses have aimed at designing lead compounds as their final goal. Although this thesis focuses on the development of methodologies for inverse QSPR/QSAR, and it could be applied to molecular design in various fields, demonstrated applications are in the field of drug design (lead design) instead of material design[15].

1-1-1 Virtual Screening (VS) and Evolutionary Algorithm (EA)-Based Molecular Design

Two heuristic strategies for molecular design: applying a QSPR/QSAR model to the molecules designed by chemists, and modifying molecules repeatedly in order to find

chemical structures exhibiting the desired y value, have been sophisticated in the field of computer-aided molecular design. The former becomes virtual screening (VS)[16] and the latter *de novo* molecular design[17].

VS filters out undesired structures as well as filters in desired ones based on certain criteria. In ligand-based drug design, these criteria may be divided into three categories: substructure count (or taboo list)[18,19], descriptor constraints (similarity criterion), and property or activity threshold based on QSPR/QSAR[20,21]. A simple flow chart of VS is depicted in **Figure 1-1**. The borders of the three categories are vague, in particular between substructure count and descriptor constraints, since the number of a substructure can be regarded as a descriptor. Success of VS depends on two factors: screening criteria and virtual library quality. For enhancing quality of filters, many studies for constructing QSPR models with high predictability have been conducted using machine learning techniques[22,23,24].

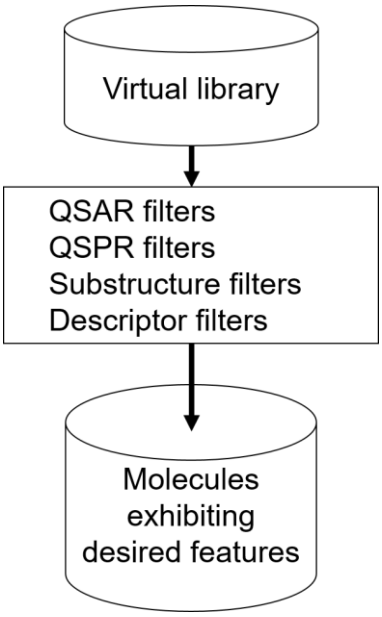


Figure 1-1 Workflow of VS with ligand-based filters

On the other hand, *de novo* molecular design means constructing chemical structures exhibiting desired properties and activities from scratch. In the field of drug discovery, a variety of methodologies for *de novo* design has been proposed—with and without target protein information, two-dimensional structure generation or three-dimensional structure generation, linking building blocks placed on the sets of specific coordinates or growing chemical structures by adding other building blocks[25]. Here, the author only focuses on a genetic algorithm-based (GA-based) approach that can be accompanied with QSPR/QSAR.

GA is a kind of evolutionary algorithms (EAs). Before describing what GA-based molecular design is, EA-based one is briefly explained since it is also successfully applied to *de novo* molecular design. EA-based design makes use of nature's biological mechanisms as a driving force for growing structures. EAs efficiently search (close to) optimal solutions of a multimodal objective function. The reason why EAs are employed for *de novo* design is that we cannot generate all chemical structures in chemical space[26]. The population of the chemical space is estimated to more than 10^{60} [27]. Exhaustive structure generation in chemical

space cannot be conceived of at the current computational power available. Therefore, EA-based design is promising, reaching local optima close to global optima with limited computational power. There have been several types of EAs-based *de novo* design besides GA, such as ant colony optimization algorithm[28] and evolutionary graph-based algorithm[29]. Among various types of EA-based molecular design *de novo*, GA-based one is widely used and the best popular methodology for *de novo* design judging from the number of publications[30,31,32]. GAs are based on Darwin's ideas about evolution by natural selection. In a GA, the goal is to find (close to) optimal solutions that maximize a fitting function. For effectively searching the solutions, solution candidates in one generation produce better candidates in the next generation by mimicking genetic operations: mutation and crossover. **Figure 1-2** shows a typical workflow of molecular design by a GA. By modifying structures in a current structure pool—mimicking the way of generating offspring in nature—next candidate structures are produced and can be evaluated by the same criteria as in VS. The biggest difference between VS and GA-based design is that GA-based design can produce novel structures, which are not in an initial pool, whereas VS can only select desired structures in an initial structure pool.

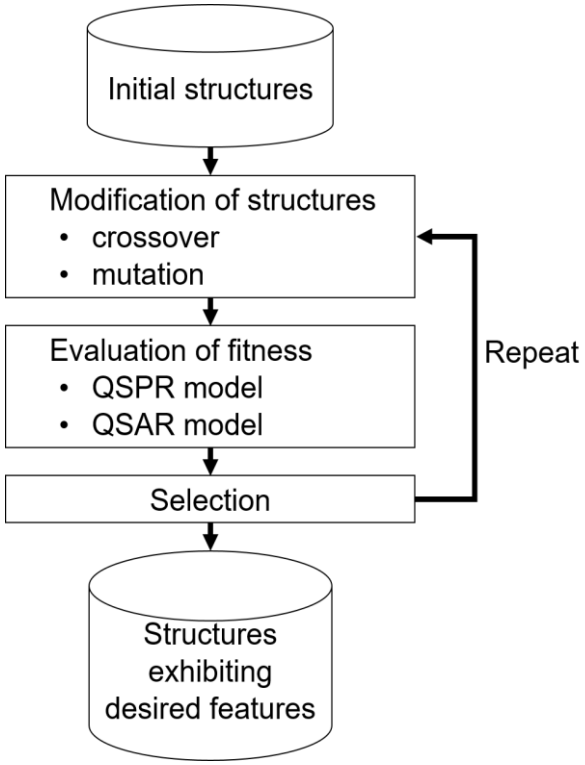


Figure 1-2 Workflow of a simple genetic algorithm-based molecular design

1-1-2 Challenges in Molecular Design with QSPR/QSAR

Although molecular design with QSPR/QSAR has been successful in both VS and GA-based design, there are some points that make these strategies insufficient: for VS, the limited

number of structures in virtual libraries, and for GA-based design, lacking the assurance of reaching global optima (**Table 1-1**).

A wide range of various filters can be adopted in VS: from molecular weight (MW) to quantum chemistry[33] in order to find molecules exhibiting desired features based on various QSPR/QSAR models. However, no matter how screening methodologies are improved, obtained results depend on the virtual library to be sought.

In GAs, local optima can usually be reached, although they seek for the global optimal solution and to avoid reaching local ones. A set of structures obtained by GA-based design may not be an optimal one, and we do not have any methodologies that are able to evaluate whether the set of solutions is really optimal or not. Besides its local optimal nature, three challenges may appear in GA-based design: initial structure dependency, generation of duplicate structures, and test of limited number of solution candidates. First, initial structure pool determines the final structures in GA-based design since descendants came to the pool by modification. Therefore, active compounds or existing drugs are frequently employed as initial structures, or are used for generating seed structures for the following repeating evolutionary procedure[34]. Second, generating duplicate structures reduces the number of solutions to be searched in solution space, leading to reduced efficiency of the algorithm. Duplication check procedure, such as the Morgan method[35], string match after converting chemical structures into canonical simplified molecular-input line-entry system (SMILES) format[36], is time consuming. Third, and this point is rather subjective, GA-based design does not restrict the generation condition precisely, and that it uses loose criteria for selecting descendants (e.g. to propose the best structure as a solution in the pool of generated structures). Otherwise it may not generate even one structure satisfying the restricted condition since the amount of searched space is too small in GA-based design compared to that in the entire chemical space. GA has a theoretical background on which a certain type of optimization problem can be solved efficiently (i.e. schema theorem)[37]. GA-based design, however, does not have it since genetic operation is applied to chemical structures, not to descriptors. It should be noted that one of the strongest points of GA-based molecular design lies in its flexibility. It can be applied to various types of problems for molecular design, and making use of not only QSPR/QSAR models but also other information such as three-dimensional ligand-protein docking models. It also takes the diversity of generated structures into account by adjusting an evaluation function[38]. Therefore, GA-based approach has been successfully adopted in various molecular design projects.

Table 1-1 Challenges in VS and GA-based approach

VS-based design	GA-based design
Impossible to generate novel structures that are not in virtual libraries	Local optima search in solution space
	Dependency on initial structures
	Generation of duplicate structures

1-2 Inverse QSPR/QSAR

1-2-1 Molecular Design by Inverse QSPR/QSAR

Inverse QSPR/QSAR is an another strategy for molecular design *de novo*. As long as filters in VS and evaluation functions in GA are based on QSPR/QSAR models, the idea that only chemical structures exhibiting preferable predicted y values by QSPR/QSAR models should be generated comes naturally. These chemical structures can be retrieved via descriptors, which is a result of analyzing QSPR/QSAR models inversely. Some researchers define inverse QSPR as retrieving chemical structures exhibiting a desired y value (e.g. GA-based design)[39]. In this thesis, inverse QSPR/QSAR is defined as chemical structure generation by analyzing QSPR/QSAR models inversely. Regression models between a set of descriptors **x** and property/activity y are analyzed inversely to obtain **x** corresponding to a specific y value. Then, chemical structures are retrieved by a structure generator. The concept of inverse QSPR/QSAR is depicted in **Figure 1-3** by comparing simple VS and GA-based design. As an input for inverse QSPR analysis, only a specific y value is needed. Then, retrieved **x** conditions are used for generating structures *de novo* by a structure generator. The possibility of generating novel structures is compensated by the necessity of a structure generator. GA-based design also generates novel structures. As is discussed in the previous section, initial structures are required for GA-based. In addition, QSPR models are applied forwardly in contrast to inverse QSPR/QSAR.

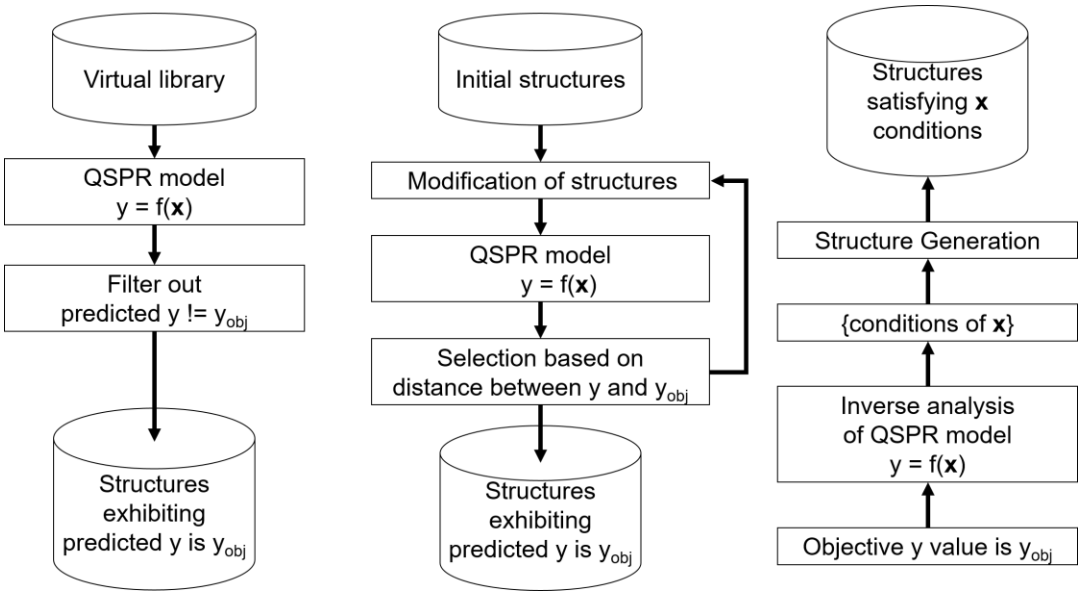


Figure 1-3 Comparison among VS, GA-based design and inverse QSPR/QSAR-based design, assuming the objective variable value is y_{obj}, and one QSPR model

One of the important differences between GA-based design with QSPR/QSAR and inverse QSPR/QSAR-based one is that the former does not use **x** information explicitly whereas the latter does. In inverse QSPR/QSAR analysis, structure generation is conducted only focusing

on the **x** conditions. Therefore, efficient or fast structure generation can be possible by checking **x** conditions without applying QSPR/QSAR models. Since descriptors and chemicals structures are related to each other, structure generation methodologies can be optimized for checking **x** conditions during structure generation. On the other hands, structure generation by inverse QSPR/QSAR scarifies flexibilities that GA-based design possesses, such as diversity-oriented generation and using arbitral complicated QSPR/QSAR models.

Not many research papers report inverse QSPR/QSAR methodologies including GCMs. In GCMs [40,41], setting a specific y value allows a structure generator to generate chemical structures satisfying (1.1). Since the equation is linear, it seems easy for computers to determine all possible sets of combinations for x_i given a certain y. However, exhaustiveness cannot be achieved when the number of groups, which is n, is relatively larger (around ten), because of combinatorial explosion[42] in solution space. Therefore, in order to obtain treatable number of chemical structures, limiting the space or introducing probabilistic operation is required.

When applying topological descriptors instead of the occurrence of groups for inverse QSPR, Kier et al. proposed to obtain a set of structures from QSPR/QSAR equations constructed by multiple linear regression (MLR) with simple descriptors: atom counts, path counts, and connectivity indices. Their strategy was to retrieve path count sequences and vertex degree sequences from the descriptors. Then, structures were reconstructed based on those two sequences via adjacency matrix[43, 44, 45]. They provided an example for designing acyclic alkanes having specific molar volume. Skvortsova *et al.* also proposed solving inverse QSPR problems[46,47,48] with topological descriptors. An assumption of their strategy was that QSPR model was constructed by MLR using topological descriptors, such as Randic indices[49], Kier indices[50], and so on. Roughly speaking, their strategy was to restrict conditions for generating chemical structures through the use of equations. A MLR equation with topological indices as descriptors was regarded as one condition. Based on the descriptor nature, they derived various equations that restricted the number of possible structures. Although the structure generation part was not described in detail, they seemed to retrieve structures from degree sequences and edge sequences consisting of the occurrence of pairs about specific adjacent vertices' degrees. Faulon *et al.* proposed descriptors called *signatures* for representing the topological features of a chemical structure. *Signatures* are atom-centered-typed descriptors; representing an atom in a chemical structure and its surroundings in a recursive manner[51]. His group also had developed an efficient structure generation algorithm (i.e. equivalent classes algorithms)[52], and combined them for solving inverse QSPR/QSAR problems[53,54]. Their proposed methodology was that, when solving inverse problems, a number of equations were introduced in addition to the target QSPR equation by MLR. These equations were necessary in order to guarantee the existence of a structure corresponding to a set of *signatures*. Combined with the QSPR equation, only positive integer solutions were searched with a method proposed by Contejean *et al.*[55].

Apart from deterministic approaches, probabilistic strategies for solving inverse QSPR/QSAR problems have recently emerged[56,57,58]. Utilizing kernel functions in regression models is important for today's QSPR/QSAR analyses, such as kernel-partial least square regression[59] and support vector regression[60]. These kernel-based approaches seek in descriptor space for the coordinates that correspond to desired coordinates in the reproducing

kernel Hilbert space projected by the kernel function. For searching the coordinates, probabilistic algorithms were employed.

1-2-2 Challenges in Inverse QSPR/QSAR

As described in the previous section, (deterministic) inverse QSPR/QSAR can be divided into two parts: obtaining **x** information from a y value, and reconstructing chemical structures from the **x** information. There are several challenges in both parts.

For the former part (i.e. obtaining **x** information from y), there are two challenges. First, applicability domain (AD)[13] was not considered during this procedure. AD is an area where predictive values by QSPR/QSAR model are reliable. It can be determined as dense area in descriptor space (density-based methods)[61]. Briefly speaking, an area far from the nearest training sample is hard to be predicted by regression models since there are no close samples around it. All previous works of deterministic inverse analysis used MLR for representing the correlation between **x** and y. However, a linear regression equation, given a particular value for y, results in the creation of a (n-1)-dimensional subspace, assuming that n is the dimension of **x**. The (n-1)-dimensional subspace as a restriction for structure generation is so broad that if one does not consider generating only a specific type of classes of molecules, such as alkane with 6 carbon atoms, the structure generation would suffer from combinatorial explosion[46]. Furthermore, in that case, the output structures would be hard to be prioritized without introducing additional criteria. These situations root in ignoring AD. A methodology to take the concept of AD into account in inverse QSPR/QSAR was required.

Another challenge was lacking of methodologies treating nonlinear data features. Many QSPR/QSAR models have been developed to treat nonlinear features of training data by nonlinear regression models, since background relationship **x** and y may be nonlinear. However, for deterministic inverse QSPR/QSAR analysis, all analyses were based on MLR to the best of my knowledge because of the mathematical simplicity for retrieving **x** information. Of course, nondeterministic inverse analysis made the search for plural sets of **x** coordinates corresponding to a specific y value possible. It was impossible, however, to obtain a chemical space perspective with which we could select an area in chemical space for retrieving chemical structures.

For the latter part (i.e. chemical structure generation from **x** information), structure generator development was required. This is because descriptors with which QSPR/QSAR models can be constructed are intimately associated with the structure generator, and it is not appropriate to borrow a structure generator from somewhere in a third party. Furthermore, efficient structure generation algorithm should be developed in order to resist combinatorial explosion. Structure generation focuses on chemical structures that are located in a specific area of chemical space. The area is determined by inversely analyzing QSPR/QSAR models before generation. Descriptors spanning the chemical space should be implemented in the structure generator. It is also important to employ various descriptors for constructing QSPR/QSAR models with high predictability while limiting the number of structures as potential candidates in the generation part. Previous studies related to inverse QSPR used specific types of descriptors, leading to insufficient predictability of the regression model. Therefore, a structure generator as well as generation algorithm should be developed so that various descriptors are employed as well as many structures were efficiently generated.

When generating chemical structures, universal AD should be considered as well as a normal AD (model-based AD)[62]. Universal AD is a theoretical concept, which is irrelevant with regression models and is determined based only on the training data before constructing any QSPR/QSAR models. The concept of universal AD is described with the boiling point model by Seybold *et al.*[63] using 39 alkanes with measured boiling points.

$$\mathrm{bp}\left(^{\circ}\mathrm{C}\right) = -126.19 + 33.42\mathrm{N}_c - 6.286\mathrm{T}_m, \qquad (1.2)$$

where N_c is the number of carbon atoms and T_m is the number of terminal methyl groups. The predictability of the model is that R-squared (R^2) is 0.987 and the standard deviation is 5.86. It is fair to say that we can trust the Seybold model when a novel compound is inside the model-based AD. For example, assuming that all the compounds in training dataset are alkanes having from 2 to 9 carbon atoms and having from 1 to 3 terminal methyl groups, and we want to predict the boiling point of 3,4-dimethylphenol. Based on the data density of training data in the two descriptor space spanned by N_c and T_m, we may conclude 3,4-dimethylphenol is inside AD because it has eight N_cs and two T_ms, meaning these descriptor values are inside the ranges of the training dataset. It is, however, not appropriate that the compound is recognized as inside AD because training data for model construction are all alkanes. Only alkanes are eligible to be applied for model prediction judging from the class of compounds in the training dataset[63]. Therefore, we cannot apply this model to any other chemical structures except alkanes, such as a benzene and an acetic acid. This limitation, however, cannot be obtained when simply considering a model-based AD of the descriptor space. For representing this limitation of chemical structures that QSPR/QSAR models can be applied to, the concept of universal AD emerged. We cannot ignore universal AD since otherwise we could obtain predicted boing points of improper compounds with high reliability. For the Seybold model, it is obvious to restrict applied molecules only to alkanes.

In inverse QSPR/QSAR analysis, ignoring universal AD easily causes combinatorial explosion since a structure generator exhaustively generates chemical structures based only on descriptor constraints. Take the Seybold model for example, only constraints of N_c and T_m could be employed when generating structures simply based on the model-based AD in (1.2). As a result, a great number of structures including fused aromatic rings and structures having hetero atoms could be generated by a structure generator. These weak constraints easily lead to combinatorial explosion. In this case, it is correct to set the structure generator to generate only alkanes. When using an arbitrary experimental dataset for constructing a QSPR/QSAR model, however, we could not easily determine which classes of chemical structures should be generated or which should not. Therefore, considering universal AD is important in particular for inverse QSPR/QSAR analysis, but it is difficult to fulfill. **Table 1-2** shows the challenges in inverse QSPR/QSAR mentioned in this section.

Table 1-2 Challenges in molecular design based on inverse QSPR/QSAR divided into two parts.

Obtaining x information from y	Structure generation based on **x**
Not considering AD	Treating limited variety (number) of descriptors
Regression model by MLR	Not considering universal AD

1-3 Objective of This Thesis

The goal of this thesis is to develop a practical chemical structure generation system based on inverse QSPR/QSAR. The proposed system should overcome the challenges mentioned in **Table 1-2** for practical molecular design. In the regression part (obtaining **x** information from y), the author aims to develop a methodology that was able to consider AD in inverse analysis. At the same time, the regression methodology should capture the nonlinear relationship between **x** and y. As for the structure generation, the author aims to develop an efficient generation algorithm that can handle various descriptors during generation. Furthermore, the generator should make their generated structures inherit training data features in a certain extent (i.e. universal AD). Through the development of the structure generation system, the author also intends to test the hypothesis that it is possible to generate exhaustive structures satisfying a specific property or activity based on a QSPR/QSAR model when considering AD.

In practical aspects, generating treatable number of chemical structures as a result of inverse QSPR/QSAR is important. Therefore, in the case that exhaustive generation ends in combinatorial explosion, development of algorithms for diversity-oriented generation is a subsidiary goal in the generation part.

1-4 Structure of This Thesis

Five chapters compose this thesis. In CHAPTER 1 (this chapter), general introduction for molecular design based on QSPR/QSAR is given. In CHAPTER 2, chemical structure generation algorithms are described in order to overcome the challenges mentioned in the previous sections. To take universal AD into account, ring systems[64] decomposed from a training dataset can be used as building blocks. However, there are no known chemical graph construction algorithms so far to treat an arbitrary ring system without treating it as its actual graph structure. This is necessary for avoiding generating duplicate structures. However, by treating a ring system as its graph structure is not an efficient way because the more vertices in a graph are used, the more time it takes to calculate chemical graph operations, such as canonicalization[65]. Furthermore, it needs more storage room on the computer memory. Therefore, a practical algorithm for treating a ring system as a reduced graph having fewer vertices than those in the original ring system is proposed in this chapter. For fast structure

generation, while calculating various descriptor values, a recursive algorithm during generation is proposed. Although early pruning in a generation tree is required, random pruning usually results in not generating even a single chemical structure. To overcome this limitation, monotonous changing descriptors (MCDs) are introduced in the generator. The definition, as well as features of MCDs, is also described in this chapter. In CHAPTER 3, inverse QSPR/QSAR methodologies are introduced for overcoming the linearity limitation and considering AD (model-based AD) of the training data in the descriptor space. Previous researches related to inverse QSPR/QSAR all employed MLR as their regression methodology for deriving constraints as an equation, meaning that AD was not considered at all. The proposed methodology is based on Bayes' theorem and using probability distribution for determining a promising area in which chemical structures have the high likelihood of exhibiting a desired property or activity value based on the QSPR/QSAR model. Employing probability distribution in inverse QSPR/QSAR analysis enables consideration of AD. Furthermore, cluster-wise MLR combined with a Gaussian mixture model (GMM) is also proposed for capturing nonlinear relationship between $\mathbf{x}$ and y. In CHAPTER 4, by combining the methodologies proposed in both CHAPTER 2 and 3, structure generation system based on inverse QSPR/QSAR is proposed. As an example of practical applications, results of ligand design of thrombin direct inhibitors are explained. In this chapter, discussions related to applying the proposed structure generation system to a practical molecular design case study are given. In order to connect the methodologies proposed in CHAPTER 2 and CHAPTER 3, high density regions of a posterior density should be translated into constraints for structure generation. For this purpose, a set of coordinates that exhibits high posterior density is aimed at when generating exhaustive structures. From the results of the analysis in this chapter, exhaustive structure generation turns out to be not always possible even when considering both ADs and focusing only on a specific region in the chemical space spanned by MCDs. In CHAPTER 5, the entire study in this thesis is summarized, clarifying whether exhaustive structure generation is possible or not by the proposed structure generation system. Furthermore, future works related to the proposed methodologies are explained. Although exhaustive structure generation under the assumed conditions was not possible at the current computational power, for practical applications of molecular design, the proposed system along with the proposed methodologies can be applied to any organic molecular design with QSPR/QSAR models.

CHAPTER 2 Structure Generation

2-1 Introduction

In this chapter, structure generation algorithms for inverse QSPR/QSAR are discussed. In inverse QSPR/QSAR, chemical structures should be designed from descriptors' information (**x** information). In other words, we cannot generate structures randomly, hoping to find structures exhibiting the desired **x** information. In this study, a chemical structure is regarded as a chemical graph, where atoms are equivalent to vertices in the same way covalent bonds are to edges. Definition and terms related to chemical graphs are described in the following section.

Structure generation algorithms have been developed since 1960s, originated in the first expert system DENDRAL[66], trying to identify unknown organic compounds from their mass spectra. Until around 1990s, practical applications with structure generators had been almost entirely for the structure elucidation (i.e. identification of unknown compounds). Structure generators for that purpose usually use molecular formula as a first constraint. They generate (enumerate) all possible combinations of chemical structures matching the formula; at the same time, atoms inside the structure must keep the valence rule. MOLGEN[67,68] is one of the fastest structure generator programs even to this day. The algorithm inside it was initially based on orderly generation[69]. Then it was improved using the group action of homomorphism in order to construct chemical graphs from regular graphs having a homomorphic relation to the original graphs. For structure elucidation *in silico*, spectroscopy information is used to reduce building blocks employed in a generation system. CHEMICS[70,71,72] makes good use of carbon 13 and proton nuclear magnetic resonance (NMR) spectroscopies, and infrared (IR) spectroscopy to reduce the number of possible building blocks. The initial building blocks are selected based on molecular formula. In combining building blocks, CHEMICS adopted the connectivity stack algorithm, for reducing the calculation cost, by early pruning of the generation tree consisting of growing structures. In general, generation strategy for structure elucidation is to fill in adjacency matrixes with 1 (meaning connection) instead of combining arbitrary building blocks because, before structure generation, necessary building blocks can be inferred from other information. Therefore, the remaining task is to determine connections among building blocks in order to make candidate chemical structures.

Although structure generators for molecular design have been built upon those for structure elucidation, required conditions in both types of generators are completely different. For designing functional molecules, it is important to consider various types of descriptors. At the same time, it should generate only meaningful structures based on which kind of molecules are designed. If you want to design small molecules for lead discovery in drug design, designed chemical structures should satisfy preferable pharmacophore profiles defined by descriptors. Needless to say, these descriptors should possess adequate description ability. The ability of descriptors is usually evaluated through constructing regression models

and similarity-based VS. DOGS[73] developed by the Schneider's group can generate chemical structures that have similar pharmacophore profile to that of a reference one. DOGS can propose synthesis paths to its proposed structures from building blocks available. Furthermore, reactive structures to proteins in the human body, proteins that are nucleophile[74], must not be generated. Fragment optimized growth algorithm (FOG) developed by Kutchukian et al. makes use of a Markov chain when adding a building block to a growing molecule[75]. In FOG, transition probabilities in Markov chain are based on frequency of specific fragments' connections in a database. When it comes to generating exhaustive drug-like chemical structures having a certain number of heavy atoms (without hydrogen atoms), a series of databases (GDBs) has been developed by Reymond et al[76,77,78]. A structure generator for generating GDBs is first to generate graphs having a specific number of vertices, then to append atom and bond information to the graphs. This structure generator is so efficient that it could generate around 166 billion chemical structures in GDB-17, meaning structures having up to 17 heavy atoms[79].

In this study, generation strategy for inverse QSPR/QSAR analysis emploies ring systems[64] as well as atom fragments as building blocks. The definition of them are described in section2-4 . Examples of ring systems and atom fragments are shown in **Figure 2-1**. The concept of ring systems was originally defined by Bemis and Murcko[64,80]. They investigated comprehensive medicinal chemistry (CMC) database and found that half of the molecules in the database consisted of only 32 frameworks. Similar results were obtained by Tayor et al.[81] They found that 1,175 drugs compiled from the listed drugs in the Food and Drug Administration (FDA) orange book only consist of 351 ring systems. These facts imply that ring systems may be fundamental elements for constructing functional molecules. Furthermore, employing ring systems from the training data used for constructing a QSPR/QSAR model is expected to consider the universal AD. The universal AD determines the class of molecules applicable to QSPR/QSAR models. Using ring systems in a training data restricts diversity of the output structures by a structure generator when compared with arbitrary building blocks that can be combined in every possible way. This restriction, however, is necessary since we generate structures based on QSPR/QSAR models.

My strategy for structure generation is to search candidate structures from the fragment space[82] spanned by ring systems and atom fragments. In order not to produce new ring systems during structure generation, building blocks are combined in a tree-like way[83,84], since ring systems may be a fundamental unit for constructing chemical structures based on a specific training dataset. Employing ring systems in the training dataset and also not producing novel ring systems during structure generation are expected to generate structures inside universal AD. An example of generated structures as well as not generated one was depicted in **Figure 2-2**.

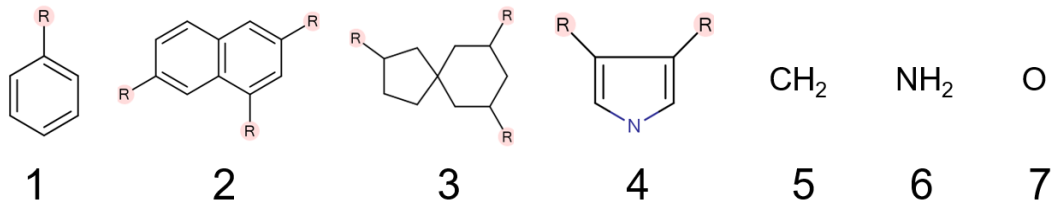


Figure 2-1 Ring systems and atom fragments as components for structure generation. R represents access points at which other fragments can connect. Numbers on the bottom row are mentioned in **Figure 2-2**.

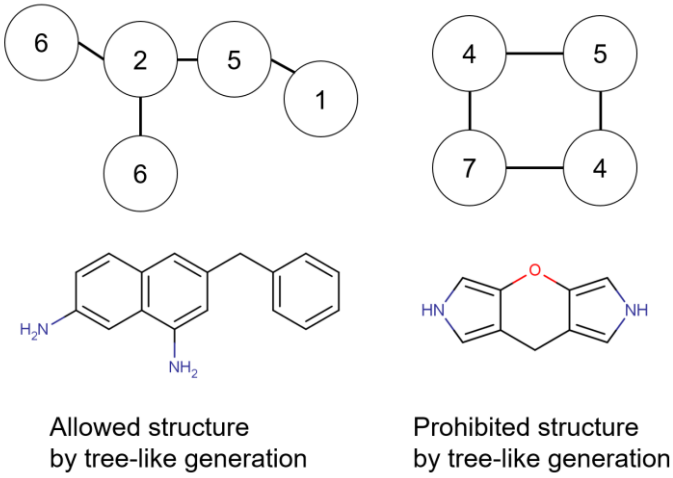


Figure 2-2 Structures that can be generated in a tree-like and not in a tree-like way. The left one is for tree-like generation, which is allowed to be generated in the proposed generation strategy, whereas on the right-hand side of the pictures, it is not allowed to exist. The numbers inside the circle corresponds to those in **Figure 2-1**.

2-2 Challenges of Structure Generation for Inverse QSPR/QSAR

As described in the previous section, ring systems are employed as building blocks for structure generation. This type of structure generator usually suffers from combinatorial explosion due to its combinatorial nature. Moreover, structures should be constructed, obeying various constraints defined by descriptor values. For satisfying these conditions, two challenges should be overcome in a designed structure generator: efficient generation algorithm in order to resist combinatorial explosion, and a framework that enables using various types of descriptors during structure generation.

First, when combining ring systems, as well as atom fragments, in a tree-like way, there have been no general algorithms to treat the symmetry of ring systems correctly without

treating them as they are. Treating a ring system as a vertex during structure generation is ideal for speeding up structure generation because the fewer vertexes in a graph are used, the faster graph operation can be calculated. In the proposed algorithm, basic graph operations, such as canonical labeling, are carried out. Calculation time for them depends on the number of vertices in a graph. Treating a ring system as a vertex, however, does not work at all, when considering the symmetry inside a ring system and that of the whole chemical structure as well. Jindalertudomdee *et al.* proposed to generate exhaustive-tree graphs containing a benzene or a naphthalene[85,86,87]. Their strategy is first to generate exhaustive tree-like structures including a benzene or a naphthalene as a vertex having valence of 6 or 8, respectively[88]. Then they assign symmetry of the rings to the corresponding vertices, and finally eliminate duplicate structures. Although their algorithm is efficient, it is still impossible for their algorithm to be applied to general ring systems as elements for structure generation.

Second, various descriptors as constraints should be introduced in a structure generator. As described in the previous section, for practical molecular design, a proper descriptor set should be implemented. These descriptors must be diverse and not be composed of a certain descriptor class. Unfortunately, there have been no publications about structure generation for inverse QSPR/QSAR, considering constraints with various types of descriptors. This is because arbitrary descriptors cannot tell a structure generator information about how to generate structures. Therefore, previous researches in this field have focused on using specific types of descriptors, such as Randic connectivity indices, *signature* descriptors. Another motivation for using many descriptors is that the more constraints are employed when producing candidates during structure generation, the fewer the possible solutions (structures) appear. One way to avoid combinatorial explosion is to limit the area for structure generation in chemical space tightly. Furthermore, it is important to select the descriptors that have low degeneracy[89] for limiting the number of solutions[51].

In addition to these two challenges, combinatorial explosion should be handled properly on condition that exhaustive structure generation is intractable. Structure generation by combining building blocks always face combinatorial explosion, no matter how efficient the generation algorithm is. Bohacek et al. estimated the size of chemical space to 10^{60} by counting the possible combination of up to 30 heavy atoms (carbon, nitrogen, oxygen, and sulfur atoms including hydrogen atoms implicitly)[27]. Combinatorial explosion can also be observed by combining ring systems and atom fragments. Miyao *et al.* did a simple experiment[42] to estimate the size of fragment space. They randomly combined 289 ring systems and atom fragments of 7 atoms in a tree-like way. The estimated number of structures reaches over 10^{21} by combining 10 building blocks. Therefore, it is important to generate a diverse set of structures in this case.

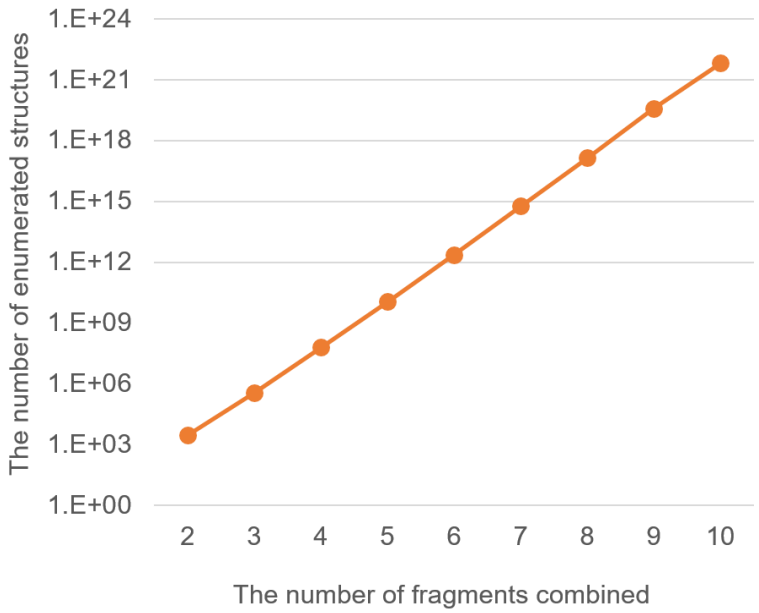


Figure 2-3 Number of estimated structures to be generated by combining 289 ring systems and 7 types of atoms against the number of fragments combined

2-3 Objective and Strategies

The goal of this chapter is to propose practical algorithms for handling three challenges as follows: efficient structure generation by combining ring systems and atom fragments, constructing a framework capable of handling various descriptors, and diversity oriented structure generation for practical application. To overcome these challenges, three algorithms are proposed, corresponding to these challenges. First, for efficient structure generation, McKay's canonical construction path method[90] is modified in order to treat building blocks' symmetry. Furthermore, the concept of a reduced graph is introduced for speeding up graph operation. Second, in order to consider constraints by various descriptors during structure generation, monotonous changing descriptors (MCDs) are introduced. MCDs are descriptors whose value change monotonously when attaching a building block to the structure. As long as a descriptor satisfies the criteria of MCD (section 2-6), the descriptor is used for inverse analysis. Consequently, various descriptors can be employed in QSPR/QSAR modeling. For calculating MCD values efficiently, a recursive algorithm for updating MCD values is also proposed. Finally, to generate diversified structures as a result of generation, a pseudo framework-based structure generation algorithm is proposed. The algorithm is deterministic and generates one structure for each pseudo framework that the author defined. With this algorithm, one-by-one comparison of generated structures is unnecessary. For the case of deterministic approach failing to generate feasible number of structures, stochastic generation is proposed based on a paper by McKay[90]. In the following sections, methodologies of these three strategies are described in detail.

2-4 Preliminaries for Structure Generation Algorithms

Before explaining algorithms, terms related to structure generation should be clarified. In this study, a chemical structure is a chemical graph. Following explanations are based on several text books[83,91,] and related articles[58,52]. A chemical structure (M) is regarded as a colored graph G(V, E), where V is a set of colored vertices, E is a set of multi-edges. Coloring corresponds to the type of heavy atoms with explicit hydrogen atoms (**Figure 2-4**). Although this simplified representation of a molecule cannot distinguish stereoisomers in nature, this is sometimes necessary for treating a vast number of chemical graphs efficiently. **Figure 2-4** shows two stereoisomers. Each pair of stereoisomers is translated into an identical chemical graph, which is a loss of information by simplification of molecules into chemical graphs.

The only reason for using two-dimensional chemical graphs instead of three-dimensional representation of chemical structures is pursuing computational efficiency. Discarding stereo information in structure generation is compensated by possibility of searching many candidates. Furthermore, there is no consensus about whether three-dimensional descriptors are superior to two-dimensional ones[92,93]. One of the reason why three-dimensional descriptors, such as comparative molecular field analysis[94], do not necessarily show higher predictability than two-dimensional ones is that it is difficult to obtain actual conformation of a molecule under the experimental condition where objective variable values were measured. When determining conformation of a molecule for calculating three-dimensional descriptors, molecular mechanics is usually employed with the help of force fields to be minimized, such as universal force filed[95] and Merck molecular force field[96]. Assuming that low predictive QSAR models can be constructed with three-dimensional descriptors calculated with wrong conformation of molecules, discarding that three-dimensional molecular information in QSPR/QSAR analysis makes sense.

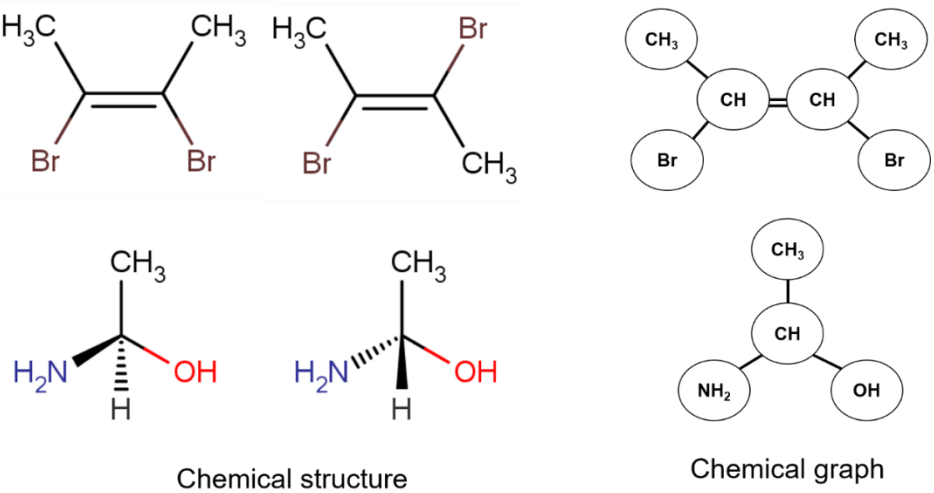


Figure 2-4 Stereoisomers that cannot be distinguished from each other by transforming them into chemical graphs

Construction of chemical graphs is conducted by combining ring systems and atom fragments. In this study, a ring system is defined as the ring system that Bemis and Murcko

defined[64] with explicit access points at which other building blocks should be attached[97]. On the other hand, atom fragments are heavy atoms with explicit hydrogen atoms. Hydrogen atoms are usually suppressed in chemical graphs due to pursuing computational efficiency. From an article by Miyao *et al.* [97], examples of ring systems and atom fragments are illustrated in **Figure 2-5**.

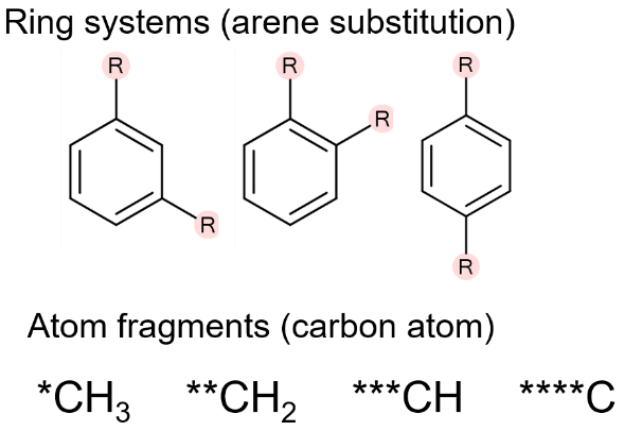


Figure 2-5 Examples of ring systems with access points and atom fragments with explicit hydrogen atoms. In the top row, three ring systems with different substitution patterns are regarded as different fragments. Rs represent access points (substitution points). In the bottom row, carbon atom fragments with explicit hydrogen atoms are depicted. *The remaining degree of the fragment to its valence. This figure was copied from the article by Miyao et al. [97] with permission of Springer.

When treating chemical graphs, recognizing graphs' symmetry is necessary for both canonization and identification of the same type of vertices. The symmetry is represented with graph automorphism. Definition of graph isomorphism and automorphism is as follows:

Definition of graph isomorphism and automorphism

Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two chemical graphs consisting of sets of vertices V_1, V_2 and sets of edges E_1, E_2.

A graph isomorphism is a bijection $\varphi: V_1 \rightarrow V_2$

s.t. $\forall u_1, u_2 \in V_1$ and $u_1u_2 \in E_1$ then $\varphi(u_1)\varphi(u_2) \in E_2$ and $u_i, \varphi(u_i)_{i=1,2}$ has the same color.

If $G_1 = G_2$, the bijection satisfying the criteria above is an automorphism.

2-5 Ring System-based Structure Generation

2-5-1 Structure Generation Algorithm

In order to combine building blocks efficiently, ring systems, which consist of many atoms, should be regarded as simplified graphs during generation (i.e. graphs having fewer vertices than those in the original ones). This is because fewer vertices are preferable for graph operations, such as canonicalization, calculation of graph invariants, and so on. Although treating a ring system as a vertex is ideal, constructed graphs by combining this type of building blocks show wrong topology compared to the actual chemical graphs. Therefore, the concept of reduced colored graphs is introduced to tackle this challenge.

A reduced colored graph has the isomorphic automorphism group of a ring system in terms of access points. The definition of group homomorphism and isomorphism are as follows:

Definition of group homomorphism and isomorphism[68]

Let G_1 be group acting on a set X_1, G_2 be group acting on a set X_2. A pair of maps $\varphi=(\varphi_x, \varphi_g)$ where $\varphi_x: X_1 \rightarrow X_2$ and $\varphi_g: G_1 \rightarrow G_2$ is a group homomorphism if φ is compatible with both actions,

i.e. for all $g \in G_1$ and all $x \in X_1$

$$\varphi_x(x^g) = \varphi_x(x)^{\varphi_g(g)}.$$

If both components of φ are bijective, φ is an isomorphism.

Then, reduced colored graph is defined as follows:

Definition of the reduced color graph for a ring system

Let A_1 be a set of access points in a ring system, and G be the automorphism group acting on A_1. The reduced colored graph R, containing a set of colored vertices Y corresponding to A_1 and the automorphism group H, is a graph where a pair of bijective maps $\varphi=(\varphi_x, \varphi_g)$ among vertices and automorphisms respectively defined:

$g \in G$ and all $x \in A_1$

$$\varphi_x(x^g) = \varphi_x(x)^{\varphi_g(g)},$$

where $\varphi_x: X \rightarrow Y$, and $\varphi_g: G \rightarrow H$.

Since a ring system is a chemical graph with access points, it can be represented as G (V_1, E_1, A_1), where A_1 is a set of access points in this ring system, which is also a subset of V_1. The corresponding reduced graph G_{red} (V_2, E_2, A_2) has the same automorphism group in terms of A_2 as that in terms of A_1. Since these two automorphism groups match each other, and every chemical graph consists of ring systems and others (*i.e.* atom fragments), the two graphs can be replaced for the purpose of graph automorphism detection (*i.e.*, graph extension considering symmetry). Therefore, we can use G_{red} instead of G when appending other building blocks to it. If we choose G_{red} having fewer vertices than those in G, calculation speed of the program is expected to increase since fewer vertices are used. In this

work, a graph consisting of reduced graphs is called a contracted graph. A contracted graph holds the same topology as that of the original chemical graph in terms of access points, even though the contracted graph has fewer vertices than those in the original one. The important premise is that every chemical graph is decomposed into ring systems and other structures (*i.e.*, atom fragments). Actions of automorphisms are only comparable between the same type of ring systems. Hence, automorphisms between a chemical graph and the corresponding contracted graph are hold in terms of access points.

An underlying idea of using contracted graphs and reduced graphs is that topological complexity of a chemical graph is transferred to a simpler colored graph. Simple examples of reduced colored graphs along with the corresponding ring systems are depicted in **Figure 2-6**. It should be noted that the colored graph that has the same automorphism access points group as that in the corresponding ring system is not unique. There might be other colored graphs corresponding to the ring system. We can use one of them. One way to find a reduced colored graph is to make use of graph templates. By connecting graph templates, and coloring them in a proper way, a candidate-reduced graph is produced. Then, the automorphism group of access points in the candidate graph is compared with that in the original ring system. An algorithm for searching a reduced colored graph is described in Appendix A. When coloring vertices in template graphs, orbits of access points in the corresponding ring system can be used. It should also be noted that the procedure of detecting reduced colored graphs is conducted before structure generation. Hence, it does not affect generation time.

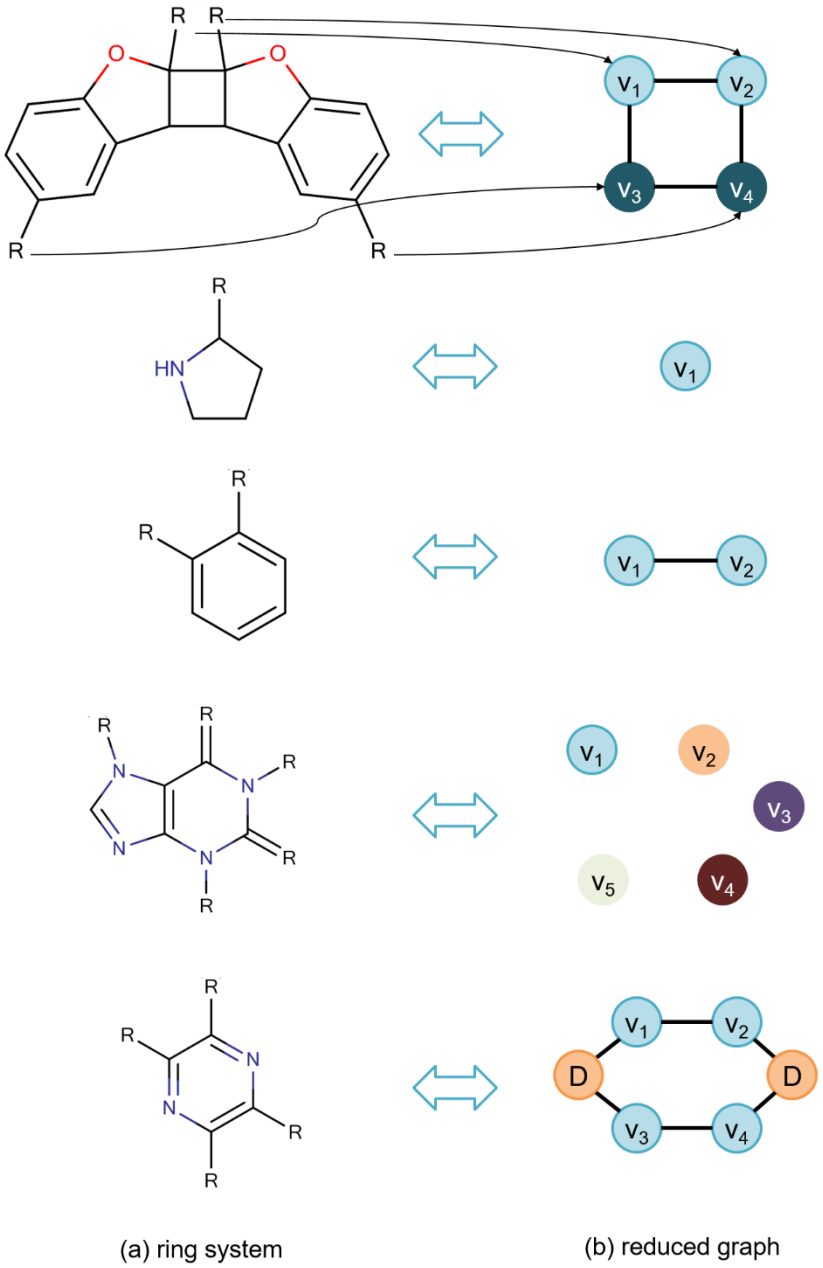


Figure 2-6 Reduced colored graphs (b) and their correspondent ring systems (a). Rs represent access points. Ds represent dummy vertices for preserving the symmetry between a ring system and the corresponding reduced graph. This figure was copied from the article by Miyao et al.[97] with permission of Springer.

An example chemical structure and a corresponding contracted graph are depicted in **Figure 2-7**. On the left-side of the figure, the chemical structure has 40 vertices. Whereas on the right-side, the contracted graph has only 8 vertices, leading to enhancing the calculation speed.

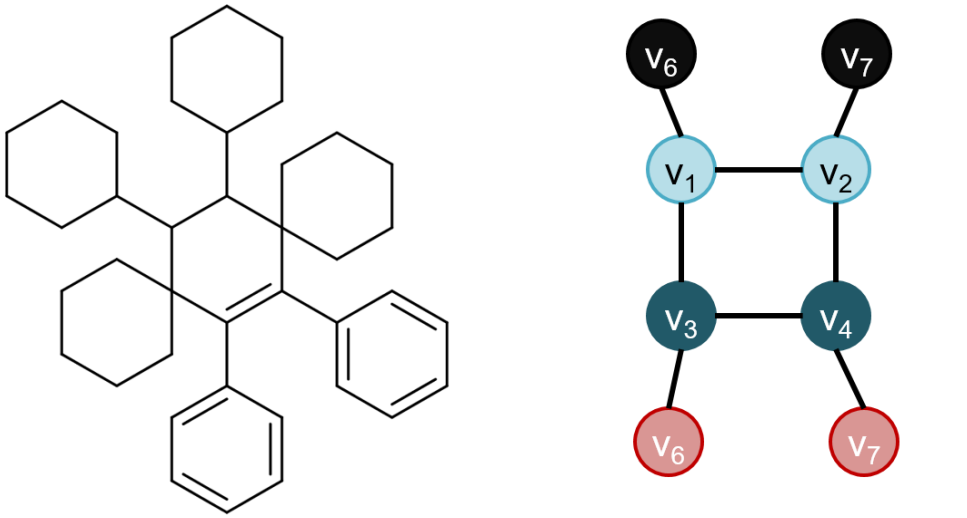


Figure 2-7 Original chemical structure consisting of five ring systems (a) and the corresponding contracted graph (b). This figure was copied from the article by Miyao et al. [97] with permission of Springer.

To construct contracted graphs instead of chemical graphs, the canonical construction path method proposed by McKay[90,98] was employed. This algorithm was modified to take symmetry of building blocks (i.e. reduced graphs) into account. The basis of McKay's original algorithm is simple. A graph is produced by adding a vertex to a smaller graph through the canonical construction path. Passing along the canonical construction path assures that generated graphs are unique as well as exhaustive. One strong point of the algorithm is that it can be applied to diverse problems. In this study, this algorithm is improved for producing contracted graphs as chemical graphs, meaning that a contracted graph is extended by adding a reduced graph to a smaller contracted one.

The proposed algorithm is written on the **Table 2-1** in the form of pseudo codes. The procedure *scan* produces all saturated contracted graphs (graphs having no access points remaining) in a recursive manner. Function *scan* takes two arguments, the growing contracted graph (X) and the upper number of reduced graphs to be combined (n). When adding a new reduced graph to X, an access point (apt) from each orbit in X is selected (line 5, 6). After checking whether ap is the proper access point at which another reduced graph (FR) can be attached, the child contracted graph (X_{new}) is produced by connecting FR to ap in X. Finally, when X_{new} is confirmed to be the actual candidate by mother function (m), function *scan* is recursively called with the arguments X_{new} and n until X_{new} becomes saturated. Function m determines the path of generated contracted graphs. Based on the paper by McKay[90], m should satisfy a condition as follows: first, $m(X_{new})$ selects an orbit of the action of all automorphisms of X_{new} on X_{new}. Then, eliminate one of the reduced graphs in the orbit one after another to make a set of $X_{ret}=m(X_{new})$. When X_{ret} contains X, the canonical construction path from X to X_{new} is extended.

A simple way to define m is to make use of canonical labeling. Selecting a set of vertices matching the specific number in canonical labeling under automorphism of the graph. And

m returns *true* if the set of vertices (reduced graphs) contains the vertex (the reduced graph) that has just been attached, otherwise m returns *false*. To recognize automorphisms of chemical graphs through contracted graphs, topologies between them should be preserved at the level of access points.

Table 2-1 Algorithm of growing contracted graphs by adding reduced graphs and atom fragments. The algorithm was modified from the code in Ref 90. This table was copied from the article by Miyao et al. [97] with permission of Springer.

line	Pseudo code
1	**procedure** *scan*(*X*: contracted graph, n: integer)
2	**if** *X* is saturated **then**
3	**Output** *X*
4	**Endif**
5	**for** each orbit *A* from the action of *Aut(X)* on *X* **do**
6	select any representative access point $apt \in A$
7	**if** remaining degree of *apt* is not zero **then**
8	**for** each reduced graph *FR* to be attached to *X* at *apt* **do**
9	make X_{new} from (*X*,*FR*,*apt*)
10	**if** X $\in$ m(X_{new}) and o(X_{new} **)** $\leq n$ **then** scan(X_{new}, n) **endif**
11	**endfor**
12	**endif**
13	**endfor**
14	**endprocedure**

Figure 2-8 shows how contracted graphs grow by appending a reduced graph to a smaller contracted one along with the corresponding chemical graphs. A reduced graph is attached to the parent graph by considering the symmetry of both the parent and the reduced graphs. **Figure 2-8** also shows that using reduced graphs during generation process makes the generation process simpler than using the original chemical graph, because different reduced graphs have different colors. Coloring is a key to reduce the calculation cost of canonicalization since finding automorphisms should be considered only among the vertices having the same color. In this example, the two children do not match each other because of the color difference between black and blue access points.

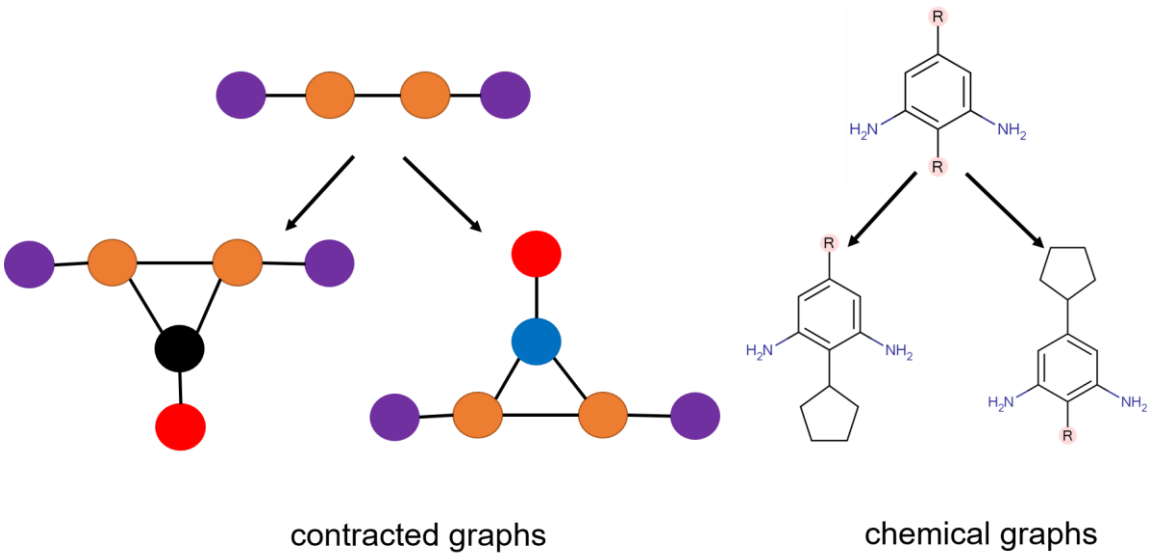


Figure 2-8 Growing structures by adding a ring system to a smaller one. Child structures are produced by adding building blocks to a parent one. On the left picture, contracted graph format. On the right, the corresponding chemical graphs.

To recognize the topology of a contracted graph as the same way as the original chemical graph, coloring should be consistent among different patterns of reduced graphs, for the same chemical graph, when the original patterns are isomorphic (**Figure 2-9**).

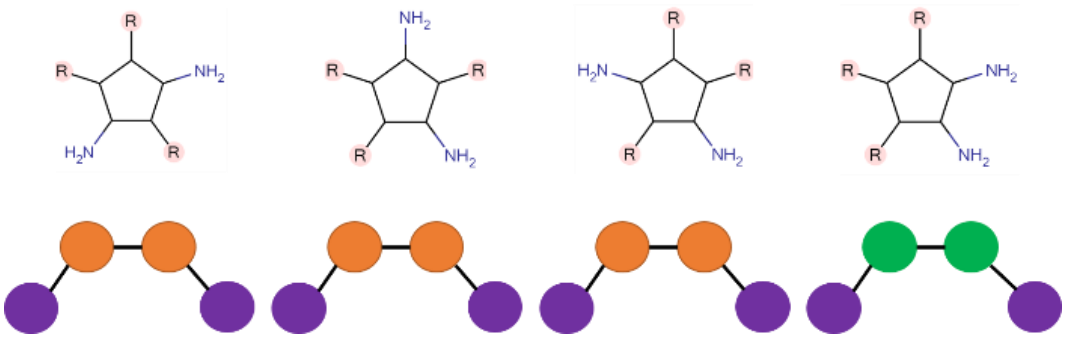
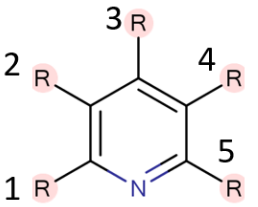


Figure 2-9 Chemical graphs (top row) and corresponding contracted graphs (bottom row). The left three graphs (both chemical graph and contracted graph) are isomorphic among one another, whereas the rightmost one does not correspond to the remaining ones.

This recognition is conducted during the mother function checking procedure. In order to increase the calculation speed, maps between filled access points and the corresponding coloring can be stored as tables in advance. An example of the tables along with corresponding graphs are in Figure 2-10. The size of both tables for a ring system having n access points is 2^n-1. And most ring systems do not have any symmetry in them, therefore it does not cause any problems in a program with fewer n (*e.g.* 8).



A

1	2	3	4	5
1	0	0	0	1
1	1	0	0	0
0	0	0	1	1

B

1	2	3	4	5
A	0	0	0	A
B	C	0	0	0
0	0	0	C	B

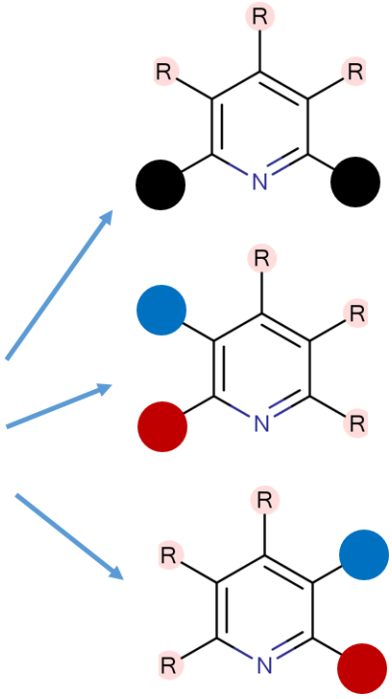


Figure 2-10 Table of containing maps between filled access points and the corresponding coloring patterns based on pyridine with 5 access points as an example. Table A represents whether or not access points are filled, and B is the corresponding color mapping. Different alphabets in Table B mean different colors.

Another trick for enhancement of the calculation speed is related to line 5 in **Table 2-1**: the procedure for detecting orbits of the access points that are not currently filled. To detect the orbits of access points without further calculation of automorphism, a lookup table can be made before structure generation. It contains mapping information between the orbits of a reduced graph and the orbits of the unfilled access points of the aforementioned graph. The idea about this lookup table is explained with **Figure 2-11**. In the middle row, pyridine with 5 access points is currently connected to three building blocks: two hexanes and one methyl. Based on the contracted graph on the center column, access points 2 and 4 are in the same orbit. This information can be stored before structure generation, because the orbits of the unfilled access points are determined based on the orbit pattern of the currently used access points. Hence, it is possible to presume all possible combinations of access point orbits in advance. On the bottom row, pyridine is connected to one hexane and one piperidine instead of two hexanes. This leads to the difference of orbits between access point 1 and 5. Therefore, access points 2 and 4 are no longer in the same orbit.

The size of this table is larger than that for detecting coloring of the used access points in a reduced graph (**Figure 2-10**). For 6 access points, there are 203 patterns, and for 7 access points, there are 877 patterns. This table is also made only for the ring systems having symmetry. The number of structures having symmetry is much fewer than those having it. Therefore, in practical application, the size of the table for at most 7 access points can be manageable on a personal computer.

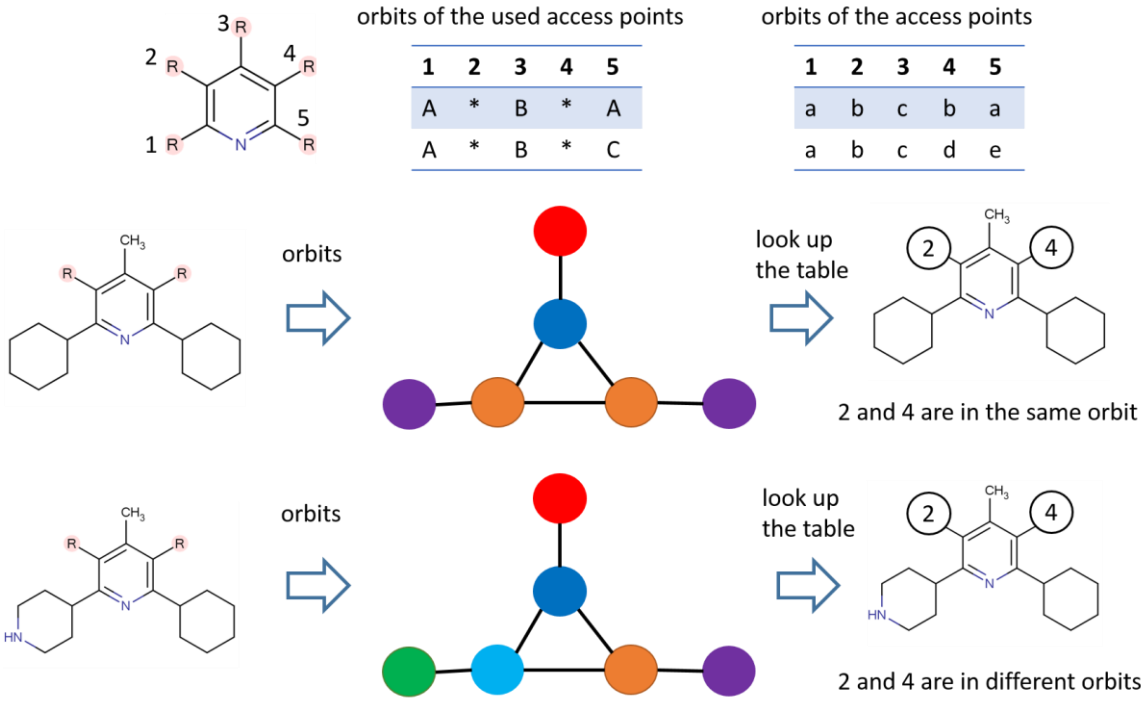


Figure 2-11 Lookup table for determining mapping between the orbits of currently used access points and that of the unfilled access points in a ring system. On the top row, mapping between orbits of the used access points and the unused ones for pyridine with 5 access points is shown. Capital letters represent orbits based on contracted graphs in the middle column. Small letters in the top-right table represent orbits inside the pyridine corresponding with coloring in the used orbits' table (the top-center table).

2-5-2 Generation Performance

Performance of the proposed algorithm was compared with that of a structure generator developed by Arakawa et al.[41]. The structure generator is implemented in Chemish[99], which is an integrated software developed by the Funatsu group at The University of Tokyo for chemoinformatics analysis. The algorithm of the generator in Chemish is simple. It exhaustively combines building blocks until the number of used ones reaches a predetermined value or saturated structures are constructed. We call the generator a simple fragment-combined-based structure generator for convenience. After structure generation, canonization operation and elimination of duplicate structures are conducted. Therefore, the simple fragment-combined-based structure generator is comparable with the proposed structure generator despite the fact that the canonicalization algorithms are different (In the simple fragment-combined-based structure generator, the canonical algorithm is based on the Morgan method[35], whereas that of the proposed generator is based on McKay's algorithm). The proposed algorithm has been implemented in a structure generator system, Molgilla, which is explained in detail in section 2-8 .

As building blocks, ten ring systems (**Figure 2-12**) and 13 atom fragments were employed. The atom fragments were CH_3, CH_2, CH, C, NH_2, NH, N, OH, O, F, Cl, Br, and I. The

number of combined fragments was set from two to eight. For each number of fragments, five trials were conducted in order to evaluate the calculation time statistically. The performance test was conducted on a Windows 10 personal computer with 3.33GHz Intel Xeon CPU and 16 GB RAM.

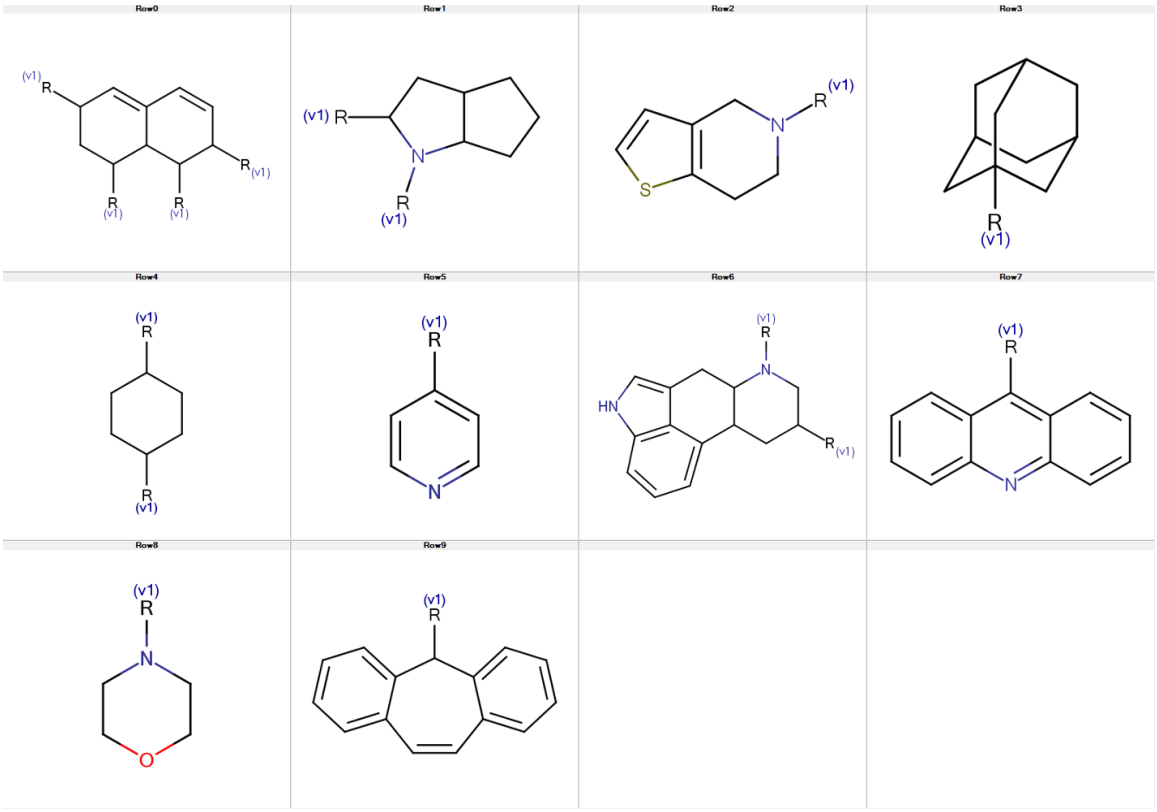


Figure 2-12 10 ring systems used for the speed test. This figure was copied from the article by Miyao et al.[97] with permission of Springer.

Results of the performance tests are shown on **Table 2-2**. The calculation time at the first three rows for Molgilla (the number of fragments are from 2 to 4) are more or less the same. This is because the overhead of the program was the most time consuming element in these trials. The simple fragment-combined-based structure generator could not generate structures by combining more than 6 fragments. It should be noted that both generators could generate the same number of structures. On **Figure 2-13**, generation time against the number of generated structures is shown. In both cases, the relationship between calculation time and the number of generated structures looks linear. For Molgilla, calculation time for the numbers of fragments 2, 3, and 4 is almost identical because limitation for these trials was preparation of structure generation (i.e. making threads) although this performance test was conducted with a single threading. For the simple fragment-combined-based structure generator, it takes 1.39×10^{-3} per structure, whereas Molgilla takes 3.83×10^{-6} per structure. These results support efficiency of the proposed algorithm in generating structures by combining building blocks

Table 2-2 Generation time between the simple fragment-combined-based structure generator and the generator based on the proposed algorithm (Molgilla). Number inside parenthesis is standard deviation based on five trials.

Fragments	Simple generator* [s]	Molgilla [s]	Structures [-]
2	0.07 (0.01)	0.26 (0.07)	100
3	0.72 (0.03)	0.29 (0.01)	812
4	6.49 (0.18)	0.29 (0.02)	8037
5	101.74 (1.42)	0.56 (0.01)	116559
6	2761.59 (22.1)	5.2 (0.12)	1995641
7		98.57 (0.97)	33674221
8		2167.92 (24.09)	566840430

*: simple fragment-combined-based structure generator

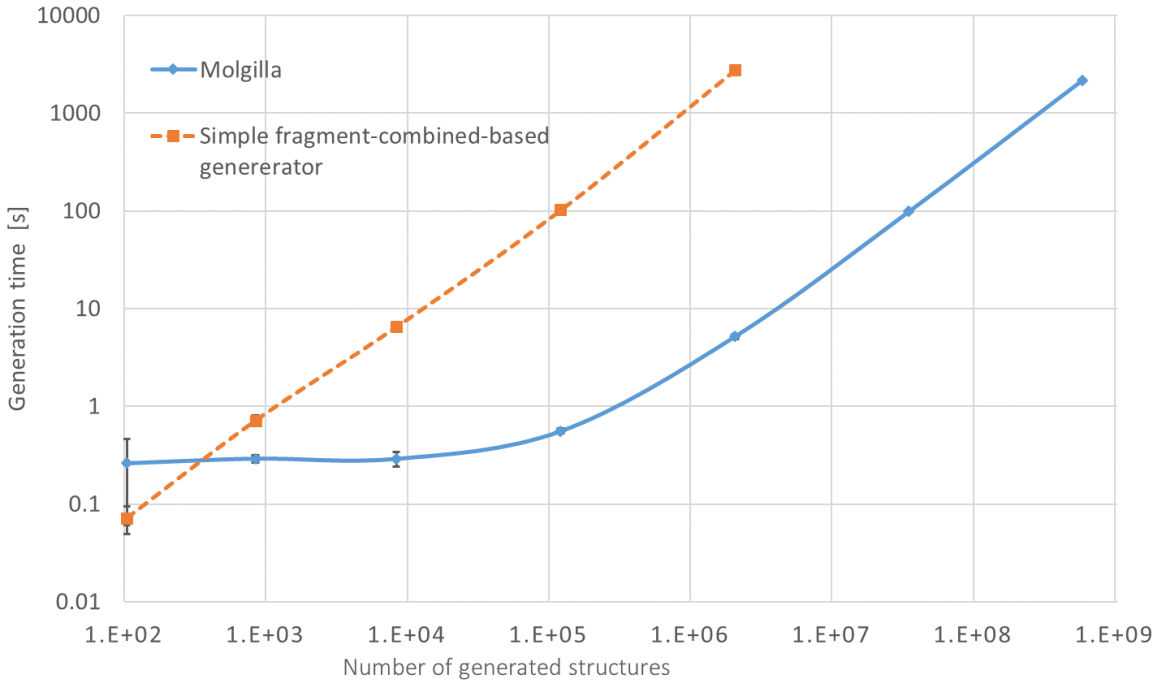


Figure 2-13 Calculation speed comparison between Molgilla and the simple fragment-combined-based generator. The error bar is based on the three times standard deviation of 5 trials. Every dot corresponds to the number of building blocks combined, from 2 to 6 for Chemish, and from 2 to 8 for Molgilla. This figure was copied from the article by Miyao et al. [97] with permission of Springer.

2-6 MCDs

2-6-1 Definition of MCDs

An efficient structure generator does not necessarily mean that it is useful for inverse QSPR/QSAR analysis. As described in section 2-1 , structure generator should generate structures only satisfying descriptor constraints. In this study, these constraints are defined as upper and lower bounds of descriptor values. Without setting constraints, structure generation resulted in combinatorial explosion.

In order to take as many constraints into consideration as possible during generation process, MCDs[100] are introduced. MCDs are also called consistency constraints[101]. The definition of MCDs is as follows. Assuming a chemical graph (G), an arbitrary building block to be attached (F) and the map (R) from G to the descriptor value, then

R is a MCD if and only if sign(R(G-F) - R(G)) is consistent, (2.1)

where sign is a map representing whether the argument is positive (including zero) or negative (including zero) and R(G-F) is the chemical graph made by attaching F to G. More than 99% of MCDs have positive consistency because molecular descriptor values tend to increase by adding extra building blocks instead of decreasing in nature. For example, molecular weight always increases by adding an extra building block (F) to a chemical graph (G).

2-6-2 Relation of MCDs with Structure Generators

Although MCDs were precisely defined in the previous section, whether a descriptor is a MCD or not depends on how to combine fragments and the type of fragments employed. **Figure 2-14** explains how generation strategies and fragment types influence MCDs. On the top row, for example, the number of hydroxyls is a MCD in case a), whereas it is not in case b). Without representing a heavy atom with explicit hydrogen atoms, there is no way to know whether the number of hydroxyls increase or not during structure generation. On the bottom row, a novel ring is produced by connecting two access points of an oxygen atom to two carbon atoms. This changes the shortest path length between carbon atoms at the end of the upper building block in c) from four to two.

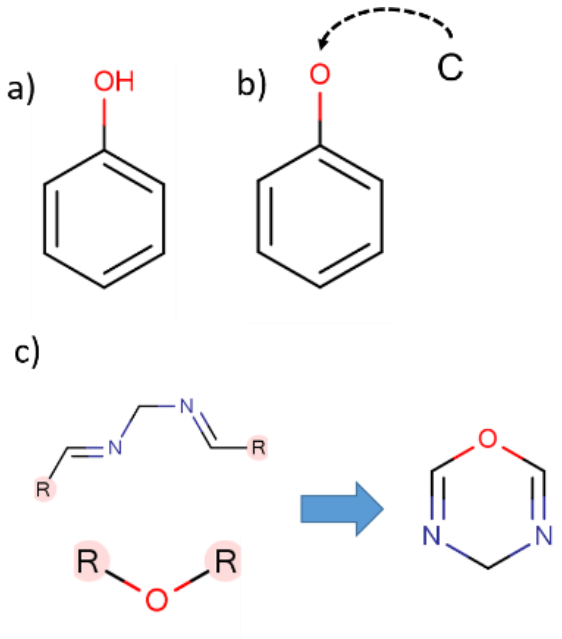


Figure 2-14 Illustration of whether a descriptor is a MCD or not. In a), the fragment is expressed by heavy atoms with explicit hydrogen atoms. In b), heavy atoms with implicit hydrogen atoms are used. In c), the building blocks are combined to form a new ring system.[102]

Using MCDs as descriptors during structure generation enables the proposed structure generator to efficiently prune branches in a generation tree. The values of MCDs of a growing structure always monotonously change by adding an extra building block. Therefore, we can remove structures once one of their MCD values is over the upper bound of the constraint. In **Figure 2-15**, the shaded area is constrained in the descriptor space spanned by two MCDs. The growing structure (arene) increases its MCD values by gaining building blocks. Once one of the values passes the upper bound of one of the constraints (MCD2), the structure is removed. Even with pruning operation being conducted, exhaustiveness of the generated structures is assured because of the nature of MCDs.

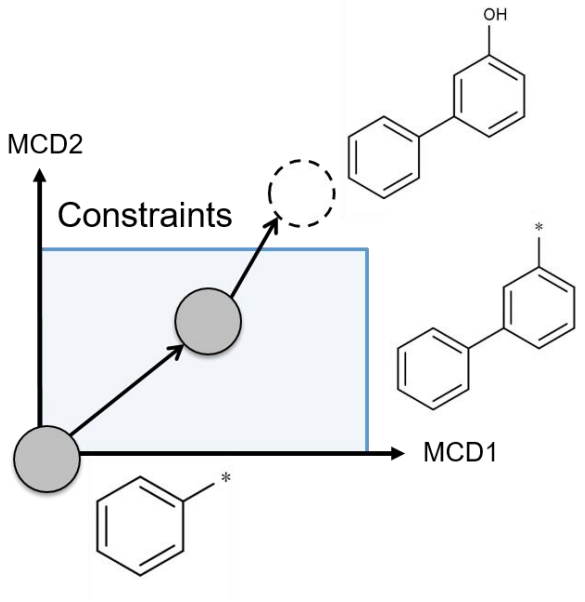


Figure 2-15 Example of trajectory of growing a structure in MCD space.

2-6-3 Types of MCDs and Description Ability

Miyao et al. examined [103] how many MCDs exist in a diverse set of descriptors. They investigated the descriptors implemented in DRAGON[104], which is one of the comprehensive descriptor calculation software today. According to their research, 547 descriptors are categorized as MCD from 929 descriptors (0D, 1D, and 2D), such as molecular weight, Randic connectivity indices, molecular walk counts, the occurrence of substructures, and so on. They constructed two partial least square (PLS) regression models with an aqueous solubility dataset[9,105] using 547 MCDs and 929 descriptors respectively. The predictability with MCDs was compatible to that using all 929 descriptors. R^2 for the test dataset was 0.823 with MCDs, whereas 0.847 for the 929 descriptors. In this thesis, similar study for QSAR was conducted in order to evaluate predictability with MCDs.

2-6-3-1 Dataset

Dataset for constructing a QSAR model was extracted from the GVK database[106] for human alpha 2A adrenergic receptor. Compounds were annotated with the corresponding pK_i values (logarithm of reciprocal of K_i (inhibition constant)). There were 1,062 ligands in it. From the original structure pool, 10 dimers, which are outliers based on visual inspection of the two-dimensional map with principal component analysis (PCA). Randomly selected 800 samples were used for model construction. The remaining 252 samples were used as test data. Software DRAGON ver. 5[104] calculated 790 descriptors (0D, 1D, and 2D) for these ligands. and the descriptors were categorized into MCD or non-MCD judging from calculation algorithms for descriptors. There were 409 MCDs, 312 non-MCDs, and the 69 descriptors which could not be determined whether they were MCD or not. When categorizing the descriptors, the author assumed that chemical structures were extended in a tree-like way by

adding building blocks consisting of heavy atoms with explicit hydrogen atoms (i.e. atom fragments). The complete list of MCDs is on Table B-1 in Appendix B.

2-6-3-2 Comparison of MCDs with DRAGON Descriptors

Correlation coefficients among MCDs and all descriptors were respectively calculated. All of the possible pairs of descriptors were calculated in each descriptor set. The histograms of the correlation coefficients are shown in **Figure 2-16**. One interesting thing in that figure is that MCDs do not necessarily show strong positive correlation among one another. The outline of the histogram for MCDs look more or less the same as that for DRAGON descriptors. Some pairs of MCDs are negatively correlated, mainly because of substructure count-based descriptors. This result supports us to construct linear regression models with MCDs in the same way with DRAGON descriptors because in linear regression, it is necessary to eliminate the rest of highly correlated descriptors in order to avoid collinearity.

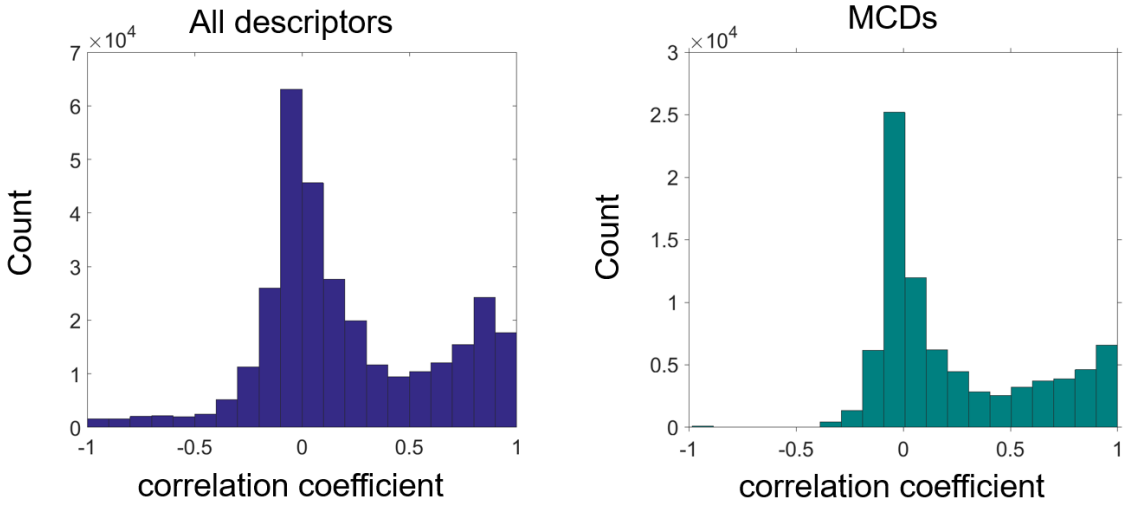


Figure 2-16 Histogram of correlation coefficient. The correlation coefficient was calculated among all the pairs of descriptors.

Next analysis is to construct QSAR models with those two descriptor sets, and to compare predictability between them. Regression methodology was PLS. The optimal number of components for PLS was determined based on Q^2 from 5-fold cross validation. The results of constructing models and validation are shown on **Table 2-3** and **Figure 2-17**. Predictability with MCDs was a little worse than that with DRAGON descriptors as expected, since the MCDs was a subset of DRAGON descriptors in this case study. Nevertheless, R_{pred}^2 was over 0.8 with MCDs. In this study, we confirmed that MCDs have as adequate description ability regarding constructing regression models as DRAGON descriptors do. It should be noted that, in this case study, all possible MCDs were employed. It is, however, difficult to realize in an actual inverse QSPR/QSAR workflow because these descriptors must be implemented in a structure generator in order to make use of their values as constraints during structure generation.

Table 2-3 QSAR models performance with MCDs and DRAGON descriptors.

	Opt. Compt. [a]	Q^2	$RMSE^b_{cv}$	R^2	$RMSE^c_{pred}$	R^2_{pred}
MCD	10	0.832	0.354	0.891	0.385	0.836
DRAGON[d]	11	0.859	0.324	0.918	0.347	0.867

a: Number of optimal components in PLS regression model.

b: Root mean square error (RMSE) for cross validation.

c: RMSE for test dataset.

d: 790 descriptors were used for constructing PLS model.

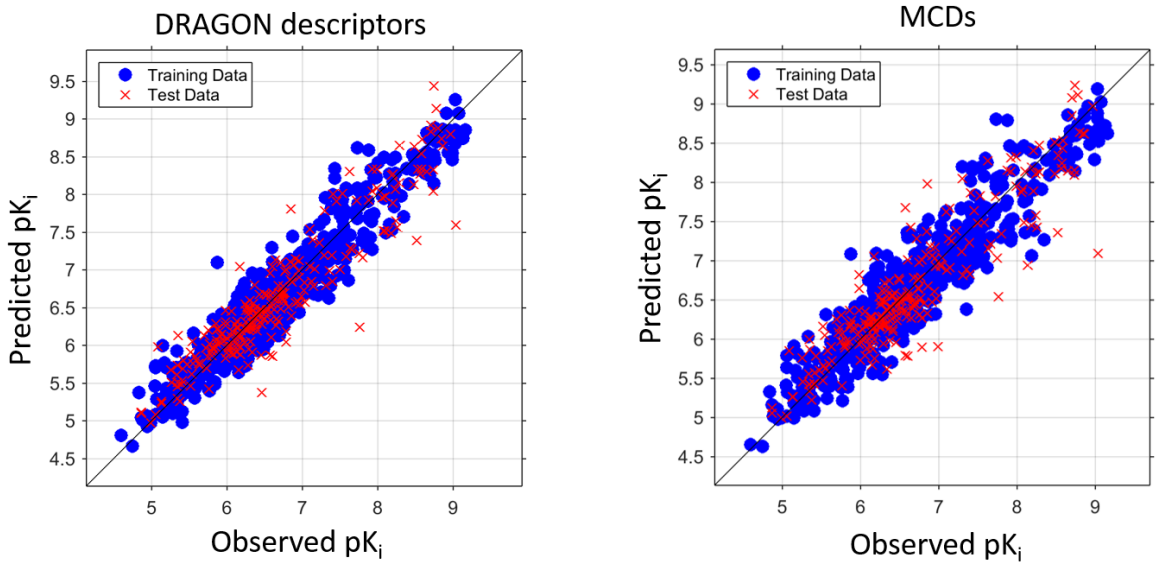


Figure 2-17 Predicted pK_i plotted against observed pK_i by PLS.

2-6-4 Sum of Topological Distances between Potential Pharmacophoric Points (STDPs)

Although there are many MCDs available in theory for the proposed generation algorithm (*i.e.* canonical construction path method), only a selected part of them can be practically employed in structure generation, because of the high cost of descriptor calculation. Therefore, it is important to determine which descriptors should be implemented and which should not, in terms of calculation load and description ability.

In designing small molecules as lead candidates in drug design, molecular shape is one of the most important factors. This is because the shape of small molecules should be complementary to their target macromolecule. Furthermore, non-covalent interactions between a small molecule and the macromolecule should also be complementary. Therefore,

methodologies for capturing pharmacophore of ligands have well been studied from both structure-based[107] and ligand-based[108,109] approaches. Unfortunately, since even ligand-based approaches require three-dimensional structures of small molecules, the methodologies cannot be directly utilized in the proposed two-dimensional approach.

The group of Schneider has developed a set of autocorrelation-type descriptors, which can represent pharmacophore features of a molecule. They named the descriptors chemically advanced template search (CATS) descriptors[110,111]. CATS descriptors purely depend on two-dimensional molecular topology (*i.e.* chemical graph). They have succeeded in capturing pharmacophoric features in many case studies, such as similarity-based VS[112], visualization of natural product space.[113] and understanding pharmacophoric features of fragments obtained by dissecting natural products[114].

In order to incorporate descriptors that are able to capture pharmacophore features into the structure generator, summation of topological distances between potential pharmacophoric points descriptors (STDPs) are defined. STDPs are MCD since they are calculated as summing up topological distances between a pair of potential pharmacophoric points (PPPs). Examples of STDPs are shown in **Figure 2-18**. Obviously, the most important part of STDPs is the definition of PPPs. The following six PPPs are defined by the author, based on the definition of PPPs in the CATS descriptors.

Lipophilic point (L):

A lipophilic point is not an aromatic atom. It is either Cl or Br or I. A carbon atom surrounded by only carbon atoms or hydrogen atoms is regarded as lipophilic. A sulfur atom connected to only two carbon atoms is also lipophilic.

Hydrogen bond acceptor (A):

A hydrogen bond acceptor point is either Cl or F or O. A nitrogen atom is also a hydrogen bond acceptor point when it does not connect to any hydrogen atoms or is not formally charged.

Hydrogen bond donor (D):

A hydrogen bond donor point is either OH or a nitrogen atom having at least one hydrogen atom at most three hydrogen atoms. Nitrogen atoms, however, should not have formal charge for being recognized as a hydrogen bond donor.

Negatively charged point (N):

A negatively charged point is the carbon, sulfur, or phosphorus atoms in COOH, SOOH, POOH respectively.

Positively charged point (P):

A positively charged point is a nitrogen atom connected to two hydrogen atoms (e.g. primary amines).

Aromatic ring (R):

A ring having (4n + 2)-electrons is an aromatic one.

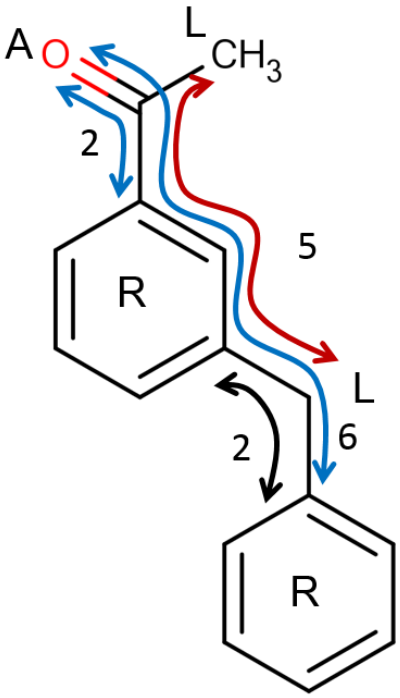


Figure 2-18 Examples of calculation of STDPs in 1-(4-benzylphenyl) ethanone. STDPs between L and L is 6, A and R is 8, and R and R is 2. L is a lipophilic point, A is a hydrogen bond acceptor, and R is an aromatic ring.

2-6-5 Calculation of MCDs

Fast calculation of MCDs during structure generation is necessary because the generator should search as many solutions as possible from a large solution space. MCDs can be recursively updated by calculating the impact of the addition of a building block to a growing structure on their values. As the same characters are used here as in the Eq. 2.1. G is a chemical graph, and an arbitrary building block F is attached to G. R maps G to a descriptor value. By attaching F to G, the descriptor value changes from R(G) to R(G-F). R(G-F) can be represented as:

$$R(G\text{-}F) = R(G) + R(F) + R(GF), \tag{2.2}$$

where, R(GF) represents the difference between R(G-F) and R(G) + R(F), meaning a non-additive effect on R(G-F). Some descriptors do not have R(GF), such as molecular weight and the number of multiple bonds. Other descriptors, however, do have this member, such as connectivity indices and STDPs because the way of connection between access points influences these descriptor values.

The non-additive descriptor values can also be efficiently updated in a recursive way as long as R(GF) can be efficiently updated. An example of calculating a STDP is described in Appendix C. The basic idea of updating R(GF) in STDPs is to make use of distance matrix. Since distance matrix can be updated in a constant time by appending a fragment in a tree-like way, descriptors making use of the distance matrix can be updated efficiently. R(GF) can also be updated by making use of the distance matrix. It should be noted that descriptor calculation does not depend on chemical graphs employed as elements for structure generation. In other words, in order to calculate MCD values, we do not need chemical graphs explicitly but do need components for calculating MCD values. For example, when calculating the number of rings in a chemical graph during structure generation, the number of rings in each ring system is only required. we do not need to store the chemical structures themselves and counting rings every time a chemical structure obtains an additional ring system. This trick also allows us to use reduced graphs instead of actual ring systems during structure generation since MCD calculation and extension of chemical graphs are separated from each other.

2-7 Diversity-oriented Structure Generation[97]

Although exhaustive structure generation is desirable, it is not always possible. As we explained with **Figure 2-3**, combinatorial explosion could occur without introducing adequate number of constraints. Furthermore, even when exhaustive structure generation is tractable, it is still important to produce a diversified set of chemical structures for later parts of time-consuming studies (e.g. docking simulation, molecular dynamics simulation, and synthesis). In this section, a strategy for diversity-oriented generation is introduced. The biggest difference between the proposed methodology and diversity-oriented sampling methodologies is the timing of considering diversity. In our methodology, the diversity is considered during structure generation, meaning that we can access a larger number of potential candidates than using the sampling methodology. Furthermore, unlike structure generators that take into account diversity by comparing one candidate with the rest of all generated structures,[38] the proposed algorithm does not need to conduct such a comparison. It makes the algorithm efficient. In addition to the development of the algorithm, stochastic generation is also proposed to reduce generated structures as well as to estimate the number of structures to be generated without exhaustive generation.

2-7-1 Pseudo Framework-based Generation Algorithms

2-7-1-1 Algorithm

Some of the generated structures by the proposed algorithm often have the same scaffold. The only difference among those structures is an atom fragment at the end of the structures. The objective of the algorithm here is to suppress generation of structures similar to one another based on structure scaffold (**Figure 2-19**).

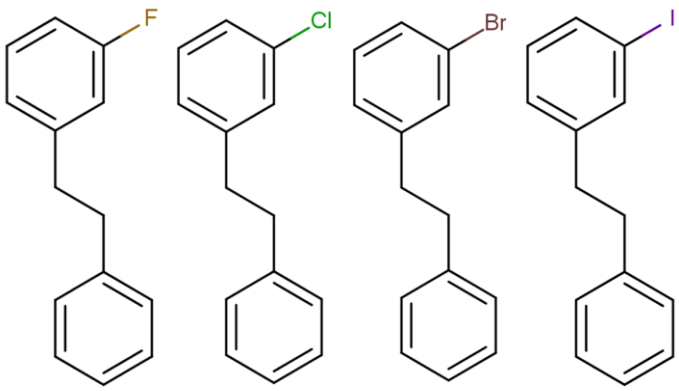


Figure 2-19 Examples of combinatorial structures lacking in diversity. This figure was copied from the article by Miyao et al. [97] with permission of Springer.

Bemis and Murcko proposed the concept of frameworks as well as ring systems, trying to extract common scaffolds of molecules in a database[64,80]. According to their papers, a molecule is divided into three parts: ring systems, linkers, and side-chains. A framework consists of ring systems and linkers (except side-chains). Unlike their definition of framework, the proposed pseudo framework-based generation algorithm is not graph-based but atom fragment-based, because the smallest unit of building blocks in structure generation is an atom fragment.

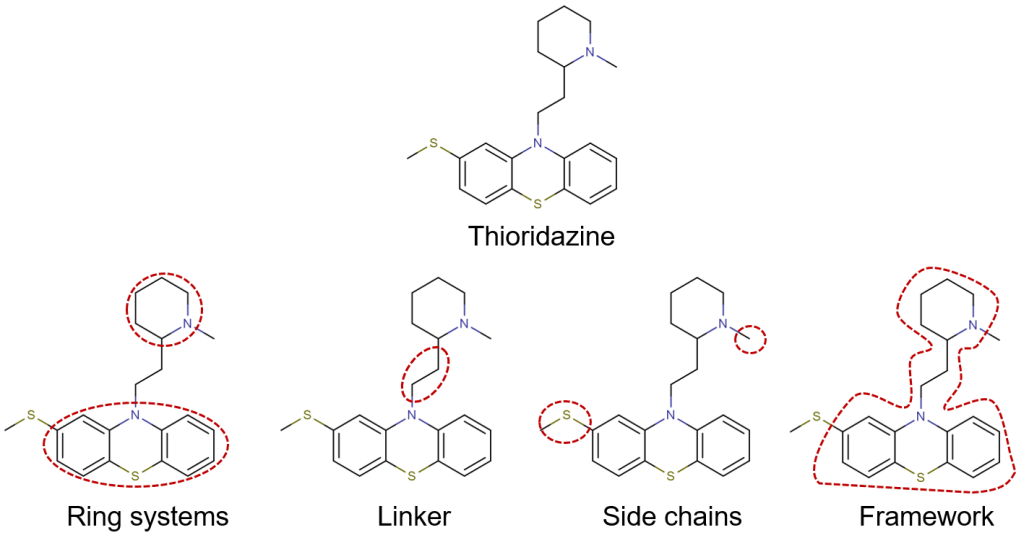


Figure 2-20 Ring systems, side chains and framework in thioridazine. The definition of ring systems is atom-based.

The algorithm is based on orderly generation,[69,101] and on depth-first search for a generation tree. Orderly generation is widely used when combining building blocks aiming at reducing the probability of enumerating duplicate structures. One of the features of orderly generation beside its canonical augmentation is to assign labels (orders) to building blocks. The assigned label corresponds to the priority of a building block. The order of appending building blocks is restricted by the labels. A building block annotated with a high priority

should be attached to a growing structure later than that with a lower priority. A building block having the highest priority should be attached at the last moment of completion of a chemical structure. For example, there are two building blocks (B_1 and B_2), and they are connected to each other, and their priorities are high and low, respectively. In this case, B_1 can be attached to B_2 in order to make B_1- B_2, not vice versa.

Before applying the proposed methodology, the order of priority should be determined as follows:

i) Atom fragments that are supposed to be located at the end of a chemical structure, such as CH_3, NH and OH, have the highest priority.

ii) Atom fragments except those categorized in case 1, such as CH_2 and N, have the second highest priority.

iii) Ring systems have the lowest priority.

The proposed algorithm is explained using **Figure 2-21**. Growing structures are stored in a stack one after another. And the structure at the top of the stack is selected as a parent for producing children. In **Figure 2-21**, structure 1 makes three children: structure 2, 3, and 4. Structure 4 has a different pseudo-framework from that of structure 1, 2, and 3. After several iterations, structure 3 is selected as a parent, and it produces structure 5 and 6. When structure 6 is made by attaching an atom fragment categorized in the highest priority (i), that structure fixes its pseudo framework. In other words, structure 6 and its descendants cannot produce structures having different pseudo frameworks. Therefore, once structure 6 is produced, the signal that suppresses the generation of structures having the same pseudo framework (a and b in **Figure 2-21**) as that of structure 6 is sent to the other structures stored in the queue.

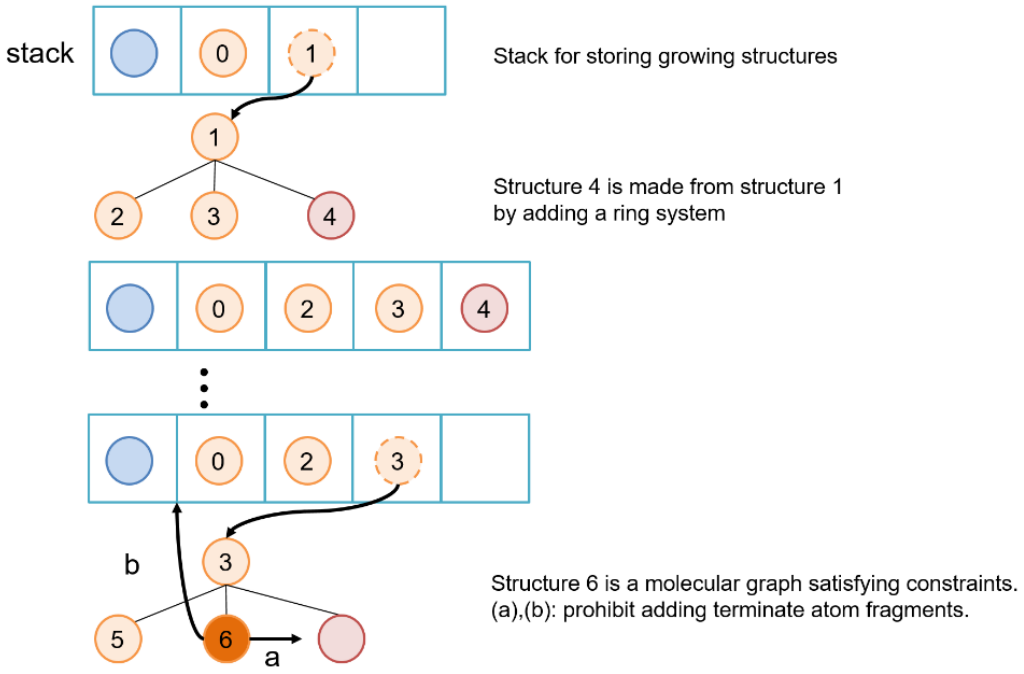


Figure 2-21 Schematics on how to generate structures with pseudo framework-based generation. Stack is for storing atomic graphs. The structure on top of it is selected as a parent structure. Once a chemical graph is completed, the program is not allowed to select terminal atom fragments for extension from the structures having the same pseudo framework (a, b). Detailed explanation is on the main body. This figure was copied from the article by Miyao et al. [97] with permission of Springer.

2-7-1-2 Performance

An effectiveness of the proposed algorithm was confirmed by actual structure generation. Employed building blocks were one ring system, which is arene with one access point, and 7 types of atom fragments, C, N, O, F, Cl, Br, and I. One strategy is to generate exhaustive structures; the other is to generate a diversity set of structures by pseudo framework-based generation. In this case study, some prohibition rules (See the section of 0) were introduced not to generate reactive and unstable structures[74]. The number of fragments combined was from 2 to 7. Generator Molgilla was forced to generate structures containing at least one ring system (*i.e.* arene).

The number of generated structures were 21,249 and 624 by exhaustive and pseudo framework-oriented strategies, respectively. To evaluate the diversity of the generated 624 structures, two databases were additionally constructed. One consists of randomly chosen 624 chemical structures from 21,249 structures (A), and the other consists of diversified 624 structures from the same structure pool by the diversity oriented sampling methodology of MaxMin (B).[115] Average similarity among the k-nearest neighbors for all the structures in a database measures the diversity (or similarity) among databases. Similarity is defined based

on Tanimoto similarity using of molecular access system (MACCS) key fingerprint.[116] **Figure 2-22** shows the average k-nearest neighbored similarities against the number of neighbors. As we expected, our strategy is just between the exhaustive database and B. It should be noted that average similarity of the exhaustive database converges to 0.282, which is smaller than that by pseudo framework-based generation. The point is that the pseudo framework-based generation can diversely generate smaller number of structures than that of exhaustive ones when the same number of neighbors is selected. The algorithm can make the generator search for a higher number of solution candidates in the entire solution space.

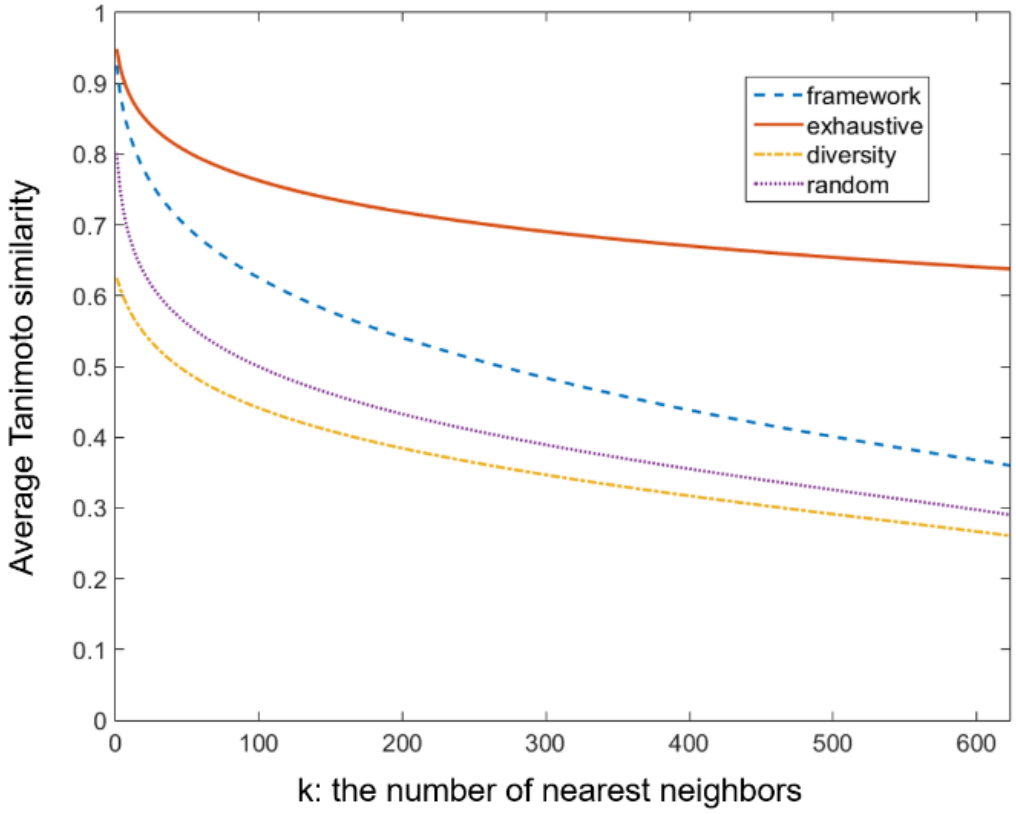


Figure 2-22 Average pairwise Tanimoto similarity among k nearest neighbors, framework: pseudo-framework-based generation, exhaustive: pool of exhaustive structure generation, diversity: MaxMin sampling from the exhaustive structure pool, random: randomly picking from the exhaustive structure pool. 624 molecules were sampled for diversity and random cases This figure was copied from the article by Miyao et al. [97] with permission of Springer.

2-7-2 Stochastic Generation

2-7-2-1 Algorithm

Another strategy to reduce the number of structures to be generated is to conduct a stochastic generation. The methodology is originally proposed by McKay[90]. It assigns a probability to

each node in a generation tree. With that probability, the node is eliminated, meaning that a branch of the generation tree is pruned with the probability. In his original paper, the probability is the same for all the nodes in a generation tree. Here, a generalized formula that allows elimination of nodes with different probabilities according to the depth in a generation tree is introduced (**Figure 2-23**). Introducing this stochastic procedure can reduce the total number of generated structures. At the same time, it enables the estimation of the number of structures to be generated, if pruning procedure was not conducted (*i.e.* exhaustive generation). The expected number of structures to be generated is

$$E[N] = \sum_{i=1}^{depth} N(i) \prod_{j=1}^{i} (1 - p_j)^{-1} \quad (2.3)$$

,where N(i) is the number of actually generated structures at depth i, and p_j is the probability of pruning at depth j. Briefly speaking, the estimation number in Eq. (2.3) is the summation of the expected numbers of vertices, loading actually generated vertices with the eliminated structures, which are supposed to be generated without the stochastic operation.

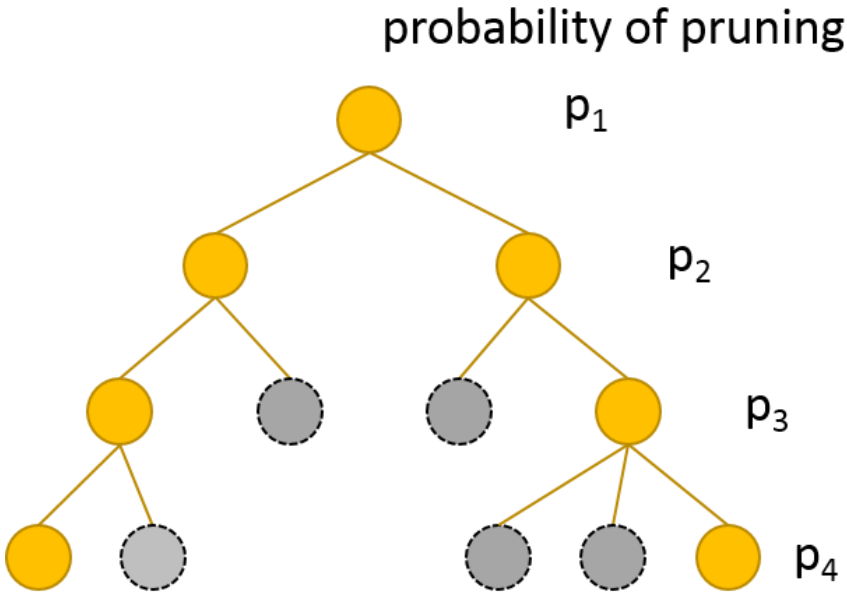


Figure 2-23 Generation tree and nodes in the tree with probabilities. Gray nodes are eliminated from the tree, and yellow ones survive in generation procedure. The probability of whether a node is eliminated or not depends on the depth in the tree.

2-7-2-2 Performance

Stochastic generation was conducted with randomly selected 100 ring systems in order to confirm how accurate it can estimate the number of structures to be generated. The ring systems are obtained by decomposing all molecules in ChEMBL20[117]. Probabilities of pruning were p_0=0, p_1=0.75, p_2=0.91, p_k=0.95 (k >2). The reason for the assigned probabilities was to accurately predict that number. Accurate estimation is only possible

when chemical structures appear at every depth of a generation tree. Therefore, assigning lower pruning probabilities to lower depth nodes and higher probabilities to higher depth nodes is desirable. From **Figure 2-24**, the stochastic generation methodology may estimate the number of structures to be generated with adequate precision.

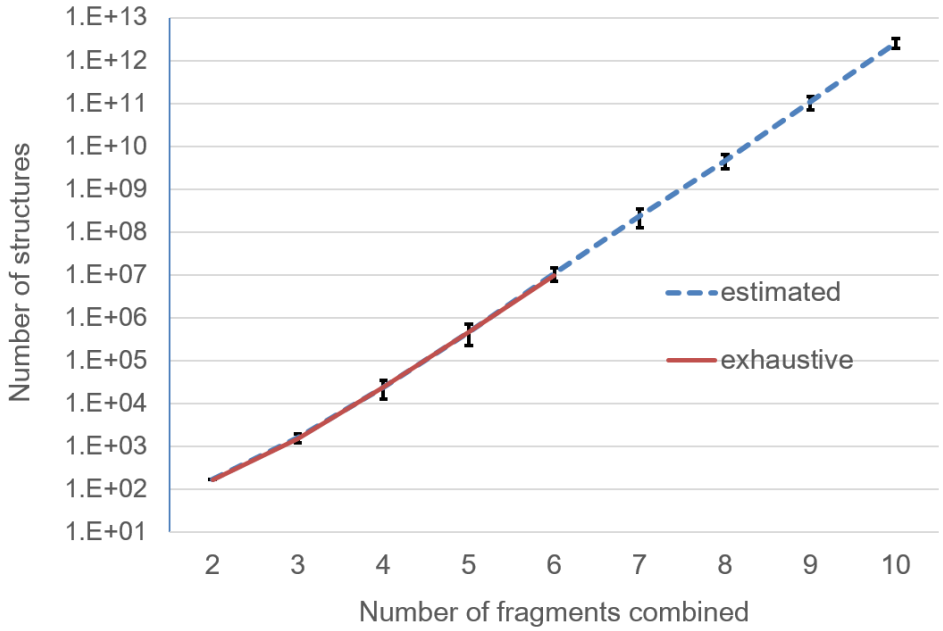


Figure 2-24 Number of estimated and generated structures against the number of combined fragments (logarithmic scale). The red line is the number of structures that were actually generated without sampling method. Blue dotted line is the average estimated number of structures and the range of error bars is 2 standard deviations from the average values. 10 trials were conducted for each fragments. This figure was copied from the article by Miyao et al. [97] with permission of Springer.

2-8 Implementation

All algorithms mentioned in this chapter were implemented in the structure generator system of *Molgilla*. *Molgilla* consists of three modules: *FragmentGenerator*, *DecomposeRingFragments* and *DescriptorCalculator*. *FragmentGenerator* is responsible for structure generation. It is a multi-threaded application, which enables parallel calculation. In this module, a taboo list of unsuitable substructures is also implemented without preventing calculation speed. Unsuitable structures are reactive and/or unstable. Based on the papers by Blum et al.[118] and Rishton[74,119], substructures to be employed in Molgilla were selected. The selected substructures can be efficiently searched during structure generation, which is a criterion of compiling the list. Furthermore, the list contains rules that make generated structures easier to synthesis. There are 16 substructures implemented in the current version of Molgilla. The complete list is in Appendix D along with chemical structures exemplifying substructures in the list.

To make a unique set of ring systems from a compound pool, the *DecomposeRingFragments* module was developed. MCDs that can be used in Molgilla are on the Appendix E with precise definition. These descriptor values can be calculated by the *DescriptorCalculator* module.

The overview of Molgilla and relations among those three modules are in **Figure 2-25**. All the codes of Molgilla are written in C++ with RDKit libraries[120] and Boost libraries[121].

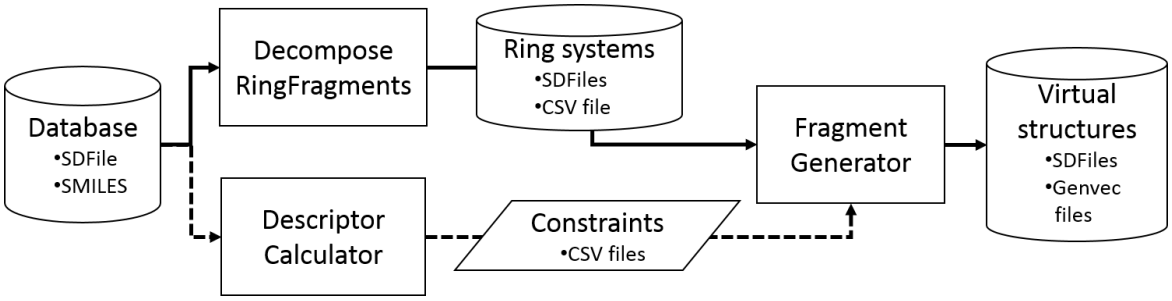


Figure 2-25 Typical workflow in the structure generator system Molgilla. File formats of chemical structures are mentioned in the picture. Genvec is a binary format for contracted graphs.

2-9 Conclusion

Structure generator Molgilla for inverse QSPR/QSAR analysis was developed. For fast calculation, reduced graphs instead of actual ring systems can be used without generating duplicate and missing structures. Calculation speed of Molgilla surpasses that of a simple fragment-combined generator in Chemish as shown by the case study of exhaustively combining building blocks. For QSPR/QSAR models in inverse analysis, MCDs should be employed as descriptors. MCDs' high predictability was confirmed through QSAR model construction for the alpha 2A adrenergic receptor. When exhaustive structure generation is intractable or undesirable, diversity-oriented structure generation or stochastic generation can be conducted. The extent of diversity and estimation accuracy were also tested through simple case studies. Concrete algorithms were provided as well as several supporting results. In the current version of Molgilla, 51 MCDs including 21 STDPs were implemented. Moreover, it suppresses the generation of unstable and reactive structures by comparing substructures in the taboo list the author compiled.

CHAPTER 3 Inverse QSPR/QSAR Analysis (from y to x)

3-1 Introduction

In this chapter, inverse QSPR/QSAR related to acquiring descriptor information (**x** information) from a desired objective variable value (y value) is discussed. The goal of this analysis is to extract useful information from a QSPR/QSAR model for structure generation. The QSPR/QSAR model is constructed with experimental dataset before conducting inverse analysis. As described in section 1-2-2 , almost all the previous researches about inverse QSPR/QSAR have been accompanied with MLR as a regression methodology. In the field of graph theory, there is a theorem that can assure the existence of simple graphs corresponding with a vertex degree sequence, i.e., Hakimi-Havel theorem[122]. The theorem also tells how to reconstruct graphs recursively. Researchers on inverse QSPR/QSAR analysis, accordingly, aimed to derive the vertex degree sequence matching the MLR equation given a y value. Descriptors of graph invariants, such as Kier indices[50], Hosoya indices[123], etc., can be efficiently transformed into the corresponding vertex sequences by Kier[43,44,45], Skvortsova et al.[48,124], and so on. Apart from deriving vertex degree sequences, Faulon *et al.* directly make use of a MLR equation as a constraint[51,53] for structure generation. They purposely introduced additional two types of equations as constraints: graphicality and consistency equations. The graphicality equation refers to a constraint analogous to deriving vertex degree sequences, meaning the summation of all vertex degrees must be even. The consistency equations make sure the existence of the graphs matching a set of signatures. They confirm the consistency among all the signatures by comparing their values with corresponding graph structures.

They all used MLR because MLR equations become constraints for structure generation once y values are determined. The methodology, however, contains two substantial disadvantages as the author already mentioned in section 1-2-2 . One is not considering AD, and the other is poor predictability due to the regression being linear.

A MLR constraint usually results in spanning (n-1)-dimensional subspaces, assuming that n is the dimension of **x**. Therefore, simply applying this constraint allows the existence of enormous chemical structures. Assuming that one constraint reduced the number of structures by 0.1 percentages, the number of estimated structures would be 10^{57}, which is derived once 0.001 is multiplied by the chemical space size (10^{60}) [27]. The missing part in this deliberation is whether training dataset information is being taken into account. MLR is not a deterministic linear equation, but a statistic model constructed with training dataset. Therefore, AD must be considered when analyzing regression models, in particular inverse QSPR/QSAR.

In order to overcome this limitation, Miyao *et al.* proposed to use the posterior distribution of **x** given a specific y value[100]. Using a parametric probability density function (PDF) for representing posterior **x** information has several merits besides the ability of considering AD:

1. Understanding which region in descriptor space shows high density and which does not is put in perspective.

2. Conducting sampling operations (diversity-oriented sampling) based on the PDF, meaning chemical structures can be stochastically sampled.

Both merits root in the fact that the posterior distribution of **x** given a desired y value brings the landscape in the descriptor space. As an example for the second advantage aforementioned, a chemical structure generation strategy that samples structures based on a PDF in descriptor space can be applied directly with the derived posterior PDF, given a y value, such as the methodology proposed by White and Wilson[125].

Although the methodology Miyao *et al.* have proposed can take AD into consideration, since descriptor information is retrieved as a parametric PDF in descriptor space, predictability of regression models is still hindered by using MLR as a regression methodology. Linear regression limitation should be overcome, so to make inverse QSPR/QSAR analysis practical. Nevertheless, it is still important to represent **x** information as PDF because of the merits that have been mentioned above. In this chapter, cluster-wise MLR (cMLR) is introduced as a compromise for these two factors[126]. In cluster-wise MLR, a MLR model is constructed for each cluster. Each MLR model captures a local relationship between **x** and y, which are expected to be linear, according to a QSPR/QSAR assumption. The assumption is that *similar compounds have similar properties*[127]. Although this is not entirely true, when thinking about the currently activated research region of activity cliff[128], there are still many QSPR/QSAR researches based on this hypothesis.

The proposed methodology has several advantages compared to deriving the posterior PDF based on a MLR model. In addition to being able to construct QSPR/QSAR models with high predictability, posterior PDFs show high density around the region that is supposed to exhibit desired y values near training data. When acquiring a posterior distribution, the prior distribution is represented with a Gaussian mixture in both methodologies. In section 3-2 , a proposed methodology is evaluated from various points of view. Results from several case studies are examined in section 3-5 . First, predictability of cMLR as a regression model is examined using the dataset of human adrenergic receptors. Second, predictability is tested using the simulation dataset having nonlinear relationship between **x** and y. In the second case study, posterior distributions with various y values as well as a prior distribution are scrutinized by comparing an author's previously proposed methodology. Finally, whether a derived posterior distribution can be employed as a criterion of AD or not is discussed based on the results of inverse QSPR analysis using an aqueous solubility dataset.

3-2 Methodologies

The proposed methodology here is Gaussian mixture models and cluster-wise multiple linear regression (GMMs/cMLR). For obtaining a posterior distribution as a mixture of Gaussians, cluster-wise MLR and a GMM are combined. The procedure of deriving a posterior distribution is as follows. First, the prior distribution of training data, which is $p(\mathbf{x})$, is represented as GMMs. Then, for each cluster, a QSPR/QSAR model is constructed using

MLR with an ordinary least square procedure (OLS). The OLS assumes a Gaussian error at every predicted y value for **x**. Hence, estimating a least square solution based on training data means that the conditional distribution of y given a value of **x** (p(y|**x**)) can be modeled. Finally, the posterior distribution of **x** given a y value (p(**x**|y)) is retrieved by Bayes' theorem. The obtained posterior distribution is still a GMM. Therefore, analytical (closed-form) solution is tractable as well as it can handle multimodal data distribution features inherent to GMM. The derivations in the following sections are based on a paper by Miyao et al[126].

3-2-1 GMMs: p(x)

A GMM is a parametric model represented by a mixture of Gaussians[129]. It is a standard methodology when considering parametric density estimation in unsupervised learning. Density with GMMs is formulated as:

$$p(\mathbf{x}) = \sum_{k=1}^{M} \pi_k N(\mathbf{x}|\, \boldsymbol{\mu}_k, \Sigma_k\,) \tag{3.1.}$$

where

$$\sum_{k=1}^{M} \pi_k = 1 \tag{3.2.}$$

In Eq. (3.1.), *N* represents a Gaussian distribution with mean vector $\boldsymbol{\mu}_k$ and covariance matrix $\sum_k$. π_k is the weight for the k-th Gaussian. Parameters to be estimated are $\boldsymbol{\mu}_k$, $\sum_k$, and π_k. These parameters are efficiently estimated with an expectation-maximization (EM) algorithm[130]. The number of Gaussians M is a hyper parameter, which should be determined before optimizing other parameters. Several criteria have been proposed in order to determine M, such as Akaike information criterion (AIC)[131], Bayesian information criterion (BIC)[132], and so on.

3-2-2 GMMs/cMLR: p(y|x)

To represent nonlinear relationship between **x** and y, the combination of MLR models have been studied for many years. One way to combine MLR models is to combine the conditional probabilities of multiple MLRs and find their adequate parameters based on a maximization likelihood criterion. Multiple MLR models constructed with this methodology may overlap[133,134]. Consequently, they capture the complex (unusual) relationship between **x** and y, such as in the case that single **x** corresponds to multiple ys. Another way of combining MLR models is to first split the training dataset into several parts, then construct a MLR model in each region[135,136]. To determine the region where a single MLR model should be constructed, other machine learning techniques are employed, such as k-means[135].

GMMs/cMLR is classified into the latter category. Clusters are determined by GMMs, meaning the number of MLR models is the same as that of clusters. In the k-th cluster, the MLR model is

$$p(y|\mathbf{x}, z_k) = N\left(y|\mathbf{a}_k^{\mathrm{T}}\mathbf{x} + b_k, \sigma_k^2\right) \quad (3.3.)$$

where $\mathbf{a}_k$ is the regression coefficient vector, b_k is a bias term, and σ_k^2 is the variance (error) of the k-th cluster. z_k is an indicator variable, representing whether the PDF is for the k-th cluster or not. These parameters are all estimated with training data by solving least square regressions in a closed-form expression.

3-2-3 Inverse QSPR/QSAR Model: p(x|y)

Inverse model can be derived analytically with the help of Bayes' theorem. The goal in this subsection is to formulate p(**x**|y) as a GMM. When combining the contribution of each cluster, it can be written as

$$p(\mathbf{x}|y) = \sum_{k=1}^{M} p(\mathbf{x}, z_k|y) \quad (3.4.)$$

where z_k is the same indicator variable as in Eq.(3.1.). Eq. (3.4.) is equal to

$$p(\mathbf{x}|y) = \sum_{k=1}^{M} p(\mathbf{x}|y, z_k)p(z_k|y) \quad (3.5.)$$

by the product rule of probability. Roughly speaking, $p(z_k|y)$ in Eq. (3.5.) represents the weight of posterior distribution of the k-th cluster. Applying Bayes' theorem to Eq. (3.5.) leads

$$p(\mathbf{x}|y) = \sum_{k=1}^{M} p(\mathbf{x}|y, z_k)\frac{p(y|z_k)p(z_k)}{\sum_{l=1}^{M} p(y|z_l)p(z_l)} \quad (3.6.)$$

where the posterior PDF of z_k can be transformed into the corresponding prior information and likelihood. Eq. (3.1.) can be rewritten as

$$p(\mathbf{x}) = \sum_{k=1}^{M} p(z_k)p(\mathbf{x}|z_k) \quad (3.7.)$$

The remaining task is, accordingly, to derive $p(\mathbf{x}|y,z_k)$ and $p(y|z_k)$, because $p(z_k)$ equals π_k. There are well known two Gaussian distribution equations directly answering the questions above as follows[129]:

$$p(y|z_k) = N\left(y\middle|\boldsymbol{a}_k^T\boldsymbol{\mu}_k + b_k, \sigma_k^2 + \mathbf{a}_k^{\mathrm{T}}\Sigma_k\mathbf{a}_k\right) \quad (3.8.)$$

$$p(\mathbf{x}|y, z_k) = N(\mathbf{x}|\mathbf{m}_k(\mathrm{y}), \Lambda_k) \quad (3.9.)$$

where,

$$\mathbf{m}_k(\mathrm{y}) = \Lambda_k\{\sigma_k^{-2}\mathbf{a}_k(y - b_k) + \Sigma_k^{-1}\boldsymbol{\mu}_k\} \quad (3.10.)$$

$$\Lambda_k = \left(\Sigma_k^{-1} + \sigma_k^{-2}\mathbf{a}_k\mathbf{a}_k^{\mathrm{T}}\right)^{-1} \quad (3.11.)$$

The mean of $p(y|z_k)$ is the prediction value for the mean of **x** is intuitive. The variance of $p(y|z_k)$ is a result of incorporating the regression coefficient and **x** information. Eq. (3.9.) shows the posterior PDF of **x** in the k-th cluster, which is the most important part in the methodology. The interesting point in this equation is that $\mathbf{m}_k(y)$ is biased towards the objective variable value y, while it still holds the prior distribution by considering $\boldsymbol{\mu}_k$. Roughly speaking, in Bayesian perspective, posterior inherits prior density feature: *posterior density* ∝ *likelihood* × *prior density*. Therefore, the posterior PDF of **x** can compromise with both prior distribution and an input y value. By replacing $p(\mathbf{x}|y,z_k)$ and $p(y|z_{k\text{ or }l})$ in Eq. (3.8.) and Eq. (3.9.) with the corresponding parts in Eq. (3.6.), $p(\mathbf{x}|y)$ is derived analytically.

Here a comment on data splitting is required. The regression coefficient as well as the bias term in each cluster should be estimated with the least square regression framework. The training data in each cluster is determined by the GMM that is constructed with all training data. Every training sample is assigned to one of the clusters, where it has the highest density. Since the clustering methodology gathers similar samples in one cluster, and the number of samples decreases, correlation among variables is expected to increase more in a cluster than among all data. Therefore, MLR with all variables may not be appropriate for modeling. In order to cope with collinearity among variables in localized samples, variable selection is recommended. The variable selection can be easily taken part in this methodology. As a result of variable selection, the unused variables for constructing a regression model in a cluster are set as 0 in Eq. (3.8.) and Eq. (3.9.). Posterior PDF for these variables is the prior distribution of them. Therefore, the derived formula is still valid after applying a variable selection procedure to the clusters.

3-3 Overview of the Methodology

To clearly understand the concept proposed here, the difference between the proposed methodology (GMMs/cMLR) and the previously proposed one by Miyao *et al.*[100] (GMMs and MLR) is described. The only difference is that the proposed methodology assumes

multiple MLR models, whereas GMMs and MLR employ a single MLR model. The difference is highlighted in **Figure 3-1** and **Figure 3-2**. In both figures, the relationship between one-dimensional descriptor x and one objective variable y is modeled. The part of constructing GMMs is common in the two methodologies (*i.e.* using the same number of Gaussians, mean vectors, covariance matrices, and weights of Gaussians). In the following explanation, the mixture of two Gaussians represents the prior distribution of x drawn as a blue curve in both figures. In **Figure 3-1**, the left picture shows the regression model in GMMs and MLR, which is a simple MLR. On the right picture, GMMs/cMLR constructs two MLR models for two Gaussians. These localized MLR models are merged into one regression model. In **Figure 3-2**, a desired y value is input in the two models (gray dotted line), and inverse analyses are conducted to obtain the posterior distribution of x given the y value (red dotted lines). On the left side of **Figure 3-2**, the two posterior Gaussians of x are gathered toward the center of the data, whereas GMMs/cMLR can capture two regions that may be responsible for the y value on the right side of **Figure 3-2**.

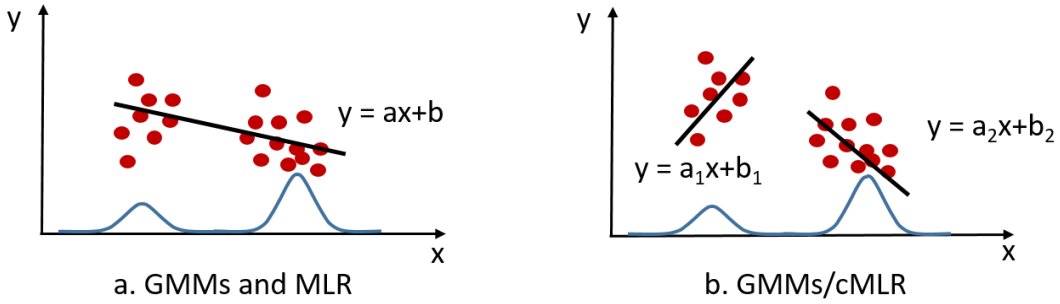


Figure 3-1 Illustration of the difference between GMMs/cMLR and GMMs and MLR as regression methodology. a) GMMs and MLR, and b) GMMs/cMLR. Detailed explanation is on the main body.

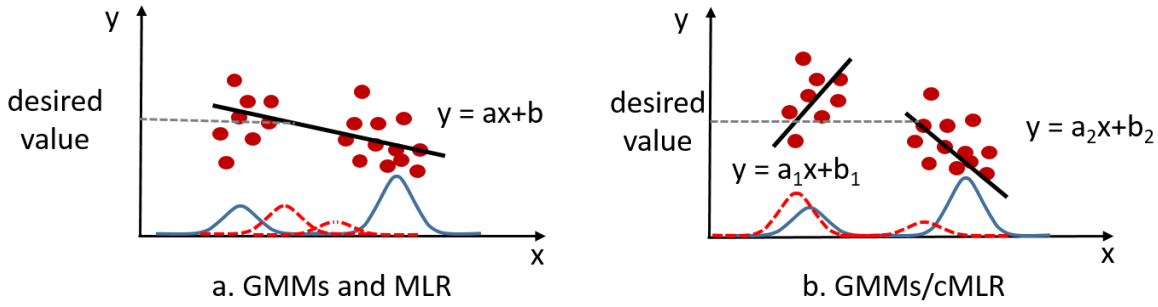


Figure 3-2 Illustration of the difference between GMMs/cMLR and GMMs and MLR in inverse analysis. a) GMMs and MLR, and b) GMMs/cMLR. Detailed explanation is on the main body.

3-4 Implementation

GMMs are constructed by *mclust ver.4.4*[137] in programing language *R*. Other than calculating GMMs, in house *R* scripts were written so to make use of *mclust* data structures.

3-5 Case Studies

3-5-1 Affinity Prediction for Four Alpha-Adrenergic Receptors

3-5-1-1 Dataset

Ligand data for four human alpha-adrenergic receptors were extracted from the GVK database[106]. The four receptors are human alpha-1B, alpha-1D, alpha-2A, and alpha-2C. Objective variable y is the logarithm of reciprocal of K_i (inhibition constant). There are 1062 compounds satisfying the criterion. From the original structure pool, 10 dimers, which are outliers based on visual inspection of the two-dimensional map with PCA, were eliminated. The number of compounds in the curated dataset was 878 after passing the in house descriptor calculation module, *i.e.* DescriptorCalculator in Molgilla. The training dataset consists of randomly chosen 600 samples, and the test dataset of the remaining 278 samples.

13 descriptors were chosen for constructing QSAR models by trial and error. The select descriptors were number of rings (CIC), number of five membered rings (R05), number of rotatable bonds (nBR), nCH_2R_2, nCH_3X, n=O, $ArNR_2$, Randic connectivity index (X1), topological polar surface area (TPSA), number of hydrogen bond donors (nHBD), STDPs between aromatic rings (RR), number of hydrogen bond acceptors (nHBA), and number of aromatic rings (aR). Definition of the descriptors is on the table in Appendix E. All samples in the dataset were randomly split into a training and test dataset.

3-5-1-2 GMM Construction and Regression Models Comparison

The same GMM was used in all QSAR models for the four alpha adrenoceptors. Parameters: $\boldsymbol{\mu}_k$, $\sum_k$, and π_k ($1 \leq k \leq 15$) were optimized by the EM algorithm. The parameter, in *mclust*, determining the shape of the covariance matrices was set "VVV", meaning all the covariance matrices can have different shape, volume, and orientation. BIC values against the numbers of Gaussians were shown in **Figure 3-3**. The maximum value was -12685.53 at four Gaussians. Hence, the optimal number of Gaussians was 4 based on BIC.

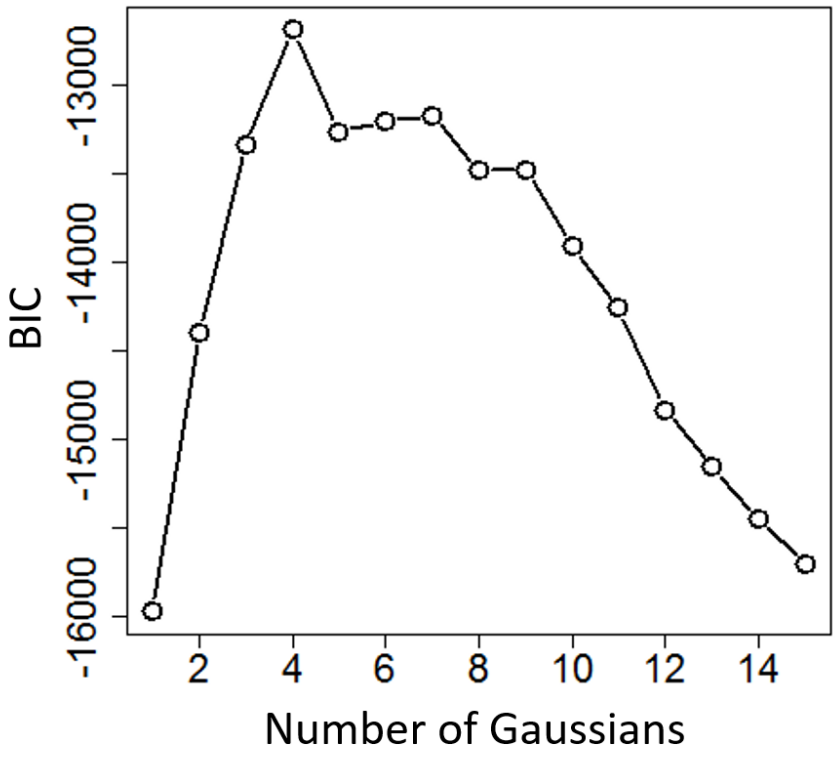


Figure 3-3 BIC value against the number of Gaussians for different covariance parameters.

Means of four Gaussians in the prior distribution p(**x**) are shown in **Table 3-1**. Cluster 1 (C1) seems to gather smaller sized compounds because it has the smallest CIC and X1 mean values. On the other hand, C2 and C4 collect larger compounds based on the values of CIC and X1. The mean of C4, however, has smaller TPSA value than that of C2. It means that there are lipophilic compounds gathered in C4 compared to C2. Affinity prediction of each target is carried out by a MLR model in each cluster. Thus, it is important to know the features of each cluster before applying regression models.

Table 3-1 Means of Gaussians in p(**x**)

	CIC	R05	nBR	CH_2R_2	CH_3X	=O	$ArNR_2$	X1	TPSA	nHBD	RR	nHBA	aR
C1*	3.0	1.0	2.9	2.4	0.2	0.3	0.0	9.2	42.1	1.4	1.6	2.7	1.7
C2	4.3	0.3	9.8	1.8	1.1	2.3	0.8	18.4	95.2	1.6	17.8	8.5	2.5
C3	3.8	1.0	6.4	4.2	0.2	2.1	1.0	14.2	60.4	0.4	0.0	7.3	1.0
C4	4.7	0.9	7.0	1.6	0.9	0.8	0.5	16.0	57.8	0.7	15.6	6.0	2.5

C1: Cluter1

Four MLR models were constructed for each objective variable. All training samples were categorized into the four clusters, *i.e*. C1, C2, C3, and C4. For constructing each MLR model, only the training samples in the corresponding cluster were used. C1 had 92 samples, C2 160, C3 141, and C4 207 for training models. When predicting pK_i of a new sample, only one MLR model in the cluster, which gives the highest density, is responsible for prediction. 61 test samples were classified in C1, 73 in C2, 45 in C3, and 99 in C4. The results of data prediction were shown on **Table 3-2**. As expected, introducing cluster-wise MLRs contributed to the increment of model predictability, which is judged from R_{pred}^2, $RMSE_{pred}$ for test data. In all the four cases, GMMs/cMLR showed the higher R_{pred}^2 and the lower $RMSE_{pred}$, meaning GMMs/cMLR could manage an overfitting problem to the training samples. Predicted values are plotted against the observed ones (yy-plot) in **Figure 3-4**.

Table 3-2 Results of model construction by GMMs/MLR and MLR methodology[126]

Alpha1B

	R^2	RMSE	R_{pred}^2	$RMSE_{pred}$
MLR	0.2887	0.7470	0.3551	0.7342
GMMs/cMLR	0.5729	0.5788	0.5336	0.6231

Alpha1D

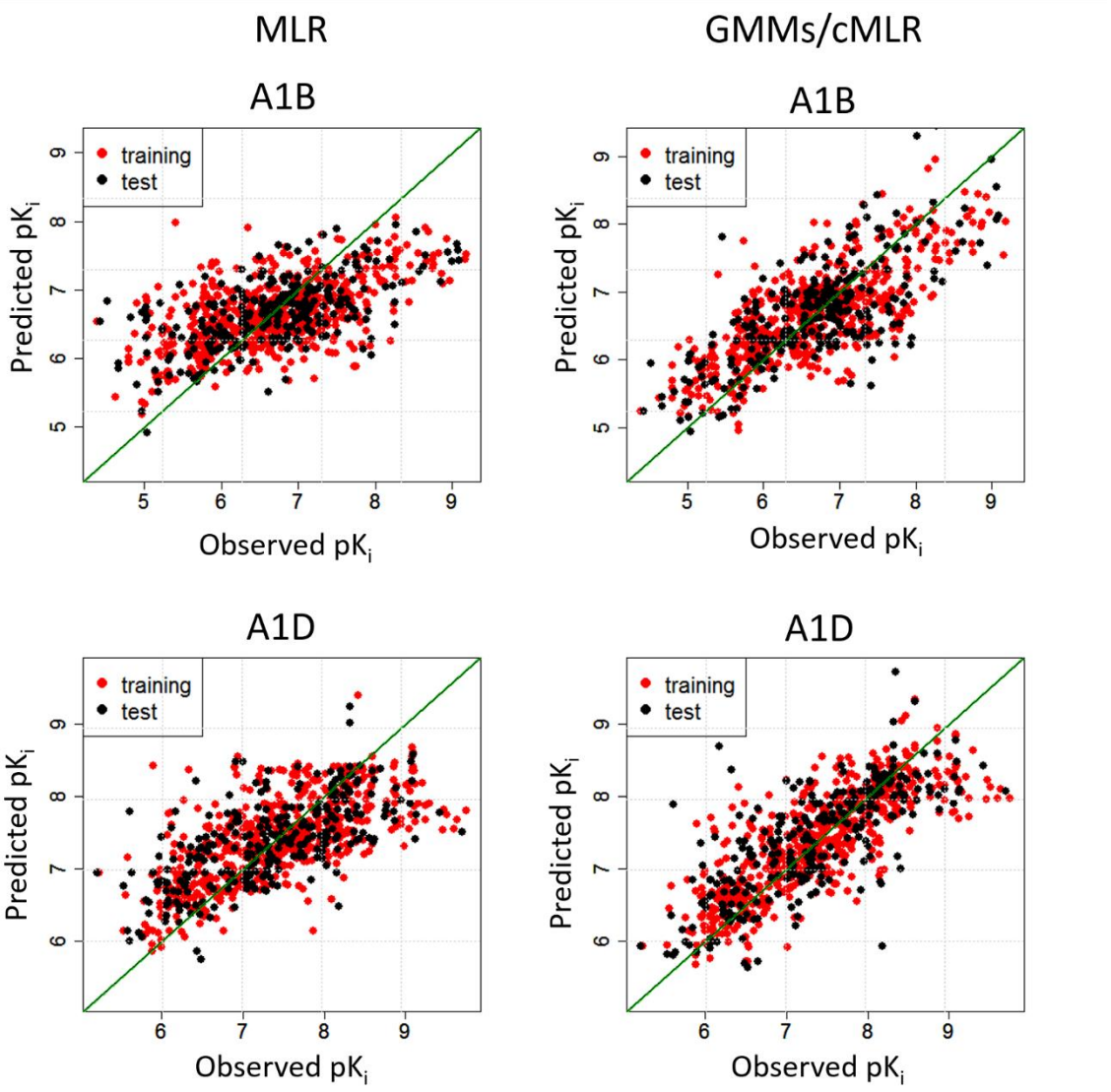
	R^2	RMSE	R_{pred}^2	$RMSE_{pred}$
MLR	0.4425	0.6598	0.4235	0.6714
GMMs/cMLR	0.6523	0.5210	0.5352	0.6028

Alpha2A

	R^2	RMSE	R_{pred}^2	$RMSE_{pred}$
MLR	0.7214	0.4963	0.6859	0.5211
GMMs/cMLR	0.8016	0.4188	0.7550	0.4601

Alpha2C

	R^2	RMSE	R_{pred}^2	$RMSE_{pred}$
MLR	0.6394	0.5511	0.6237	0.5586
GMMs/cMLR	0.7476	0.4610	0.6759	0.5185


MLR
GMMs/cMLR
A1B
A1B
A1D
A1D
training
test
Predicted pK$_i$
Observed pK$_i$

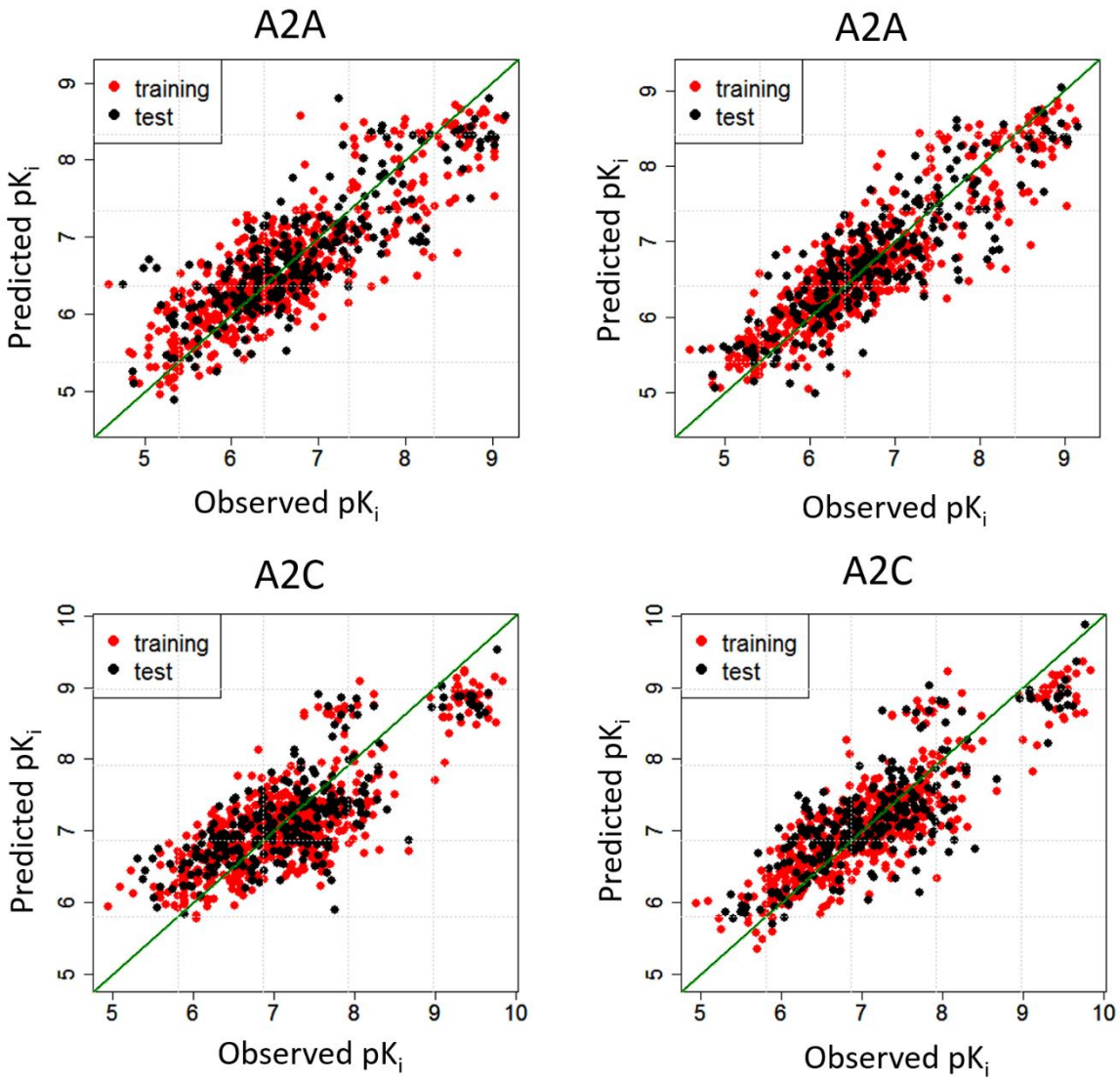


Figure 3-4 Predicted pK_i value against observed value for the four alpha adrenoceptor data. A1B is alpha-1B, A1D alpha-1D, A2A apha-2A, and A2C alpha-2C.

One of the strong points of linear regression compared to non-linear one is that regression coefficients of independent variables can be directly used for interpreting the model. Although they do not always give a proper interpretation due to correlation among variables, they give a direction for the users of the model toward the point at which designed molecules (chemical structures) exhibit better property or activity. Therefore, it is worthwhile to mention the difference of the regression coefficients between MLR and GMMs/cMLR. Alpha 1B shows the most significant differences in predictability (R_{pred}^2 0.3551 by MLR and 0.5336 by GMMs/cMLR). Standard regression coefficients, and nominal ones are shown in **Table 3-3** and **Table 3-4**, respectively. There are some missing cells in both tables, meaning

they were unused variables for constructing a MLR model. This was expected, since some of the variables are discrete, such as occurrence of a substructure. By the clustering operation carried out, all samples in the cluster have the same value for a certain descriptor. For example, in C1, all the samples do not have a single $ArNR_2$ substructure, whereas all the samples in C3 have exact one value for that substructure. In the MLR model, sign for the R05 is negative, whereas it is positive in the C3 local model. By simply interpreting signs of correlation coefficient of a single MLR model, one might obtain the opposite instruction for molecular design to the corresponding GMMs/cMLR model.

Table 3-3 Standard regression coefficients of GMMs/cMLR (from C1 to C4) and MLR models for the alpha 1B adrenergic receptor.

Descriptor	C1	C2	C3	C4	MLR
CIC	0.20	0.08	-0.40*	0.20	0.11
R05	-0.06	-0.45*	0.38*	-0.31*	-0.23*
BR	0.11	-0.17	-0.34*	0.02	-0.10
CH_2R_2	-0.05	0.50*	0.17*	0.14*	0.36*
CH_3X	-0.05	0.23*	-0.10	0.36*	0.36*
=O	0.09	0.42*	-0.41*	0.44*	0.25*
X1	0.28	-0.78*	-0.15	-0.71*	-0.87*
TPSA	-0.08	-0.12	0.44	-0.11	0.05
HBD	-0.05	0.13	-0.62*	0.04	-0.07
nHBA	-0.31	0.37*	0.02	0.32*	0.23*
aR	-0.32*	1.15*		0.38*	0.15*
RR	-0.47*	-0.49*		0.05	0.48*
$ArNR_2$		0.23*		-0.13*	0.05

* is significant at $p < 0.1$ by t-test

Table 3-4 Regression coefficient of GMMs/cMLR (from C1 to C4) and MLR for the alpha 1B adrenergic receptor

Descriptor	C1	C2	C3	C4	MLR
CIC	0.18	0.12	-0.53*	0.23	0.11
R05	-0.06	-0.86*	0.52*	-0.48*	-0.31*
BR	0.04	-0.08	-0.14*	0.01	-0.03
CH_2R_2	-0.02	0.28*	0.07*	0.06*	0.14*
CH_3X	-0.07	0.23*	-0.17	0.33*	0.36*
=O	0.11	0.39*	-0.58*	0.52*	0.19*
X1	0.10	-0.36*	-0.08	-0.33*	-0.22*
TPSA	-0.00	-0.00	0.02	-0.01	0.00
HBD	-0.04	0.13	-0.71*	0.04	-0.06
nHBA	-0.15	0.24*	0.01	0.18*	0.08*
aR	-0.31*	1.71*		0.48*	0.15*
RR	-0.13*	-0.05*		0.01	0.04*
$ArNR_2$		0.28*		-0.20*	0.08
Intercept	6.32*	7.05	10.10*	8.32*	7.75*

* is significant at $p < 0.1$ by t-test

Each cluster has its own regression model, meaning predictability by GMMs/cMLR depends on into which cluster a test sample is assigned by a GMM. Predictability of each MLR for alpha-1B is explained on **Table 3-5**. From the table, the MLR model in C2 shows the best predictability among the four sub-models. This trend can be confirmed by visual inspection of yy-plot **Figure 3-5**.

Table 3-5 Predictability of each MLR for Alpha 1B prediction (from C1 to C4)

	R_{adj}^{2*}	RMSE	R_{pred}^2	$RMSE_{pred}$
GMMs/cMLR	0.5729	0.5788	0.5336	0.6231
C1	0.4212	0.556	0.500	0.579
C2	0.6186	0.529	0.571	0.559
C3	0.5084	0.553	0.437	0.612
C4	0.4542	0.639	0.434	0.695

R_{adj}^2: Adjusted R^2 by the degree of freedom

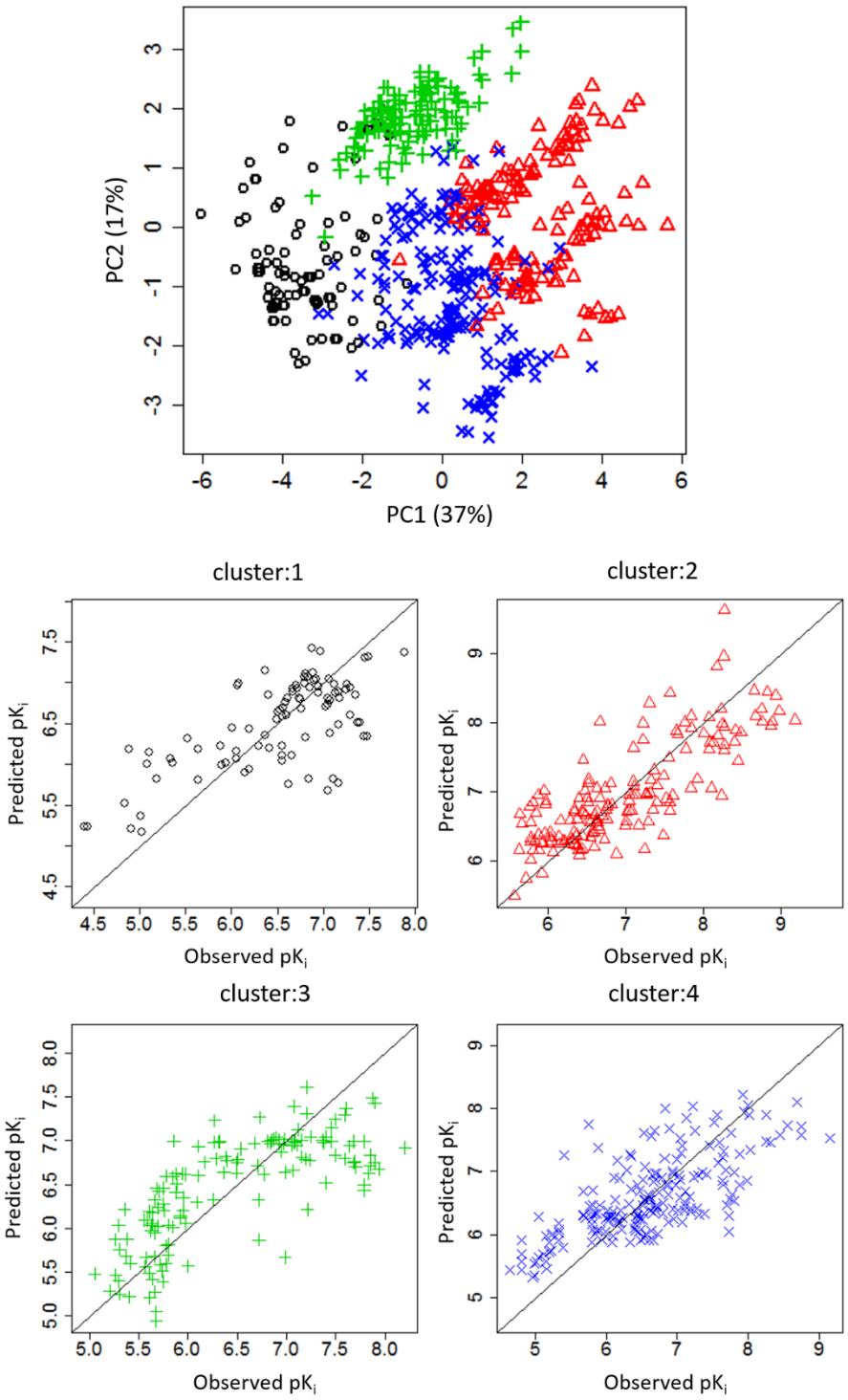


Figure 3-5 PCA map of the training dataset along with yy-plots of clusters. In these plots, only training samples are projected. For PCA map, numbers inside parentheses are contribution of axes (ratio of variance to axes).

3-5-2 Posterior distribution comparison using simulation dataset

3-5-2-1 Dataset

Simulation dataset was prepared for highlighting the difference between "GMM and MLR" and GMMs/cMLR in terms of posterior distribution p(**x**|y). Independent variable **x** was randomly sampled from the three two dimensional normal distributions. From each Gaussian, 100 training samples and 20 test samples were randomly sampled. The number of training samples was 300 and that of test samples was 60. Gaussian means for generating **x** are (3, -6), (4, 1), (2, 5), and covariance matrices are, respectively:

$$\Sigma_1 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 2 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 1 & 0.2 \\ 0.2 & 1 \end{bmatrix}, \Sigma_3 = \begin{bmatrix} 1 & -0.2 \\ -0.2 & 1 \end{bmatrix} \quad (3.12.)$$

The set of objective variable y was created by calculating the corresponding y values according to Eq.(3.13.), then adding Gaussian noise with variance 0.1.

$$y = \sin(0.5x_1)\cos(0.5x_2) + 0.1x_1 \quad (3.13.)$$

Training and test dataset can be seen on **Figure 3-6**.

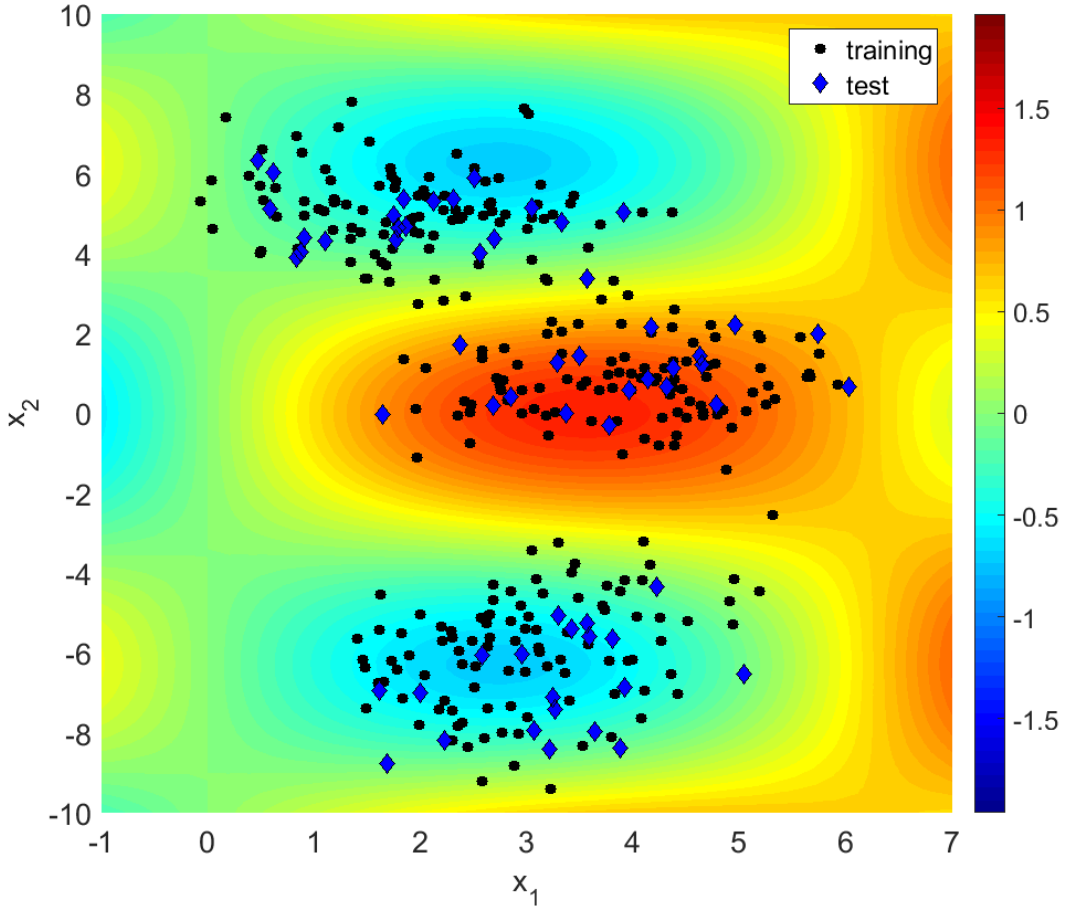


Figure 3-6 Training and test dataset. Background color represents y values

3-5-2-2 Results and Discussion

Two regression models were constructed with the training dataset. One was by GMMs/MLR, and the other by MLR. The number of optimized Gaussians in a GMM was three based on BIC (**Figure 3-7**). The maximum value was -1385.46 for three components. The chosen shape of covariance was EEE, meaning three Gaussians had identical covariance matrixes having elliptical distribution. The Gaussian means were C1 (2.99, -5.97), C2 (1.89, 5.06), and C3 (3.96, 0.78). The covariance matrix was $\begin{bmatrix} 0.844 & 0.064 \\ 0.064 & 1.308 \end{bmatrix}$. The weights for Gaussians were C1 0.330, C2 0.336, and C3 0.334. Assuming the acquired three Gaussians correspond to the closest ones used for data generation, 6 training samples were misclassified, and none of the test samples.

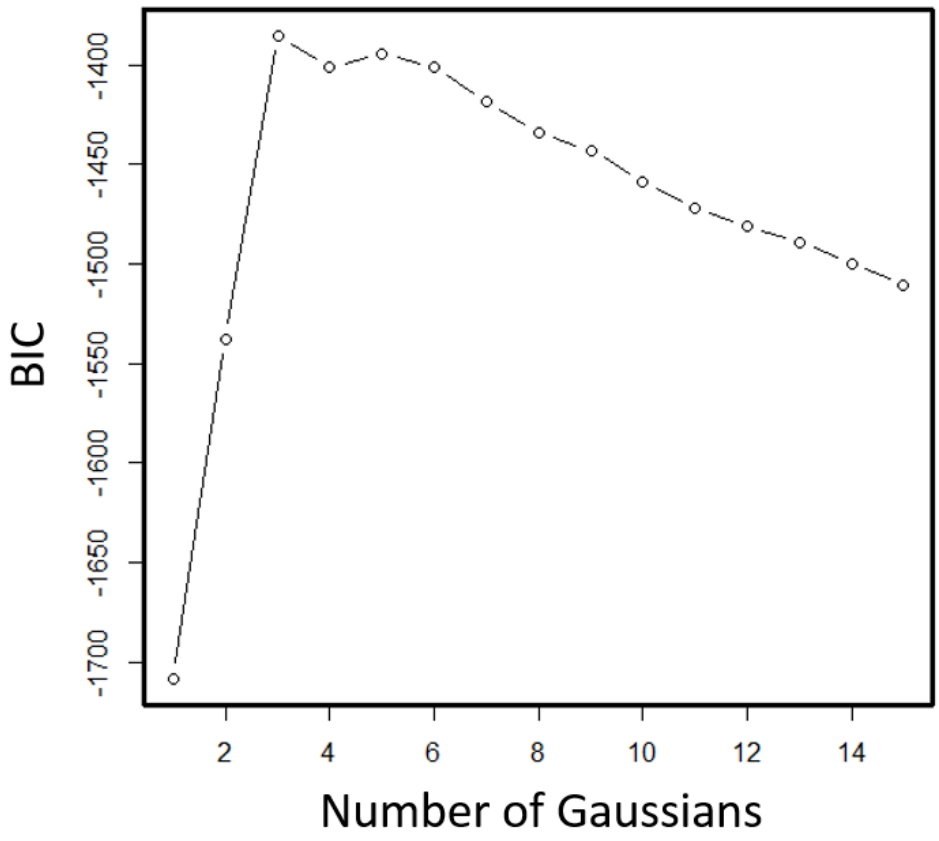


Figure 3-7 BIC value against the number of Gaussians

The model construction results by MLR and GMMs/cMLR were shown on **Table 3-6** and **Figure 3-8**. As expected, the MLR model resulted in poor predictability for the dataset having strong nonlinearity.

Table 3-6 Regression models predictability for simulation dataset

	R^2	RMSE	R_{pred}^2	$RMSE_{pred}$
MLR	0.357	0.572	0.293	0.570
GMMs/cMLR	0.932	0.186	0.927	0.183

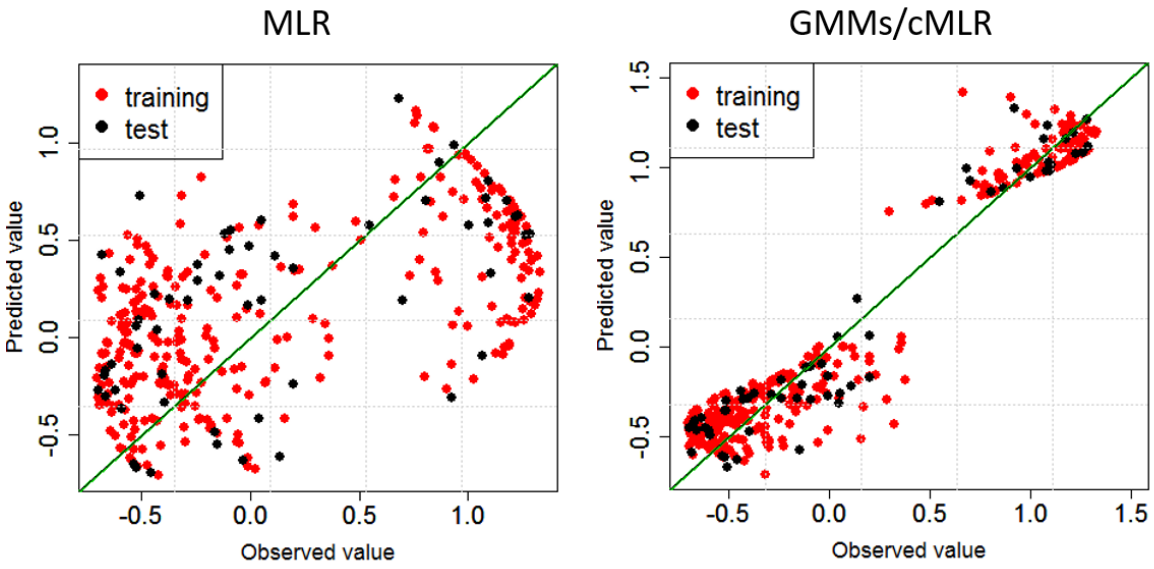
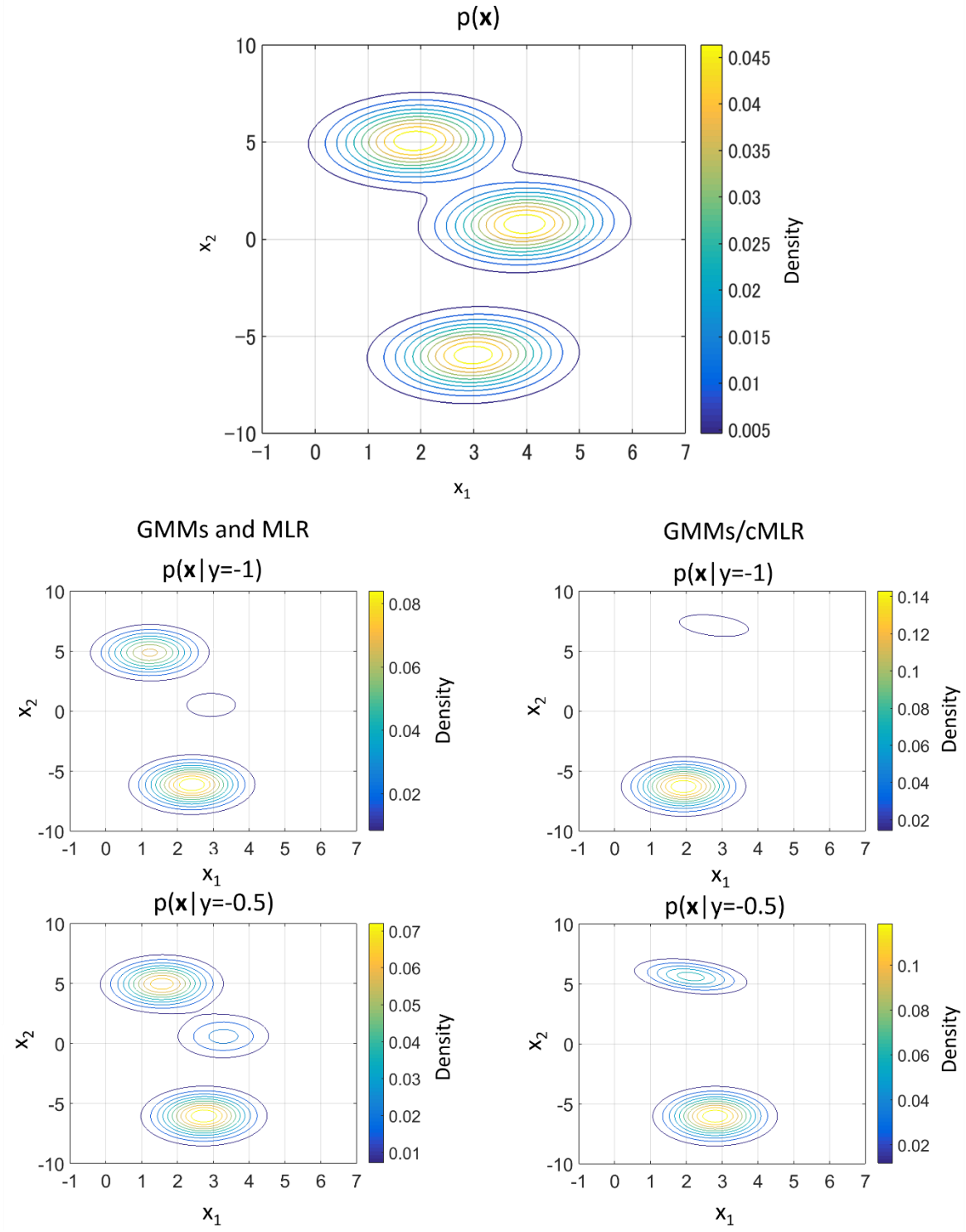


Figure 3-8 Predicted value against observed value by MLR and GMMs/cMLR models.

The next trial was to construct posterior PDFs of **x** given y values (p(**x**|y)). Contours of PDFs on a two-dimensional map are shown in **Figure 3-9**. True regions that should be captured by the posterior PDFs can be inferred from background colors on the top-left picture in the figure. PDFs by GMMs and MLR (the previous methodology) could not move drastically from a place where the prior PDF exists. This is because the MLR model for GMMs and MLR became rather flat, insensitive to x_2 changes, as a result of adjusting a linear function to the nonlinear multimodal data. The posterior distribution moves towards higher x_1 region as y value increases, as the MLR model is

$$y = 0.341x_1 + 0.044x_2 \qquad (3.14.)$$

As will be discussed in the next section, using three Gaussians may not capture the "true" regions where y values are close to a desired one. Take p(**x**|y =1) (on the bottom row in **Figure 3-9**) for example. The true regions that should be selected as high density area by the posterior PDF include upper and lower parts of the right side of the shown picture, *e.g.* around (x_1, x_2) = (7, 6) and (7, -6). This, however, is not correct when thinking about AD. There are no training samples around these points, where we do not have any measures to predict a y value for a new sample outside AD. As long as statistical models (*i.e.* regression models) are employed, AD is an important concept that we have to pay attention. In this section, quantitative evaluation of AD is not carried out, but the posterior distributions obtained with GMMs/cMLR seem to hold a distribution feature of training data by visual inspection.


p(x)
Density
x1
x2
GMMs and MLR
p(x|y=-1)
p(x|y=-0.5)
GMMs/cMLR
p(x|y=-1)
p(x|y=-0.5)

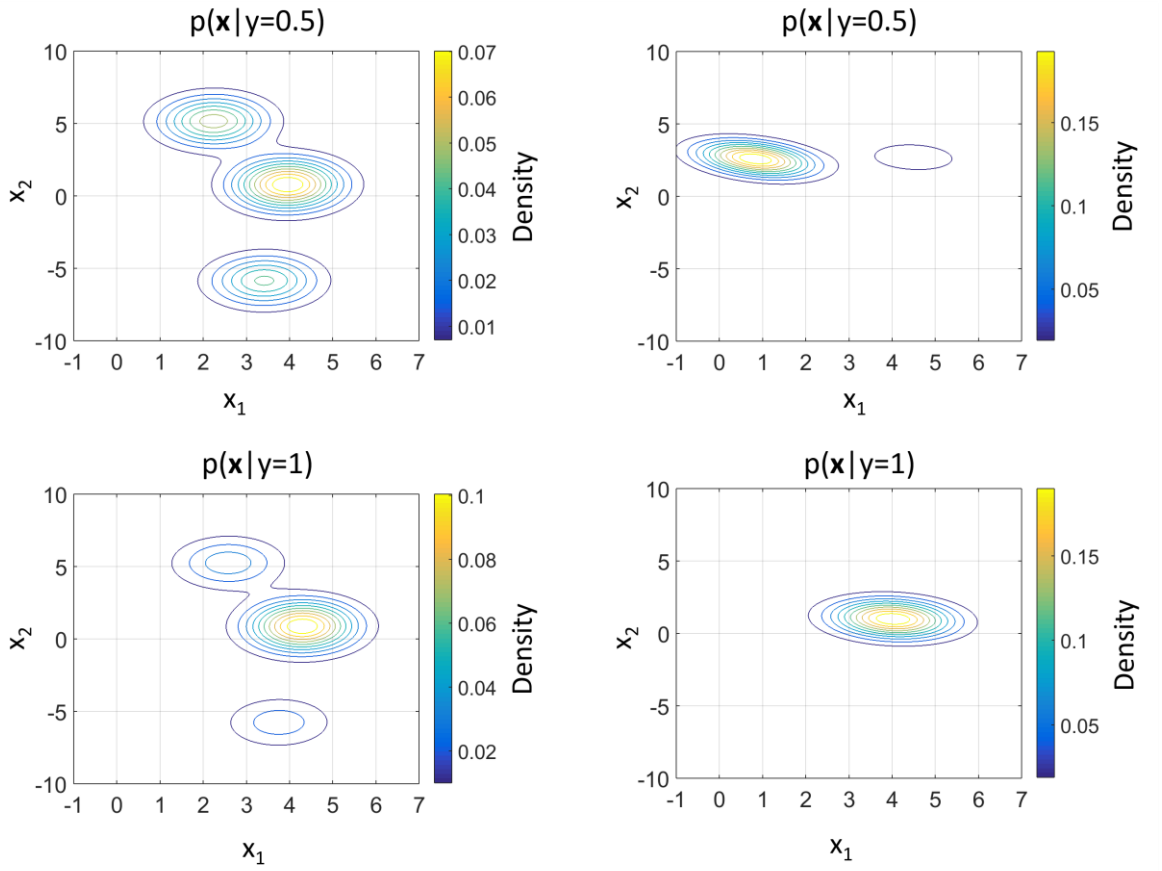


Figure 3-9 Contours of posterior PDFs of **x** given y = -1, -0.5, 0.5, and 1. On the top row, prior distribution with a GMM (right) is shown.

3-5-3 AD Evaluation with the Aqueous Solubility Dataset

An AD based on training data distribution in descriptor space is important. It is closely related to outlier detection[138]. If a novel **x** that differs significantly from any **x**s in training dataset is to be predicted, the model cannot make reliable prediction for it. There have been many researches carried out about this topic [139,61,13]. There are two ways of estimating model uncertainty: ensemble methodologies and density-based. The ensemble ones usually calculate the variance of predicted values for a novel **x** value with different models[61]. Based on the assumption that the variance is positively correlated with RMSE between the predict value and the true value for the corresponding sample, variance is employed as an AD criterion. In order to have multiple predicted values for a single sample, ensemble learning methodologies, such as random forests[140] and bagging[141], are employed. On the other hand, the density-based methodologies require modeling distribution of training datasets with unsupervised learning methodologies, such as k-means, GMMs, kernel-density estimation. The assumption in this approach is that the density of **x** is negatively correlated with RMSE values between predicted and true y values for a novel sample. Both approaches have been successfully demonstrated with several case studies, in which RMSE is plotted against

distance to model[142]. The following analysis and discussion related to AD start from the main premise that training data density is positively correlated with prediction reliability.

3-5-3-1 Dataset

Aqueous solubility dataset downloaded from a web site[105,9] was analyzed for evaluating posterior distribution as a measure of AD considering an objective variable value at the same time. The dataset consists of 1,290 diverse drug-like compounds annotated with measured aqueous solubility (logS) at 20-25 degrees Celsius [mol/L]. From the dataset, 8 pairs of duplicate structures determined with canonical SMILES format given by RDKit[120] were removed (total 16 structures). Then, only molecules that can be treated by the *DescriptorCalculator* module were selected for further analysis. The remaining 1,154 molecules were randomly divided into either a training dataset or a test dataset. Training dataset consists of 900 samples and test dataset of 254 samples. Distribution of objective variables for training and test datasets are shown in **Figure 3-10**.

Descriptors employed for this analysis were, molecular weight MW, number of hydrogen bond donor (nHBD) and number of hydrogen bond acceptor (nHBA) based on the Lipinski's rule[14], number of rings CIC, topological polar surface area TPSA, and number of rotatable bonds nBR.

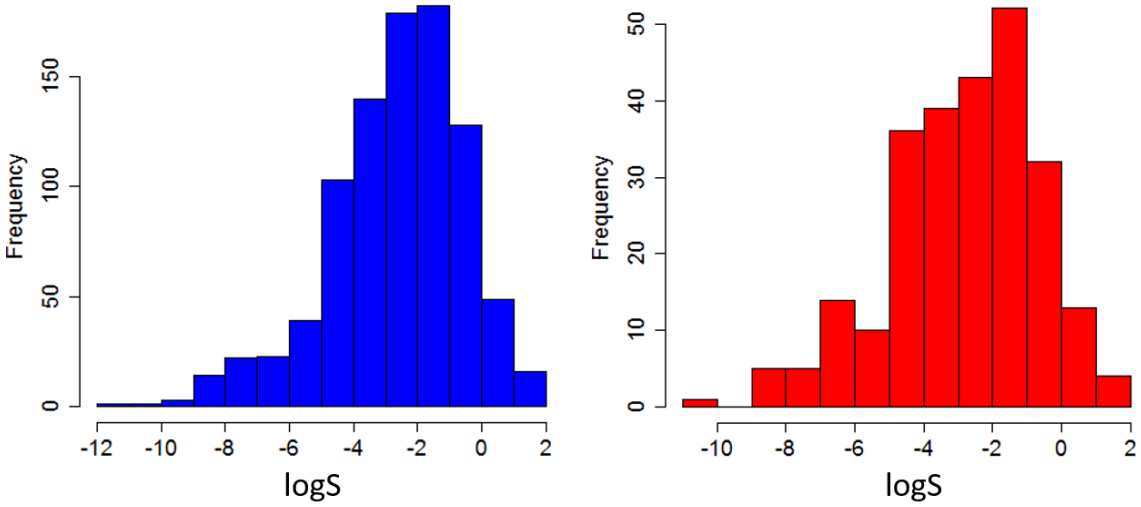


Figure 3-10 Histogram of logS values in training (blue) and test (red) datasets.

3-5-3-2 AD Evaluation

Before evaluating AD based on posterior PDFs, regression models with GMMs (GMMs/cMLR) were constructed. Based on BIC, the optimal number of Gaussians was 7 (BIC value was -4470.465). The BIC value against the number of Gaussians is shown on **Figure 3-11**.

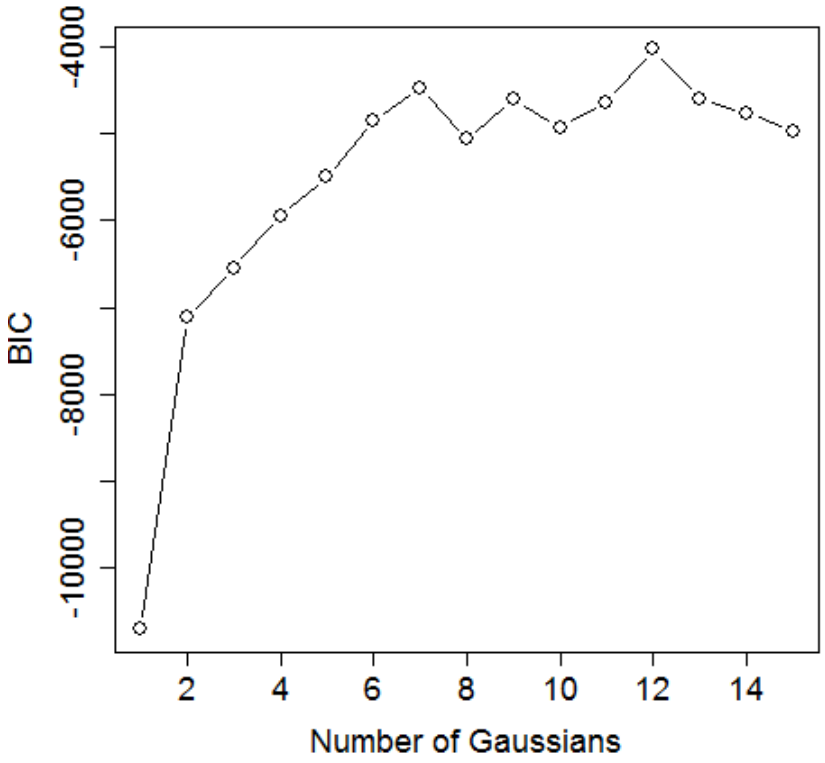


Figure 3-11 BIC value against number of Gaussians for logS dataset.

Consequently, 7 clusters had their MLR models in the GMMs/cMLR methodology. The numbers of training data categorized in one of the 7 clusters are cited on **Table 3-7**. Predictability of GMMs/cMLR as well as MLR are shown on

Table 3-8, and corresponding yy-plots are shown on **Figure 3-12**. The yy-plot of GMMs/cMLR looks better than that of MLR. The former prediction is more rigorous than the latter judging from these yy-plots. By combining this regression model ($p(y|\mathbf{x})$) with GMMs ($p(\mathbf{x})$), the posterior PDFs of $\mathbf{x}$ given various y values were constructed as closed-form GMMs ($p(\mathbf{x}|y)$).

Table 3-7 Number of the training data and the test data categorized in 7 clusters.

	C1	C2	C3	C4	C5	C6	C7
Training	215	129	147	38	150	134	87
Test	63	31	41	7	48	28	36

Table 3-8 Predictability of the MLR and the GMMs/cMLR models.

	R^2	RMSE	R_{pred}^2	$RMSE_{pred}$
MLR	0.736	1.061	0.722	1.131
GMMs/cMLR	0.853	0.791	0.854	0.820

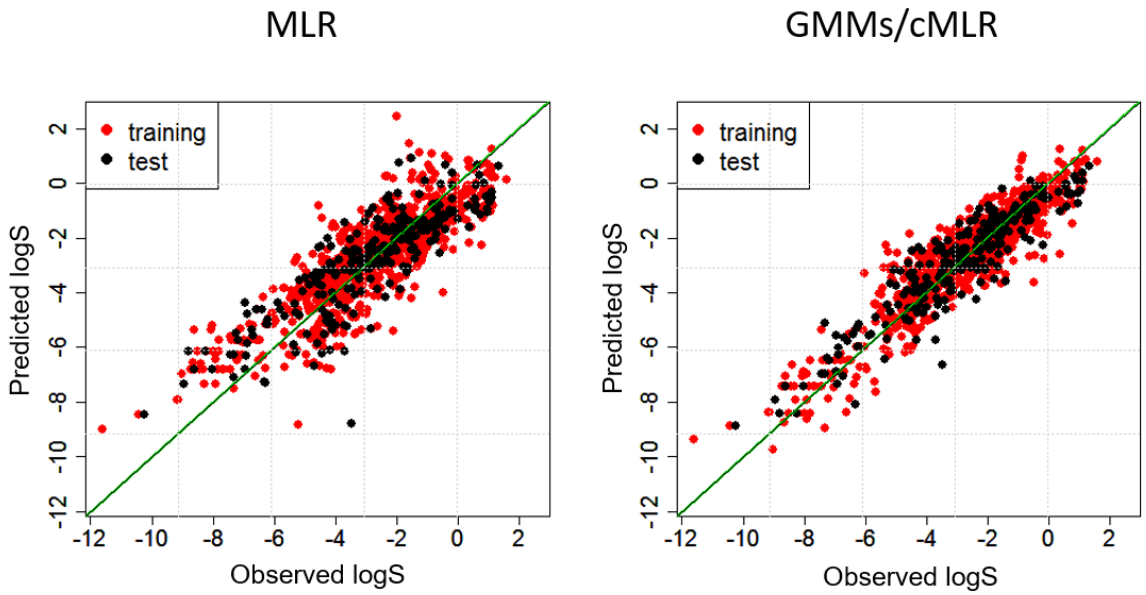
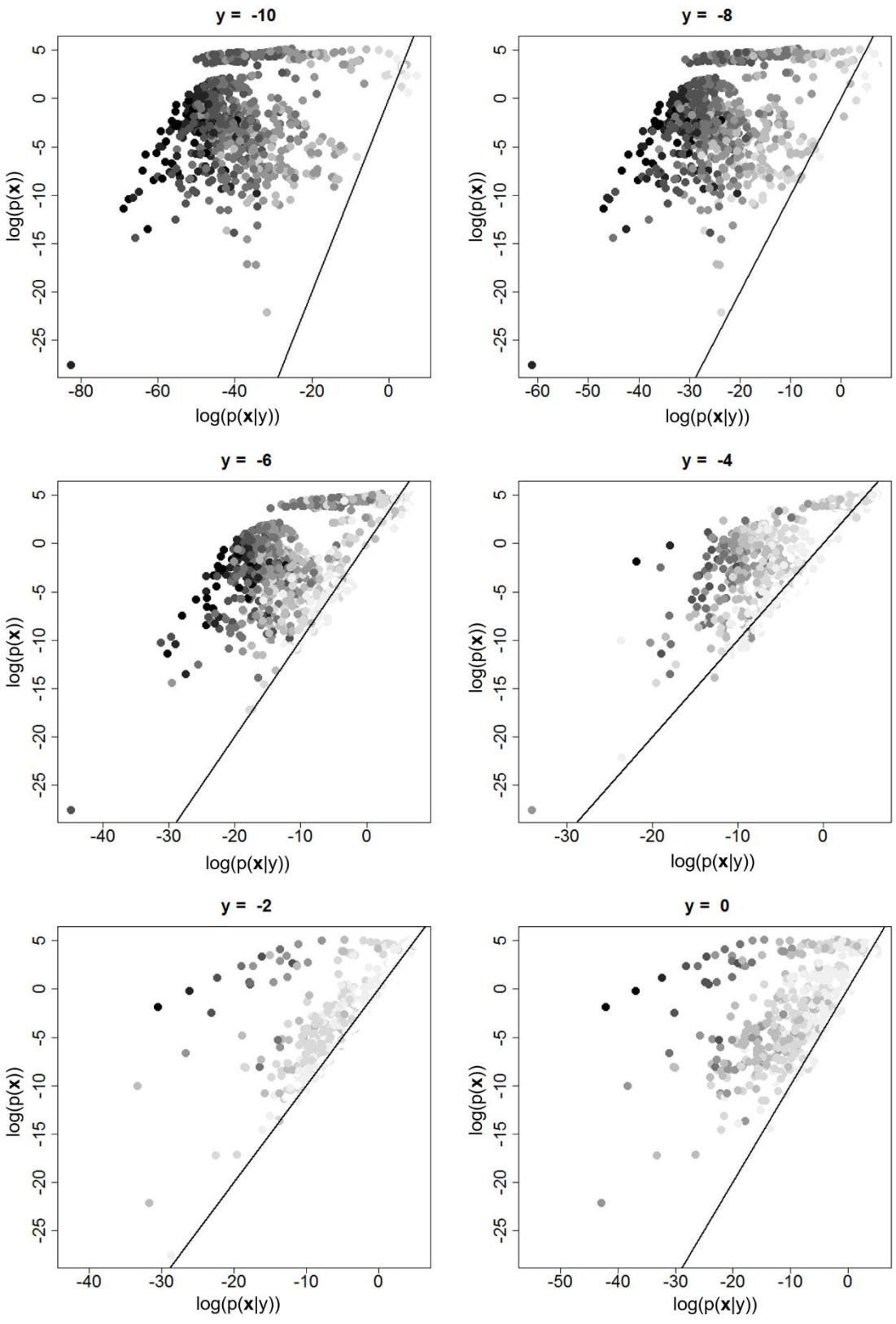


Figure 3-12 Predicted value against observed value in MLR and GMMs/cMLR models.

Posterior PDFs should have higher density in the area where the desired property value is exhibited. Furthermore, based on the premise of AD (i.e. we can rely on the predicted y values of samples located in high density areas), posterior PDFs are expected to inherit prior PDF features about training data density. This inference is natural based on Bayesian probability (i.e. $p(\mathbf{x}|y) \propto p(y|\mathbf{x})p(\mathbf{x})$). In this respect, $p(\mathbf{x})$ and $p(\mathbf{x}|y)$ were compared with each other in various y values. $p(\mathbf{x})$ is plotted against $p(\mathbf{x}|y)$ with different y values using the training dataset in Figure 3-13 (GMMs/cMLR) and Figure 3-14 (GMMs and MLR). Color intensity represents the difference between a target y value and a measured one. Thicker colors mean the samples have large difference, and lighter colors have small difference. In almost all pictures for training dataset, the former hypothesis can be confirmed. The higher $p(\mathbf{x}|y)$ of a sample becomes, the lesser the absolute error between the measured y value and the target y value becomes. Furthermore, it can be seen that $p(\mathbf{x}|y)$ inherited the $p(\mathbf{x})$ feature. No matter how close the measured y value of a training sample is to the desired one (y), $p(\mathbf{x}|y)$ does not excessively exceed the diagonal line. Therefore, it is fair to say that $p(\mathbf{x}|y)$ represents the likelihood that **x** exhibits the y value after considering AD for training dataset. Four compounds from the training dataset were extracted for visual inspection in **Figure 3-15**. In this case, the target y value was set to -10. Compound a has low $p(\mathbf{x})$, meaning this compound was likely out of AD for this QSPR model. The fact that compound b has low $p(\mathbf{x}|y)$ makes sense because it exhibits 1.09 logS, which is far from -10. Compounds c and d

exhibit similar p(**x**) values, but significant different p(**x**|y) values. For compound c, the measured logS was -1.87 whereas for compound d, that was 8.71. These results support the proposed methodology could represent the closeness to the target y value in p(**x**|y).

As for the test dataset, both methodologies, "GMMs and MLR" and GMMs/cMLR, succeeded in inheriting the feature of the prior distribution (**Figure 3-16** and **Figure 3-17**). In almost all pictures, the lower p(**x**) of a test sample is, the lower p(**x**|y) of that sample becomes. In almost all the pictures, dots are located above the diagonal line. p(**x**|y)s in both methodologies seem to express well the similarity to the target y value for the test dataset, in particular GMMs/cMLR. For -8 and -10,. GMMs/cMLR was able to present samples having high p(**x**) (over 0) in order based on the difference between the target y value and the measured one better than GMMs and MLR (p(**x**| y =-10) and p(**x**| y = -8)). Although "GMMs and MLR" was also able to distinguish samples based on the closeness to the target y value, the distinction is not as clear as that of GMMs/cMLR. For y = -4, p(**x**|y) values of samples in "GMMs and MLR" do not seem to change from the corresponding p(**x**) values. In other words, p(**x**|y) generated from GMMs and MLR did not reflect well the distance to the y value in it (or the degree of effect is small). From these visual inspections, GMMs/cMLR is better than GMMs and MLR for deriving posterior PDFs.


y = -10
y = -8
y = -6
y = -4
y = -2
y = 0
log(p(x))
log(p(x|y))

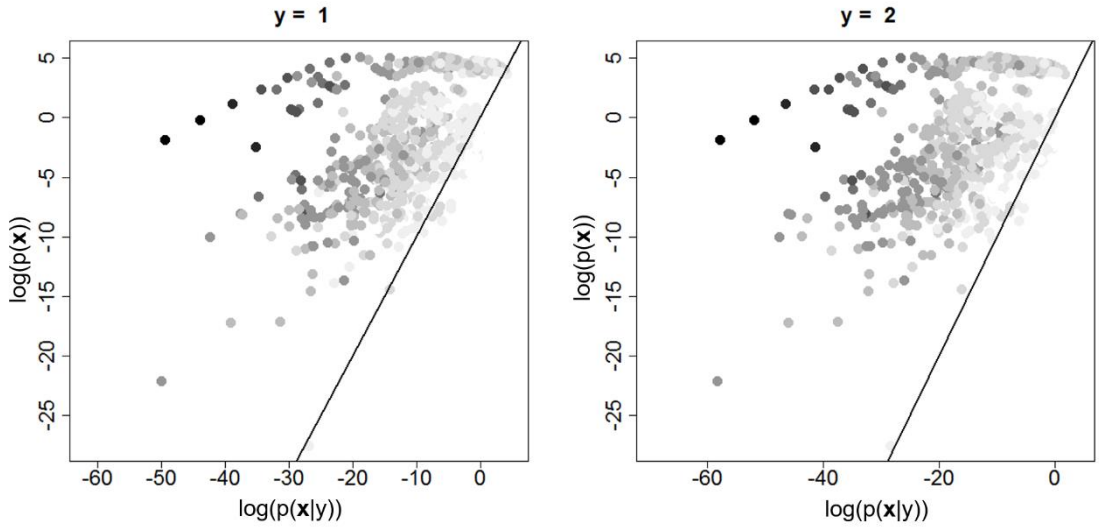
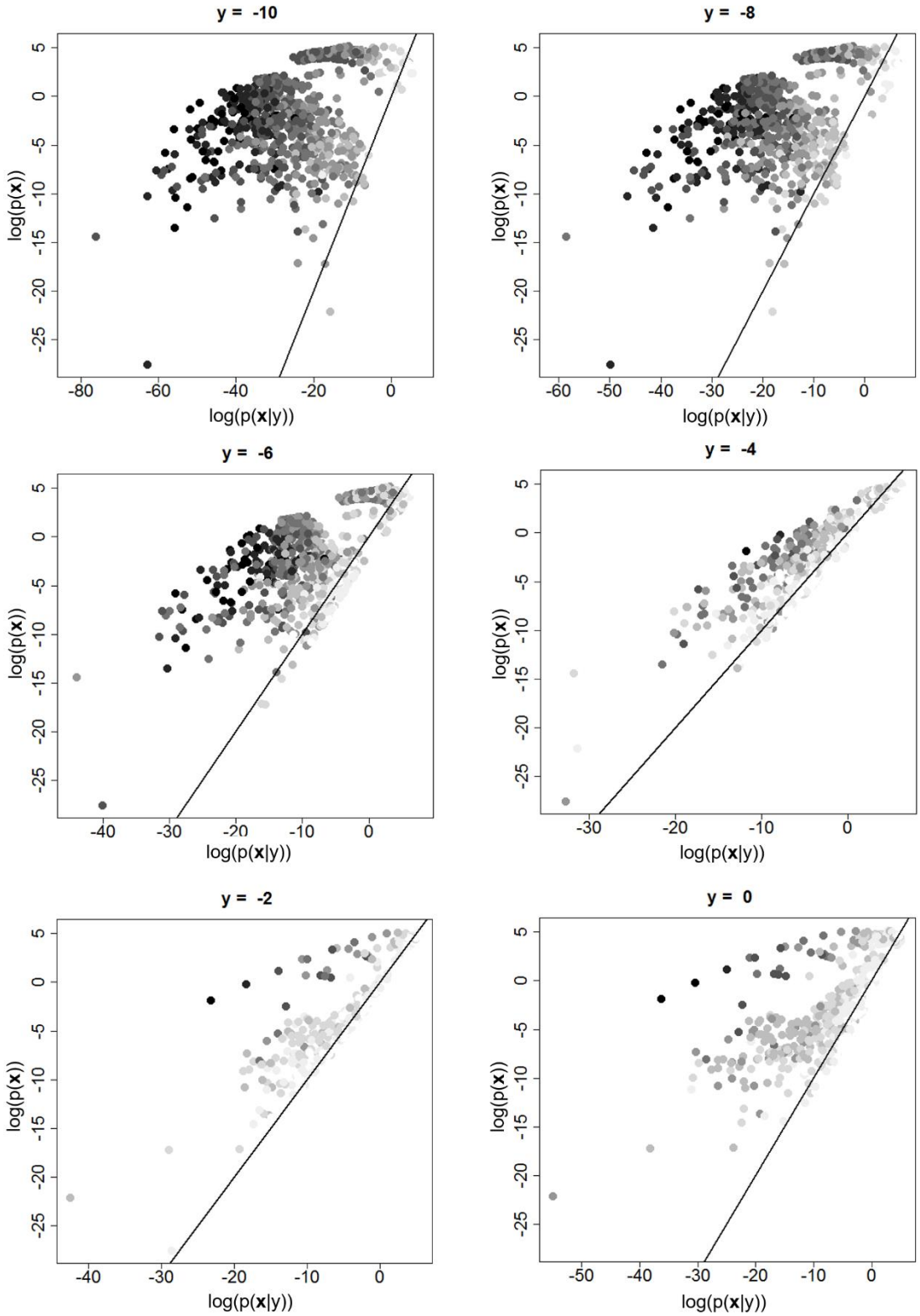


Figure 3-13 Logarithm of p(**x**) is plotted against logarithm of p(**x**|y) with various y values by GMMs/cMLR. Dots represent samples in the training dataset. Color scale represents the absolute error between the y value set in inverse analysis and the measured one. The thinner the color becomes, the less error the dot exhibits.


y = -10
y = -8
y = -6
y = -4
y = -2
y = 0
log(p(x))
log(p(x|y))

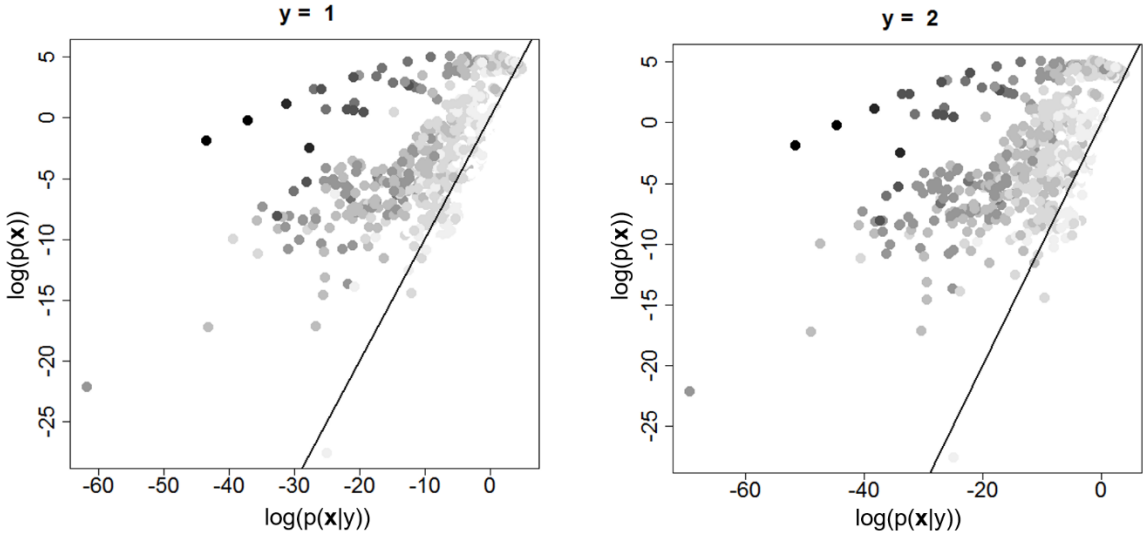


Figure 3-14 Logarithm of p(**x**) is plotted against logarithm of p(**x**|y) with various y values by GMMs and MLR. Dots represent samples in the training dataset. Color scale represents the absolute error between the y value set in inverse analysis and the measured one. The thinner the color becomes, the less error the dot exhibits. This figure corresponds to **Figure 3-13**.

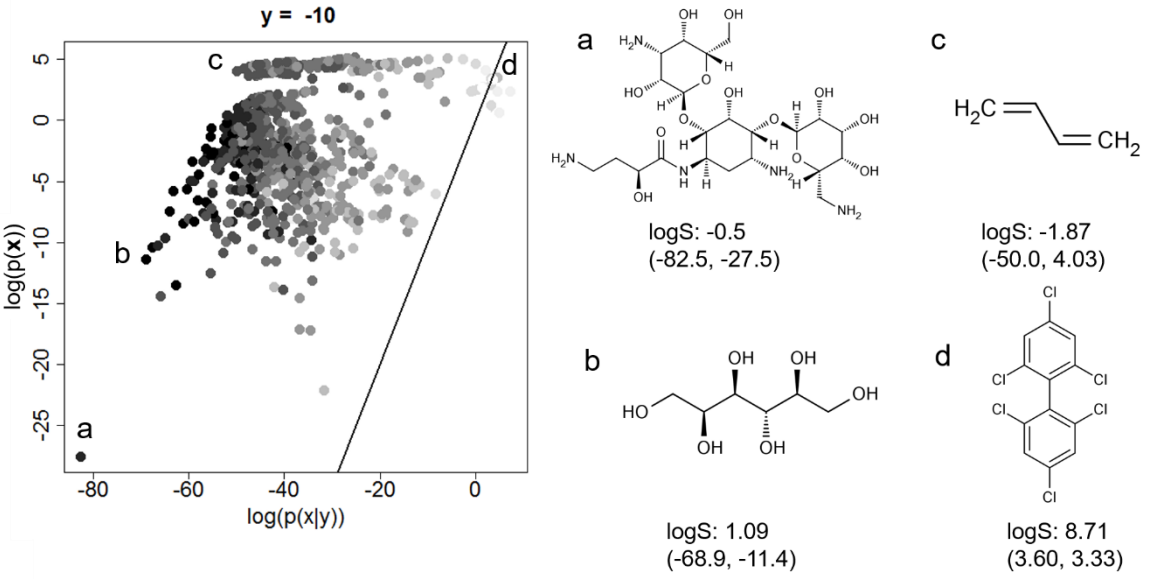
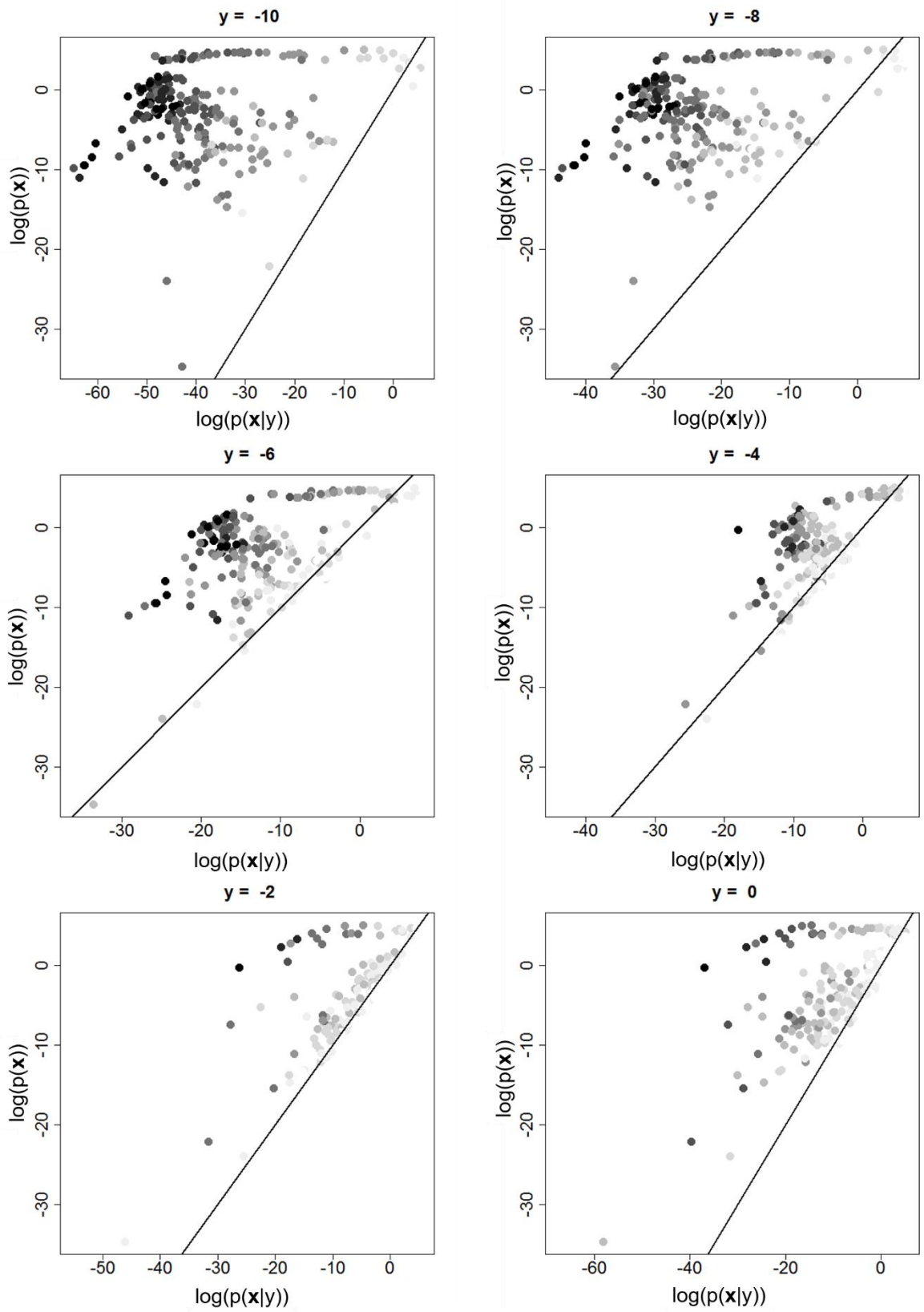


Figure 3-15 Selected compounds in the training dataset with the log(p(**x**))-log(p(**x**|y =-10)) plot by GMMs/cMLR. A set of coordinates in the parenthesis represents (log(p(**x**|y = -10)), log(p(**x**))).


y = -10
y = -8
y = -6
y = -4
y = -2
y = 0
log(p(x))
log(p(x|y))

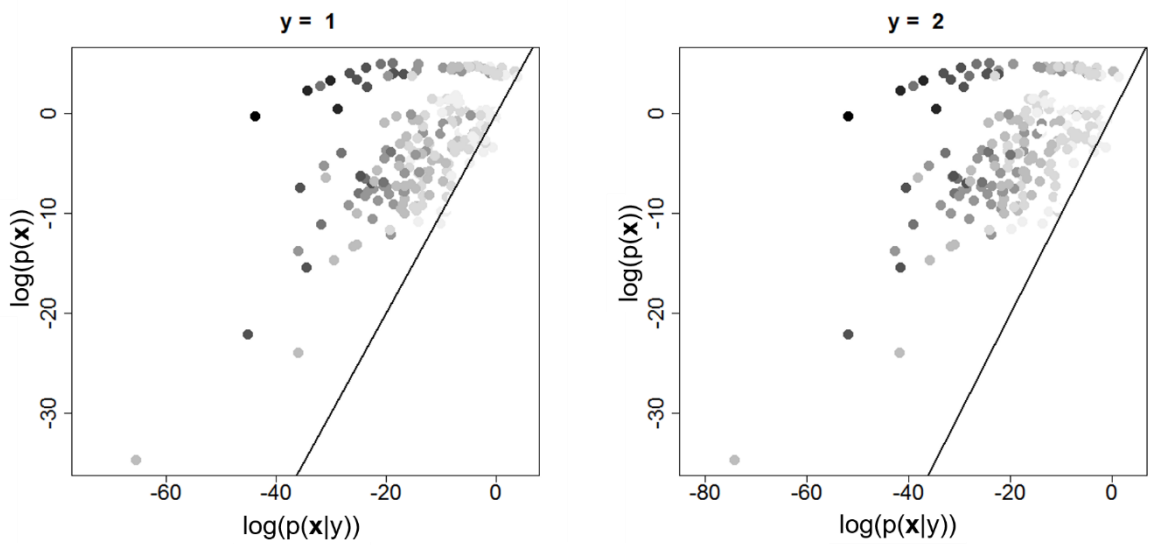
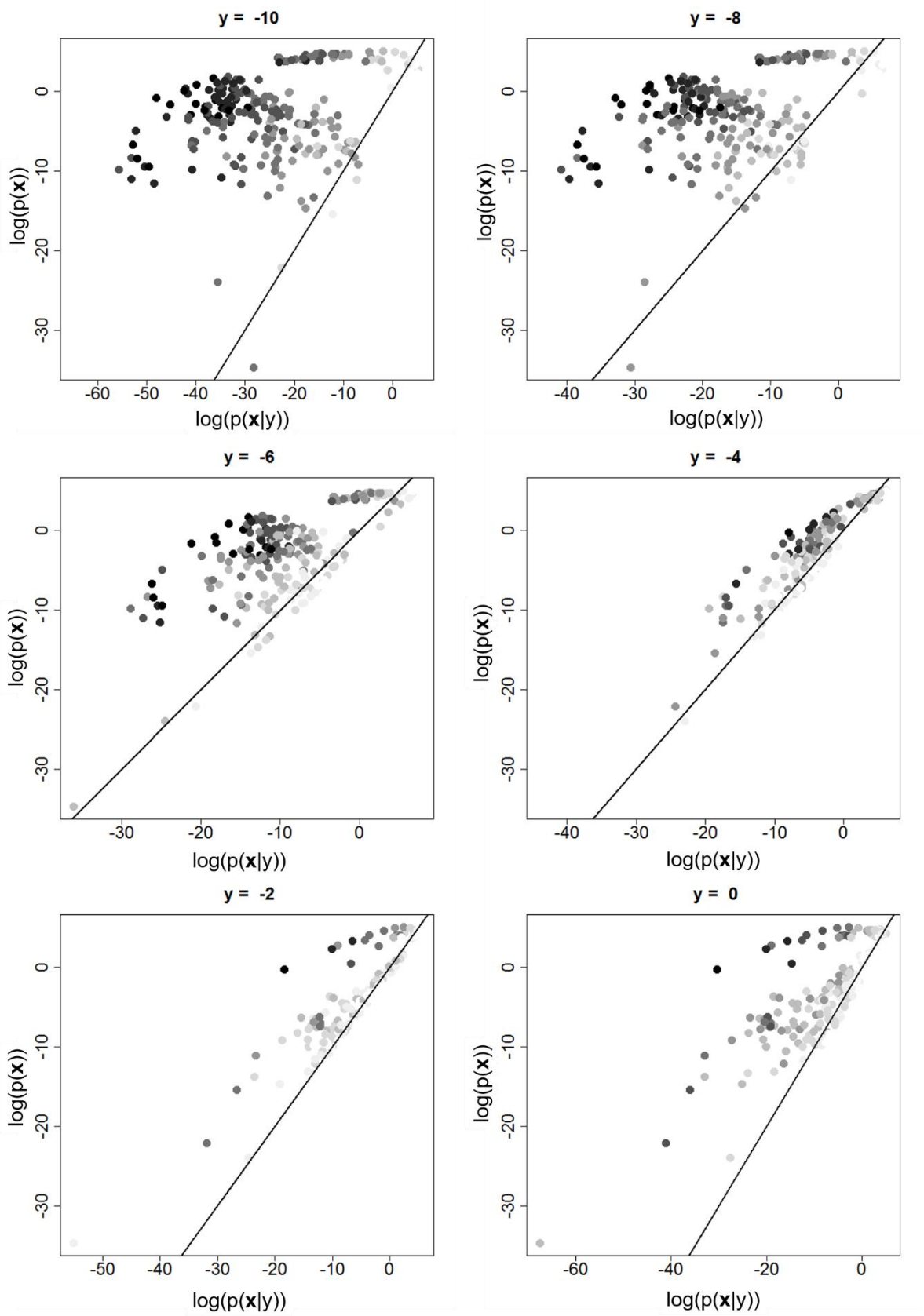


Figure 3-16 Logarithm of p(**x**) is plotted against logarithm of p(**x**|y) with various y values for test dataset by GMMs/cMLR. Color scale represents the absolute error between the y value set in inverse analysis and the measured one. The thinner the color becomes, the less error the dot exhibits.


y = -10
y = -8
y = -6
y = -4
y = -2
y = 0
log(p(x))
log(p(x|y))

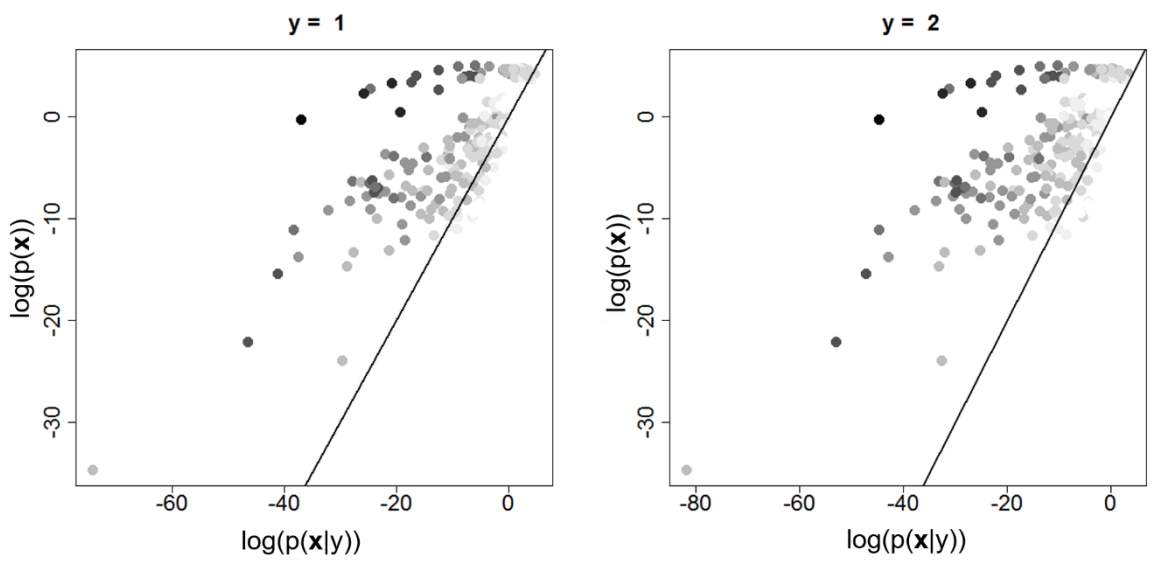
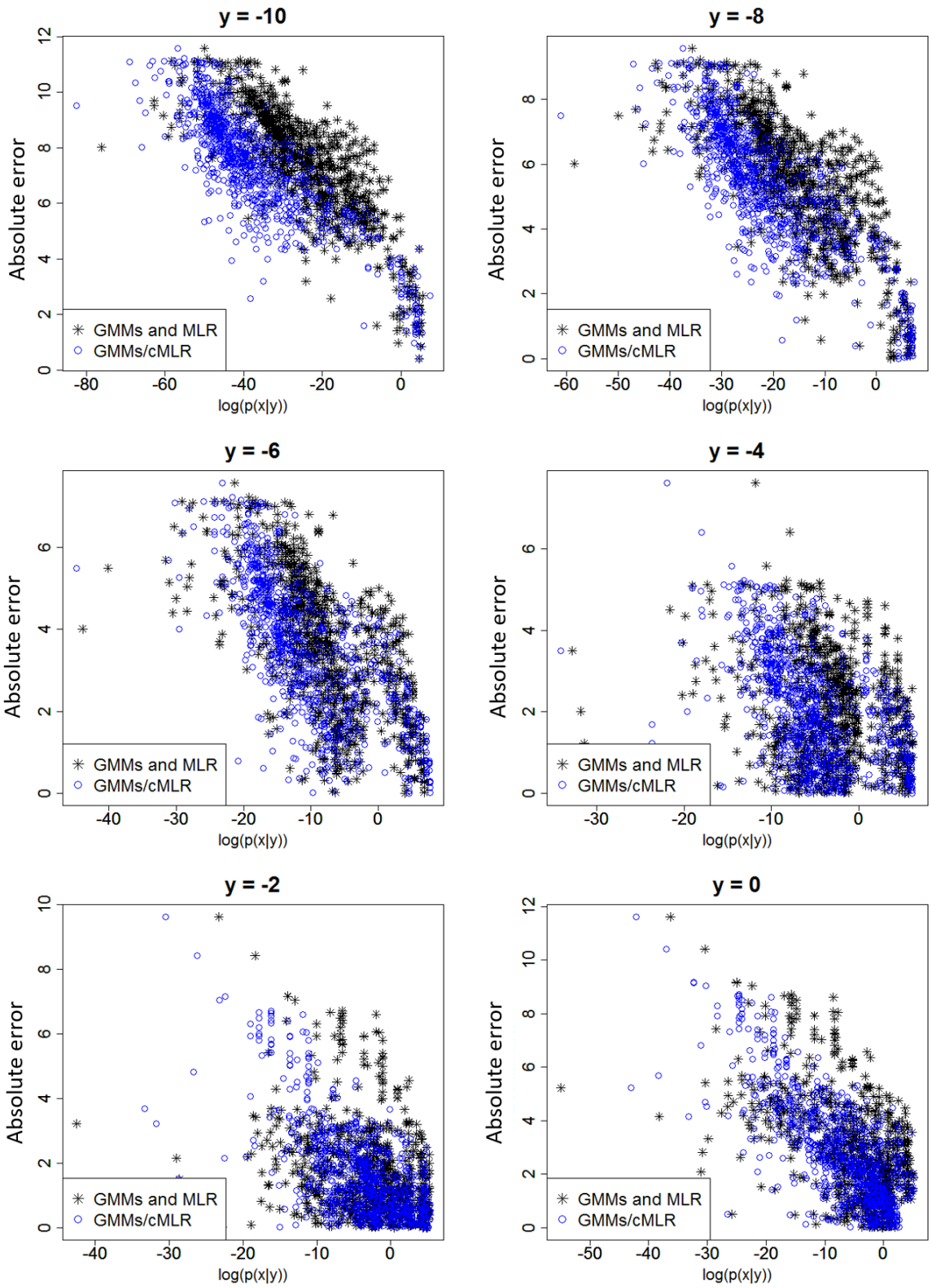


Figure 3-17 Logarithm of p(**x**) is plotted against logarithm of p(**x**|y) with various y values for test dataset by GMMs and MLR. Color scale represents the absolute error between the y value set in inverse analysis and the measured one. The thinner the color becomes, the less error the dot exhibits. This figure corresponds to **Figure 3-16**.

In order to emphasize that p(**x**|y) is able to express the degree of closeness between its target y value and the actual value for novel samples, absolute error (*i.e.* degree of closeness) between them is plotted against log(p(**x**|y)). **Figure 3-18** shows these plots for the training dataset, and **Figure 3-19** for the test dataset. In **Figure 3-19**, GMMs/cMLR expresses a decreasing tread of absolute error as p(**x**|y) increases. On both pictures for y = -4 in training and test datasets, the decreasing tread becomes weak compared to other target y values.


y = -10
y = -8
y = -6
y = -4
y = -2
y = 0
Absolute error
log(p(x|y))
GMMs and MLR
GMMs/cMLR

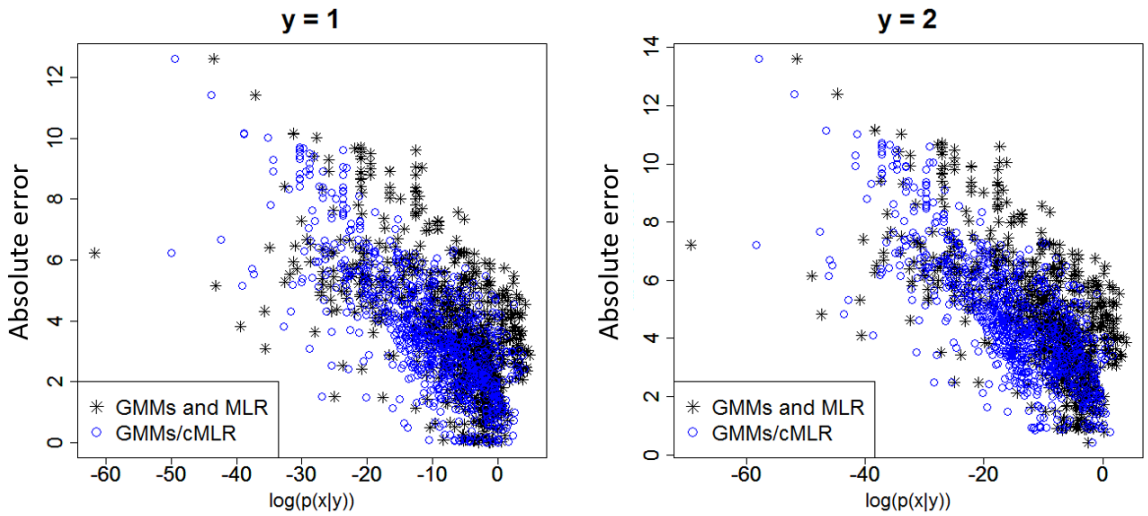
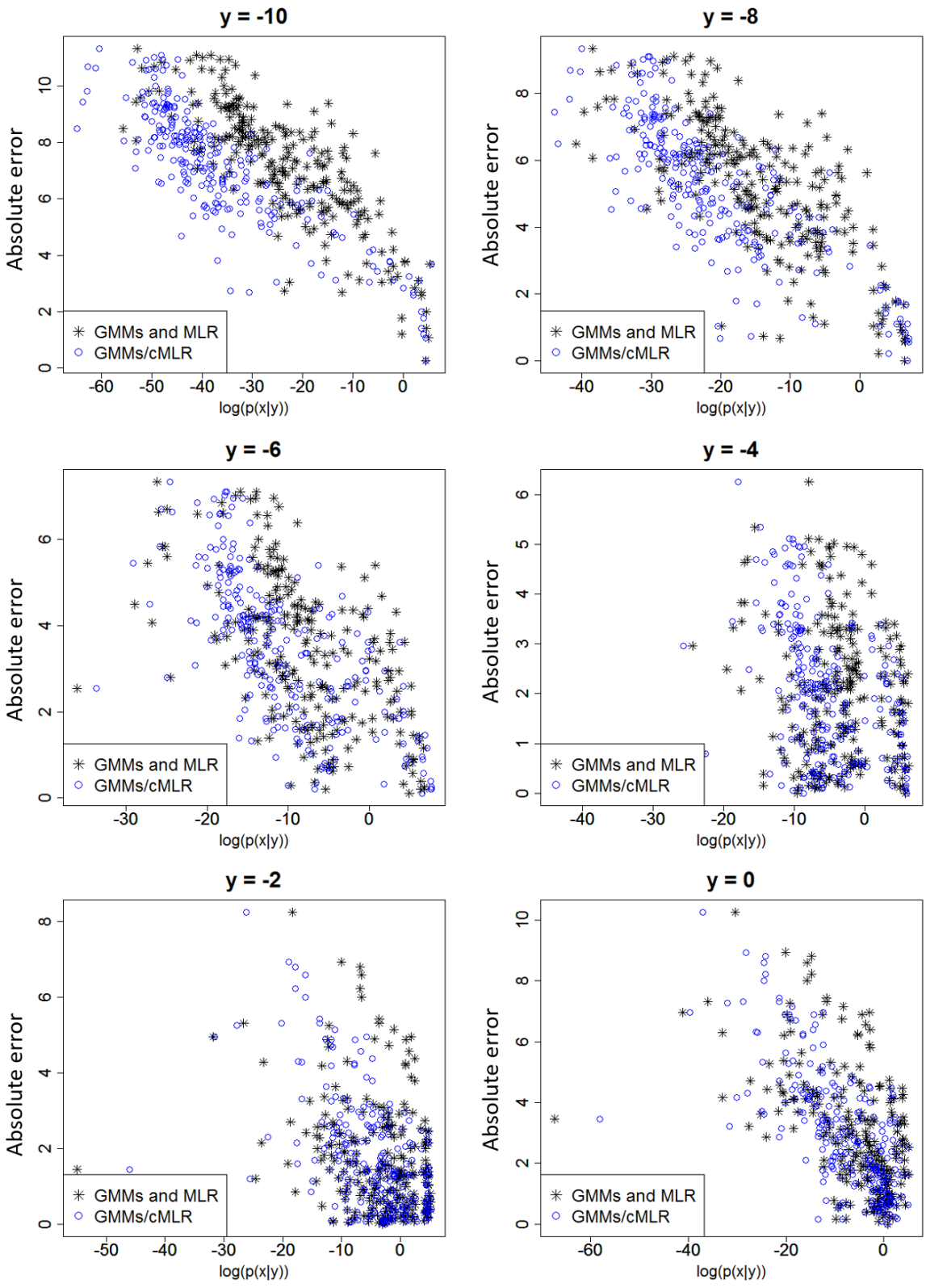


Figure 3-18 Absolute error of the target y value and measured one against p(**x**|y) for the training dataset. Black *s are with GMMs and MLR, blue circles are with GMMs/cMLR.


y = -10
y = -8
y = -6
y = -4
y = -2
y = 0
Absolute error
log(p(x|y))
GMMs and MLR
GMMs/cMLR

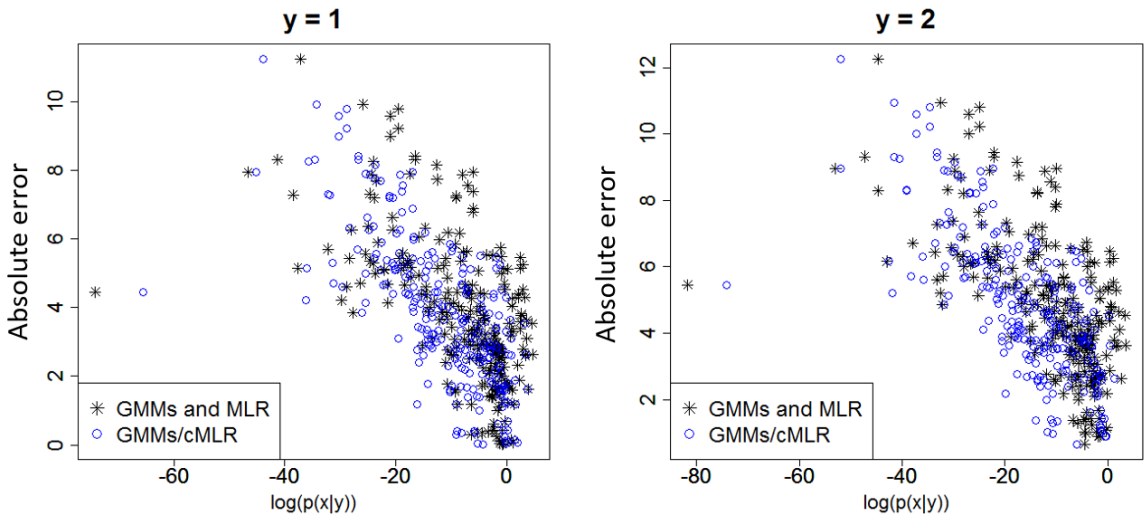
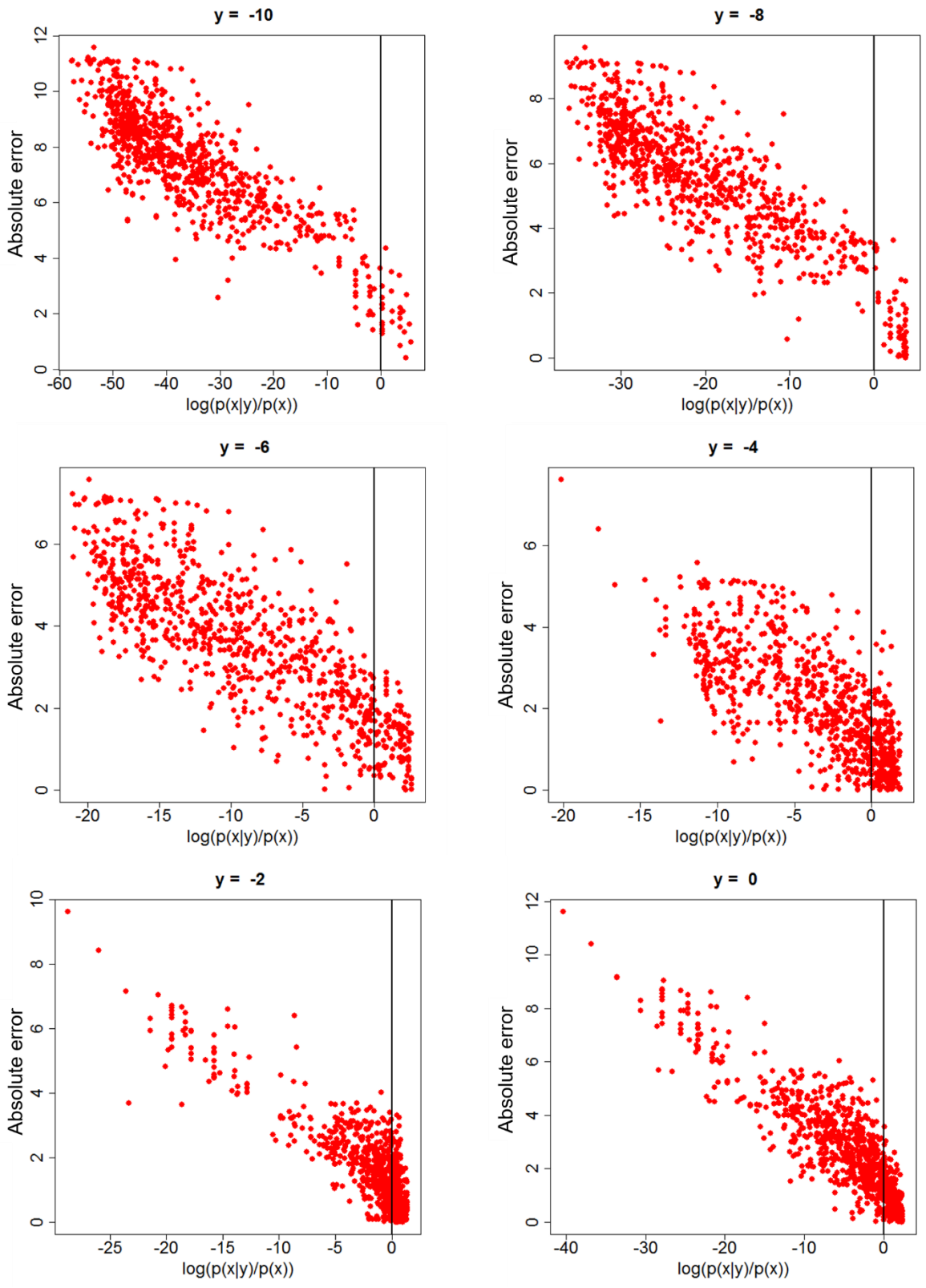


Figure 3-19 Absolute error of the target y value and measured one against p(**x**|y) for the test dataset. Black *s are with GMMs and MLR, blue circles are with GMMs/cMLR.

These decreasing treads explained above can be emphasized by calculating log(p(**x**|y)) – log(p(**x**)) in **Figure 3-20** and **Figure 3-21** only for GMMs/cMLR. From these two figures, p(**x**|y) contains the degree of closeness, since p(**x**|y)/p(**x**) is negatively correlated with the distance to the target y value. Unfortunately, as explained above, many samples wrongly gained information in the posterior distribution for y = -4. However, overall tendency in these pictures reveals that posterior distributions contain information about the degree of closeness to the target y value.


y = -10
Absolute error
log(p(x|y)/p(x))
y = -8
Absolute error
log(p(x|y)/p(x))
y = -6
Absolute error
log(p(x|y)/p(x))
y = -4
Absolute error
log(p(x|y)/p(x))
y = -2
Absolute error
log(p(x|y)/p(x))
y = 0
Absolute error
log(p(x|y)/p(x))

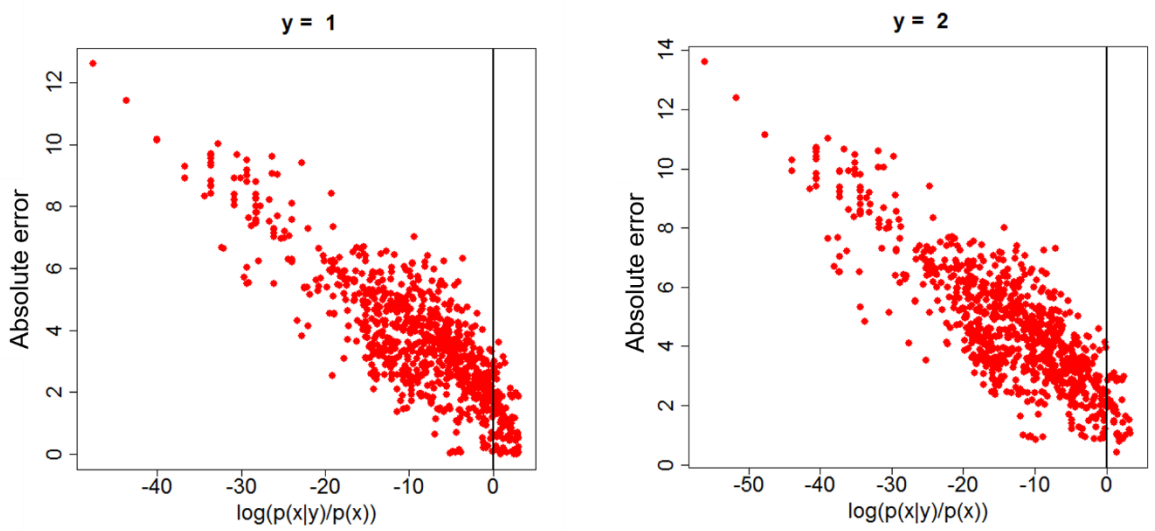
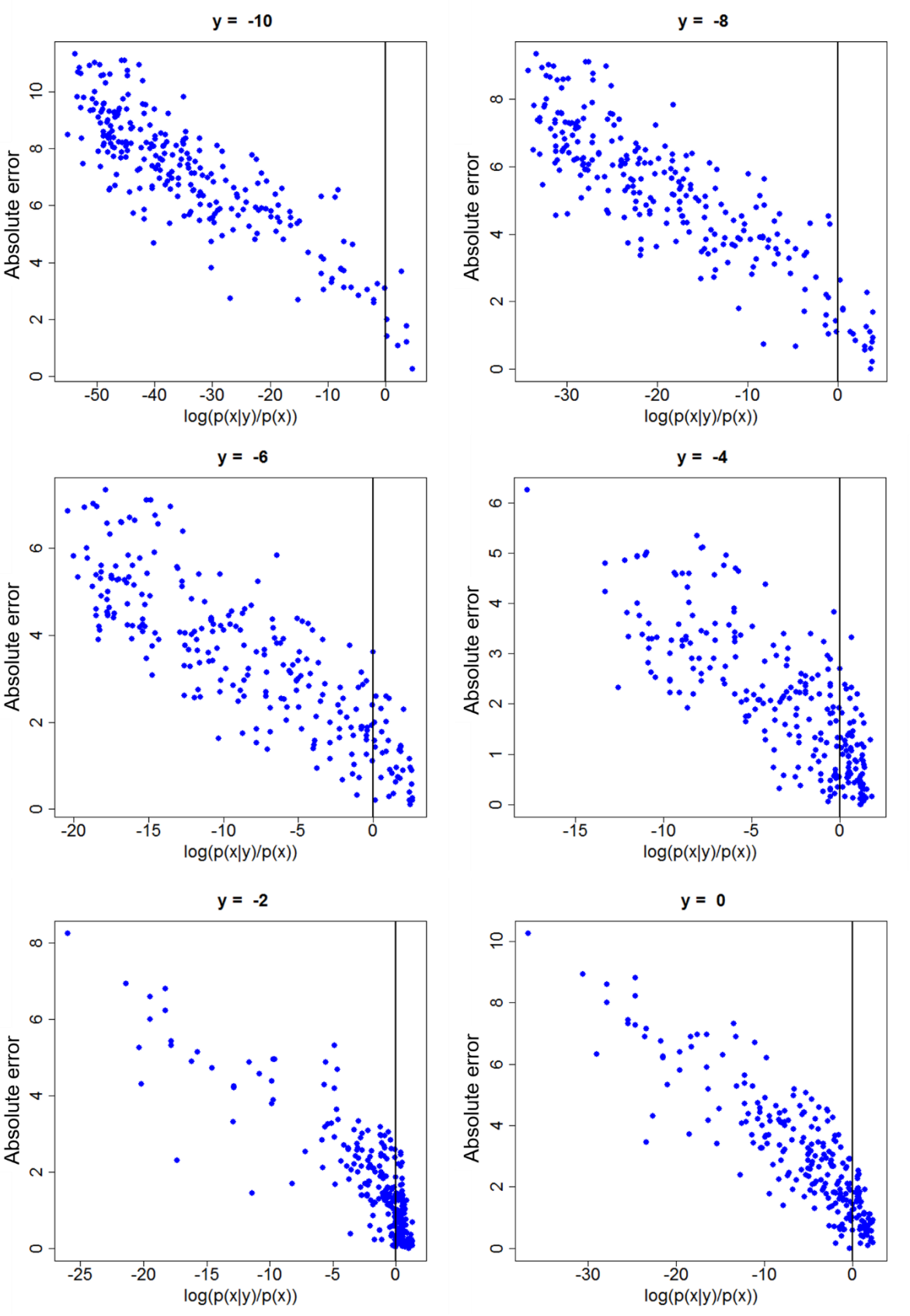


Figure 3-20 Absolute error of the target y value and measured one against p(**x**|y)/p(**x**) for the training dataset by GMMs/cMLR.


y = -10
Absolute error
log(p(x|y)/p(x))
y = -8
Absolute error
log(p(x|y)/p(x))
y = -6
Absolute error
log(p(x|y)/p(x))
y = -4
Absolute error
log(p(x|y)/p(x))
y = -2
Absolute error
log(p(x|y)/p(x))
y = 0
Absolute error
log(p(x|y)/p(x))

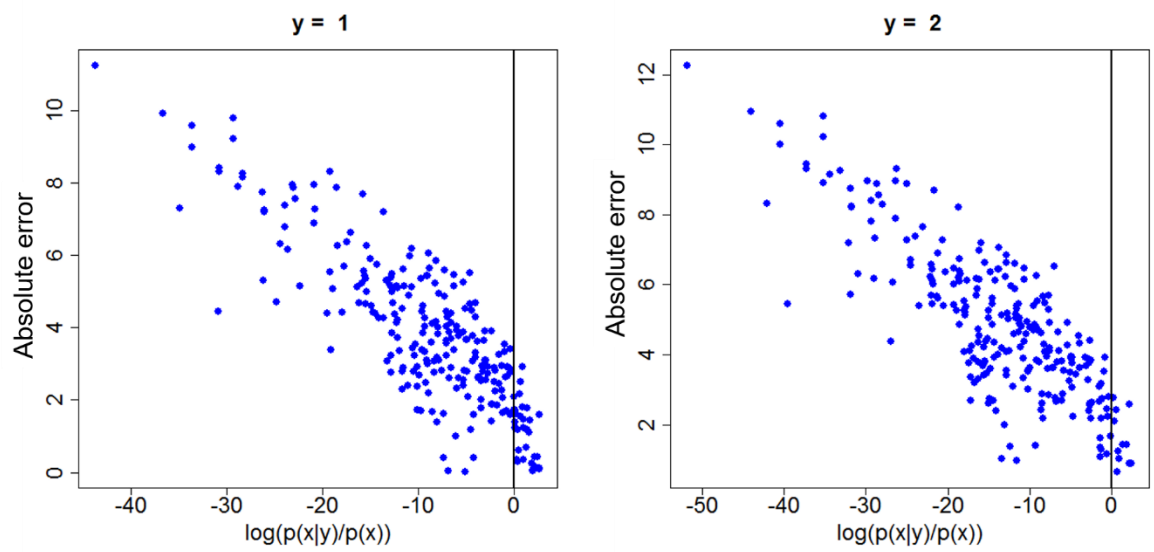


Figure 3-21 Absolute error of the target y value and measured one against p(**x**|y)/p(**x**) for the test dataset by GMMs/cMLR.

Discussions above reveal that p(**x**|y) derived from p(**x**) and p(y|**x**) using the proposed methodology (GMMs/cMLR) can have the preferable properties: inheriting distribution feature of p(**x**) and expressing closeness to the target y value. It is, however, not straight forward to determine the proper threshold of p(**x**|y) since the scale of p(**x**|y) varies drastically based on y values. GMMs/cMLR seems to be superior to GMMs and MLR judging from these pictures and predictability as a QSPR/QSAR model.

3-6 Conclusion

In this chapter, retrieving **x** information from y was explained and also discussed. GMMs/cMLR is proposed to capture the nonlinear relationship between **x** and y. It can derive p(**x**|y) as a GMM in a closed-form solution. The methodology was explained in detail and its difference from the previously proposed one (GMMs and MLR) was highlighted. Three cases studies: the adrenoceptor dataset, the simulation dataset, and the aqueous solubility dataset, were carried out for understanding traits in the proposed methodology From the first case study, predictability of GMMs/cMLR is found to be superior to that of MLR. Furthermore, it can be interpreted the same way as other linear regression models can. In GMMs/cMLR, regression coefficients differ from cluster to cluster, therefore, it sometimes does not give a consistent interpretation unlike a normal MLR. Second case study emphasized on the posterior distribution differences between GMMs/cMLR and "GMMs and MLR". The distributions for both methodologies were different from each other as shown in p(**x**|y) contour plots. As the simulation dataset had a nonlinear relationship between **x** and y, predictability of GMMs/cMLR was higher than MLR. R_{pred}^2 with GMMs/cMLR was 0.927 whereas that with GMMs and MLR was 0.293. p(**x**|y) by GMMs/cMLR, accordingly, was shown to be suitable for this type of datasets. Finally, regarding the aqueous solubility dataset, p(**x**|y) containing AD information as well as the degree of closeness to the target y was

discussed. Both types of information can be seen in p(**x**|y) for both methodologies, GMMs/cMLR and GMMs and MLR. GMMs/cMLR shows a better profile than "GMMs and MLR" does overall.

Although p(**x**|y) by GMMs/cMLR contains AD information and closeness to the target y information, it is still an unsolved problem how to determine a proper threshold for p(**x**|y).

Despite the fact that there is still room for improvement regarding p(**x**|y), one that compromises both data density and the closeness to the target value (it is the same problem using p(**x**)), it is fair to say that p(**x**|y) can potentially be used for AD focusing on y

CHAPTER 4 Structure Generation System Based on Inverse QSPR/QSAR

4-1 Introduction

In the previous two chapters, the elements of inverse QSPR/QSAR system were separately described. In CHAPTER 2, structure generation strategy for retrieving chemical structures satisfying constraints was explained. As long as the constraints are defined as a set of MCD values, it is possible to retrieve chemical graphs from the set of values using the proposed methodology. Structure generator *FragmentGenerator* in *Molgilla* has been developed by implementing the proposed generation algorithms. In CHAPTER 3, a methodology for acquiring **x** information from y was proposed. It proposes to retrieve **x** information as the posterior PDF of **x** given y. Furthermore, nonlinear relationship between **x** and y can be represented by the methodology (*i.e.* cluster-wise MLR). In order to reach the goal mentioned in this thesis, which is to propose and develop a chemical structure generation system based on inverse QSPR/QSAR analysis, those two strategies can be simply connected sequentially (Figure 4-1).

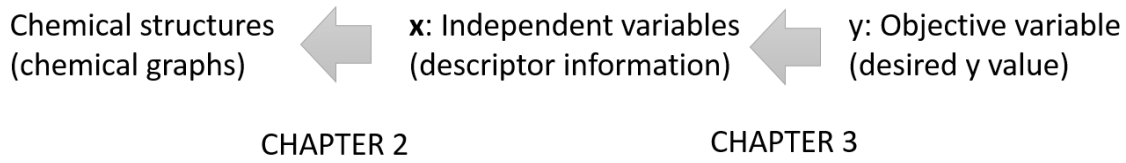


Figure 4-1 Relation between the goal of this thesis and CHAPTER 2 and 3.

Figure 4-1 is a simplified representation of the proposed workflow. In order to connect the two strategies, several challenges should be overcome. One of the unsolved matters is how to determine **x** coordinates from a posterior PDF. Input for the proposed structure generator is the constraints defined as upper and lower bounds of MCDs. The output of inverse analysis is, on the other hand, p(**x**|y), where y is a desired property or activity value. Therefore, a methodology to retrieve **x** coordinates from p(**x**|y) should be developed.

In this chapter, details of the proposed system will be explained, followed by a practical inverse QSAR application. When applying the proposed system to actual chemical structure design, a number of challenges emerged. In the following sections about the application, how to overcome these challenges is emphasized. The author emphasizes that the methodology for structure design based on inverse QSAR is trial and error rather than operating the system as it is. Hence, tuning up the proposed methodology to an application object is necessary.

4-2 Proposed System for Chemical Structure Generation

An overview of the proposed chemical structure generation workflow is illustrated on **Figure 4-2**. Basically, ring systems are extracted from the molecules in the dataset for constructing a QSPR/QSAR model. In addition to the ring systems, atom fragments are also employed as elements for structure generation in order to make diversified molecules (2 in **Figure 4-2**). MCDs are used for constructing both a QSPR/QSAR model with cMLR and prior density with a GMM. In inverse analysis, an input is a y value. Then posterior PDF of **x** given the y value ($p(\mathbf{x}|y)$) is derived as a closed-form solution by combining $p(\mathbf{x})$ and $p(y|\mathbf{x})$. The **x** coordinates in descriptor space are determined based on $p(\mathbf{x}|y)$, followed by the transformation of **x** to constraints (1 in **Figure 4-2**). To apply the proposed generation algorithm explained in the section 2-6-2 , constraints are set as the upper and lower bounds of MCD values. Determining the upper and lower bounds equals to identification of a range. Since the range of each MCD is determined based on $p(\mathbf{x}|y)$, Miyao *et. al*. proposed to make use of one-dimensional marginal distribution of the variable[100]. The marginal distribution is obtained by summing up the other variables (integral of these variables). Although they succeeded in designing acyclic hydrocarbons that exhibit the boiling point they specified, using one-dimensional marginal distribution means ignoring a correlation between the variable and the rest of variables as a nature of Gaussian distributions. Ignoring correlation among variables increases the probability of finding a wrong region because GMM is a multimodal distribution. Another problem about determination of a range based on one-dimensional marginal distribution is that the size of the rectangle consisting of the ranges tends to be large. Specifically, the procedure of determining each variable's range is based on probability. Only the continuous region having the probability exceeding a certain threshold (*e.g.* 90%) is selected. The more descriptors are employed, the more volume the hyper-rectangle contains. When the range for one descriptor is broad, the number of possible combinations for discrete descriptors or the volume of hyper-rectangle increases. Consequently, the number of corresponding structures would be beyond our handling.

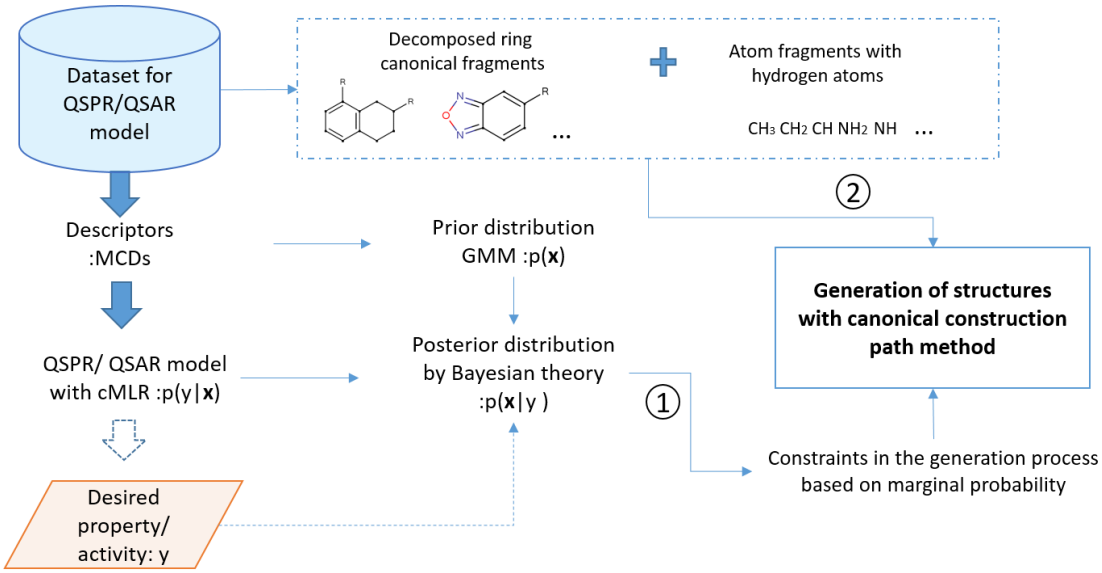


Figure 4-2 Overview of the proposed chemical structure generation system based on inverse QSPR/QSAR. This figure is modified from Figure 1 in the paper of Miyao *et al.*[102]

Unlike using one-dimensional marginal distributions for determining constraints, here the author proposes to focus on the center of one Gaussian. Although both methodologies result in determination of a hyper-rectangle in descriptor space, the rectangle sizes differ greatly. Focusing only on one Gaussian means the ranges can be determined as narrowly as possible. In the extreme, the size of hyper-rectangle being zero is acceptable, because it means the constraints is the center of a Gaussian.

It should be noted that, even inside a narrowly determined hyper-rectangle, densities of a Gaussian distribution vary tremendously. This is due to the curse of dimensionality, or simply due to the increment of the Euclidean distance between the center of a Gaussian and surrounding points[143] as the dimension goes higher. Assuming there is a two-dimensional normal distribution whose mean is zero and covariance is an identity matrix, densities of the center and the vertex at (1,1) are 0.159 and 0.0585, respectively. In contrast to the low dimensional density feature, density in high-dimension is counterintuitive. With a 30-dimensional normal distribution whose mean is zero and covariance is an identity matrix, as the same way as in the two-dimensional example, densities of the center and the one vertex on the corresponding hyper-rectangle are 1.06×10^{-12} and 3.26×10^{-19}. The density at one corner of the hyper-rectangle is 3×10^{-7} times smaller than that of the center of the Gaussian.

Although we cannot overcome the curse of dimensionality simply by tightly restricting the area surrounding the center of one Gaussian, at the current moment, this is the best effort that the author can do to repel its effects.

4-3 Design of Thrombin Inhibitors

As a practical application for chemical structure generation based on inverse QSAR analysis, ligand design for thrombin inhibitors was carried out. The goal of this case study is to propose novel ligands that might be candidates as lead structures having high affinity to thrombin. Through this case study, various points that should be taken care of are described, as well as practical comments on the improvement of the proposed system.

4-3-1 Dataset

From the ChEMBL 20 database, conditions used for extracting records regarding thrombin are as follows:

- Target: Thrombin (ChEMBL ID CHEMBL204)
- Organism: *Homo sapiens*
- Target confidence: more than 7
- Bioactivity type K_i
- Operator =
- Assay type B
- Unit: nM

K_i values were transformed into pK_i. If a compound (or a set of compounds in Salts) has multiple records, having multiple pK_i values, the average value is employed as its affinity unless the standard deviation exceeds 0.5. The records over the threshold were discarded. The remaining 3,259 molecules were preprocessed by Standardizer ver.6.1.0 from *ChemAxon*[144] with options as follows: Dearomatize, Neutralize, Remove Explicit Hydrogens, and Strip Salts. Then, MCDs were calculated for the molecules by the *DescriptorCalculator* module in *Molgilla*. From the surviving molecules, only those having less than 10 amide bonds and molecular weight less than 1,000 were selected. The number of samples was 1,705 for this analysis. The reason for eliminating peptides is that it may bind to thrombin in different ways than small molecules do (*i.e.* direct thrombin inhibitors). Furthermore, contaminating a few molecules that take much bigger descriptor values than the rest of the molecules in the dataset do distorts correlation coefficients between descriptors towards high values. From the remaining 1,705 samples, 1,000 samples were randomly selected as training data. The rest of 705 samples were used as test data. Histograms of pK_i values in the both training and test datasets are shown in **Figure 4-3**. Ranges were (2.54, 12.2) of the training dataset and (1.00, 10.89) of the test dataset.

Reported experimental data always contains experimental errors in it. In order to check the accuracy of the reported values in the dataset, K_i values of randomly picked 10 samples in the training dataset were examined. The results of manual inspection are shown on **Table 4-1**. Seven records out of the 10 records did not mention experimental errors in the original papers. Only three of them have been reported the errors ranging from around 8% to 30% of the reported K_i values [nM]. This implies the limitation of improving accuracy for QSAR models aiming at predicting K_i values. There is no sense in saying that we can construct a QSAR model having a predicted error below the value of experimental error. Assuming every reported sample contains 30% error in K_i value, the corresponding pK_i would contain (-0.15-0.11) ($\log_{10}(0.7)$-$\log_{10}(1.3)$) as possible experimental error. Although this simple manual

sampling only covers 1 % of the total compounds in the training dataset (10 compounds out of 1000 compounds), this manual inspection made us to understand features of the data that we were about to treat.

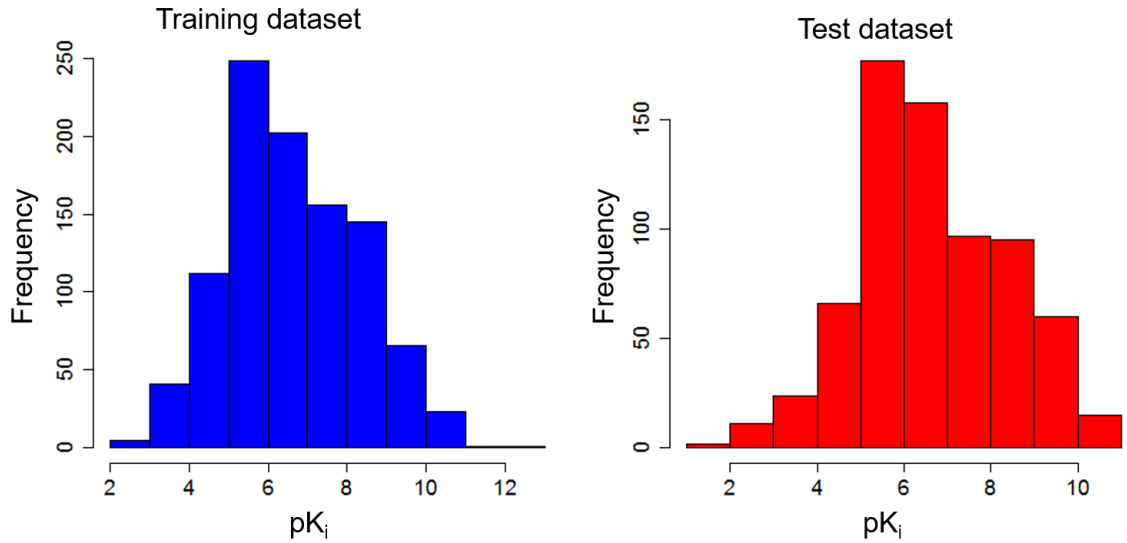


Figure 4-3 Histograms of the pK_i values in the both training and test dataset.

Table 4-1 Reported Ki values and experimental errors

ChEMBL ID	K_i [nM]	Error	Trials*[a]
176000[145]	0.12	0.01 (standard error)	6
287084[146]	29	Not reported	
200153[147]	1	Not reported	
166530[148]	830	Not reported	
317059[149]	560	Not reported	
32612[146]	22	Less than 10%	At least 2
44249[150]	340	Not reported	
50933[151]	260	Not reported	
428982[152]	130	Not reported	2
376791[153]	300	Less than 30%	At least 2

*a: The number of trials for estimating error of K_i

4-3-2 Descriptors

DescriptorCalculator calculated 51 descriptors. In order to construct stable MLR models, variable selection was conducted as follows. First, the variables for which over 90% of training samples exhibit the same value were eliminated. Then, one of the pair variables having correlation coefficient over 0.9 was eliminated. This procedure was repeated until every pair of surviving descriptors has the correlation coefficient less than or equal to 0.9. Collinearity is one of the biggest problems in MLR as described in 3-2-2 In order to construct a ridged MLR model, backward stepwise regression was carried out, and variables were selected based on AIC[131] by the package *MASS*[154] in R. From the MLR model using 34 variables (all variables remaining), it reduces one variable at a time and selects the model having the minimum AIC value. This procedure is repeated until the value does not decrease. Variable selection resulted in 27 variables shown on **Table 4-2**.

Table 4-2 Selected variables based on AIC for model construction. Definition of variables is described on TableE-1 in Appendix E.

CIC	R05	aR	ZM1V	nBM	nHAccLipin	nCH_2R_2	$nCHR_3$	nCH_3R
nCH_3X	nOH	=O	$nArNR_2$	nArCO	TPSA	LL	LD	LP
AA	AP	AN	DD	RL	RA	RD	RP	RR

4-3-3 Model Construction

With the selected 27 descriptors (**Table 4-2**), affinity prediction models for thrombin were constructed by both MLR and GMMs/cMLR. As GMMs, the covariance parameter in *mclust* was “VEV”, which means variable volume, variable orientation, and equal shape is assumed among covariance matrices. The optimal number of Gaussians was five based on BIC. The corresponding value was -31,551. The number of training samples and test samples in each cluster were shown on **Table 4-3**. Predictability and yy-plots by the two regression methodologies are shown on

Table 4-4 and in **Figure 4-4**, respectively. F-statistic of the MLR model was 29.13, and of C1 in GMMs/cMLR was 12.99, C2 8.766, C3 8.608, C4 17.09, and C5 16.48, meaning, all the models including sub-models in GMMs/cMLR are significant at the level of p-value < 2.2×10^{-16}.

Table 4-3 Number of training data and test data categorized in one of 5 clusters.

	C1	C2	C3	C4	C5
Training	147	167	137	343	206
Test	92	103	112	261	137

Table 4-4 Results of model construction by GMMs/MLR and MLR methodology

	R^2	RMSE	R_{pred}^2	$RMSE_{pred}$
MLR	0.448	1.251	0.354	1.372
GMMs/cMLR	0.667	0.972	0.425	1.294

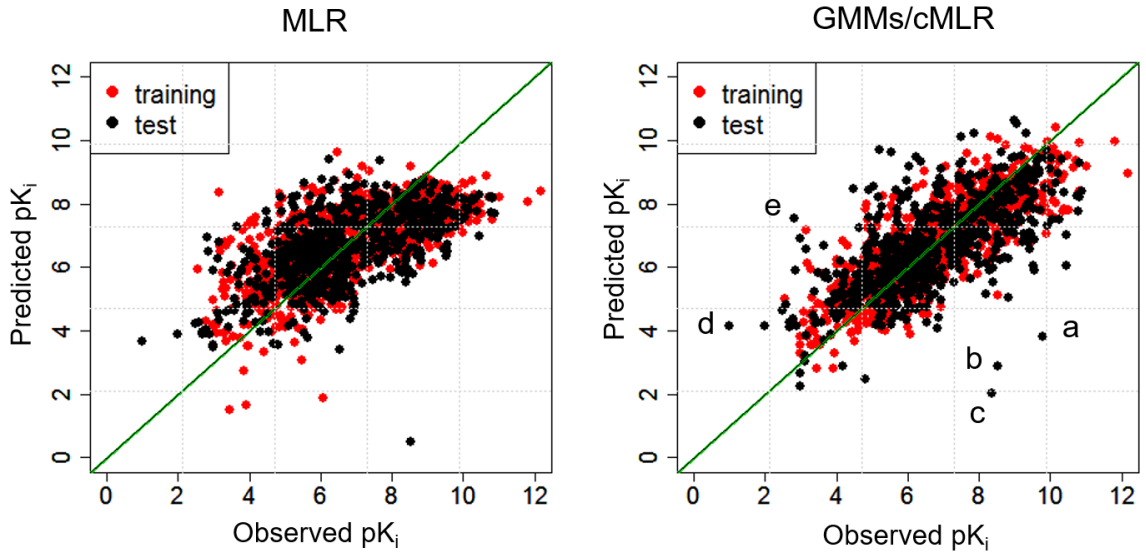


Figure 4-4 Predicted pK_i value against observed value for thrombin dataset. For GMMs/cMLR, five outliers are marked as a, b, c, d and e.

There are five outliers by GMMs/cMLR as pointed out in **Figure 4-4**. These are all included in the test dataset. Structures as well as their density by the constructed GMM are shown in **Figure 4-5**. **Figure 4-6** shows the histogram of the logarithm of p(**x**) values. These two figures imply that the three outliers: a, b, and c based on the y-y plot are out of AD, meaning the predicted values for them cannot be trusted. As supplement information, ChEMBL 539697 (b) is dabigatran etexilate, which is a FDA approved drug in the class of the direct thrombin inhibitors. It is a prodrug of dabigatran, and rapidly converted into a dabigatran by metabolism[155]. ChEMBL 539697 is wrongly annotated with the affinity of dabigatran instead of that of dabigatran etexilate. Two other outliers having lower p(**x**): (a) and (c) are not surrounded by samples in the training dataset. The constructed model underestimated the predicted values for both outliers. ChEMBL197593 (a) exhibits 9.795 as pK_i (0.16 nM as K_i). The structure does not contain NH2 atom fragments, but do contain a carboxylic group. With the current rules for recognition of PPPs, only carboxylic groups are recognized as negatively charged points, and NH2 atom fragments are as positively charged points. It is a well-known fact that the Asp189 in the S1 pocket of thrombin is negatively charged and many ligands as direct thrombin inhibitors have positively charged points for making an interaction for that part. There are only two samples in the training data, samples which have a carboxylic group and no amines. The criteria for these two substructures were [CX3](=O)[OH1] and [NX3H2] in a SMILES arbitrary target specification (SMARTS) format, respectively. The two structures are ChEMBL 3134468 (pK_i 4.75) and ChEMBL 2391037 (pK_i 4.78). Since the training dataset does not contain molecules similar to a, the predicted value for a is not reliable based on the QSAR model with the current descriptor set.

This also implies the limitation of using QSAR models. ChEMBL 197593 was actually proposed as a result of the discovery of ligands without amines as their substructures. Deng et al. found that 5-chloro-2-methylaniline occupies the S1 pocket in thrombin[156], meaning that a does interact with thrombin in a different way from the previously well-known ionic interaction. Although the constructed QSAR model cannot make accurate prediction, the model can insist that a is out of AD. On the other hands, molecules d (ChEMBL 3247182) and e (ChEMBL 122726) have high p(**x**), meaning that they are inside AD. d exhibits pK_i of 1.00 and e does pK_i of 2.82. The predicted pK_i values for d and e were 4.15 and 7.52, respectively. Neighbor molecules for each molecule were selected from the training dataset based on the Euclidian distance in the scaled MCD space. For molecule d, three molecules having the benzamidine substructure. Although they are not identical to d in terms of PPPs, the pK_i values for these neighbor compounds are higher than that of actual d, leading to overestimate the pK_i value. For molecule e, there is an analogue found in the training dataset (the bottom row in Figure 4 7). Based on the fact that these two structures exhibit the substantial difference of affinity (for e, pK_i was 2.82 and for the nearest neighbor of e, pK_i was 7.23), there two structures form an activity-cliff[128], leading to the wrong prediction of the proposed QSAR model.

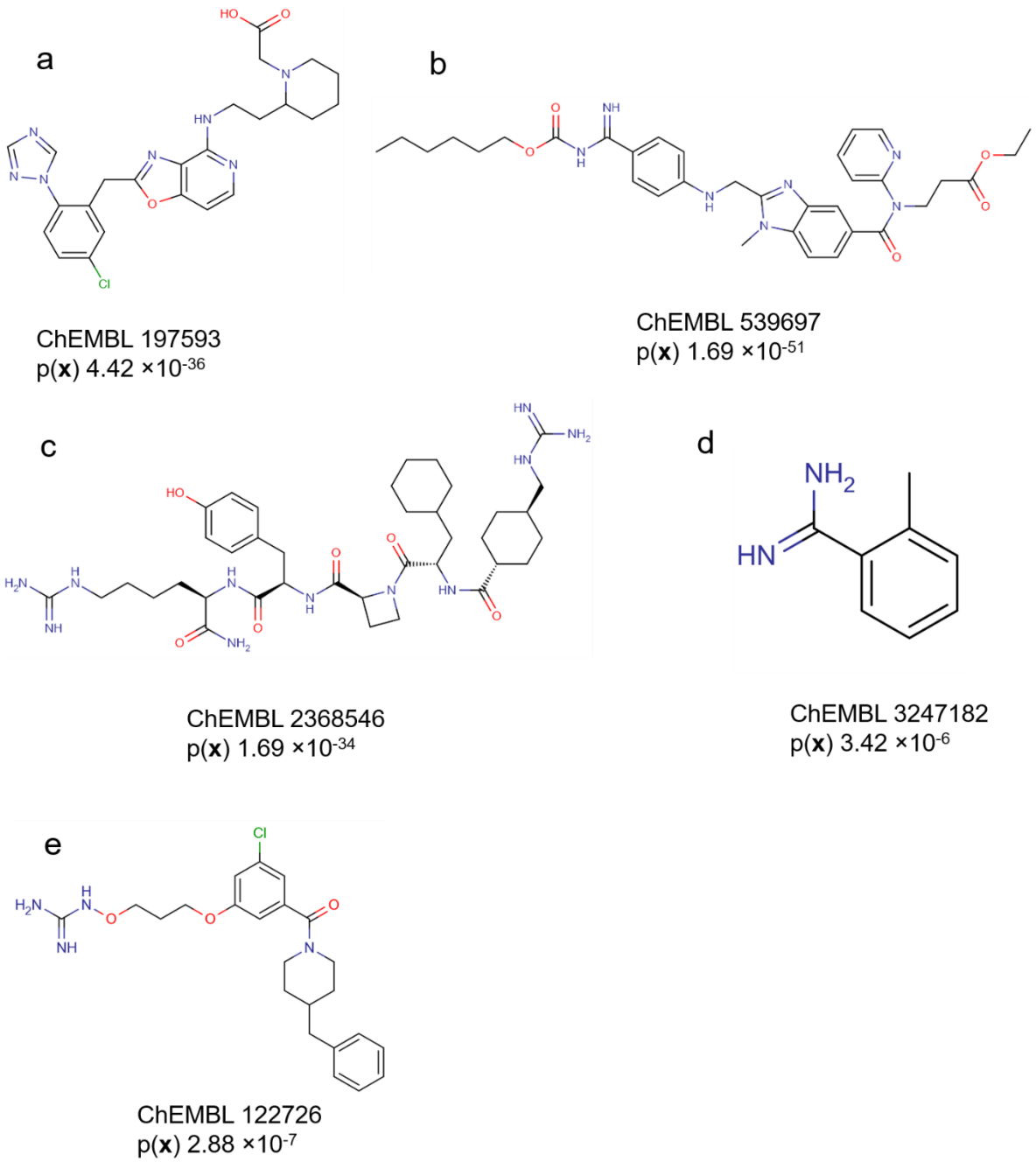


Figure 4-5 Five outliers pointed out based on the yy-plot in **Figure 4-4**.

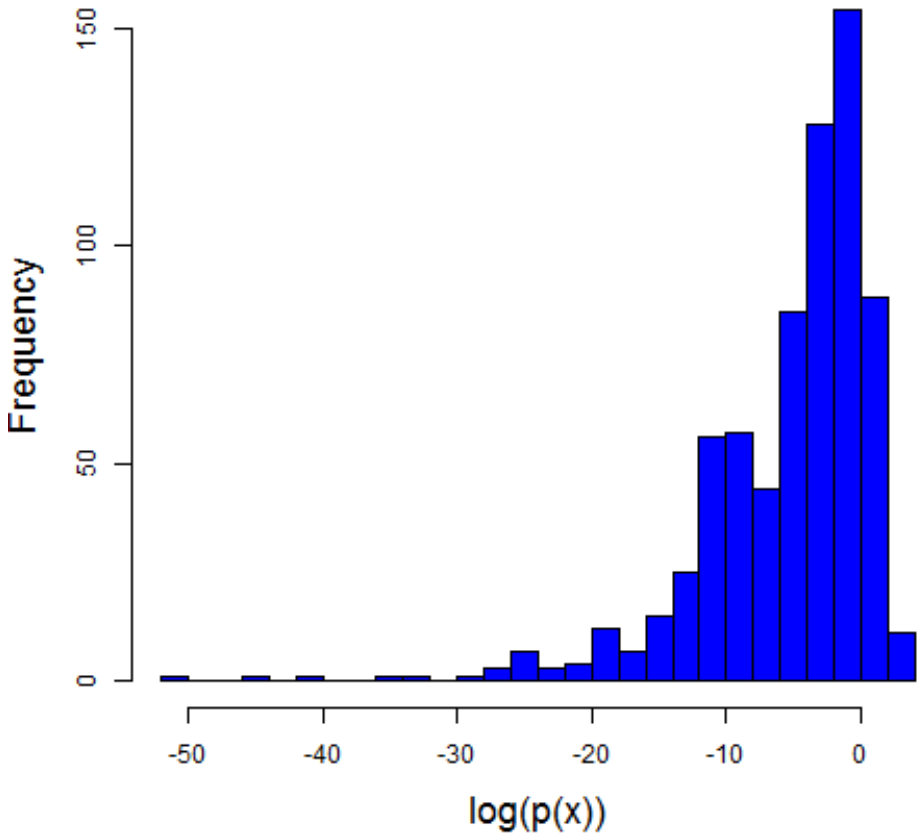


Figure 4-6 Histogram of the logarithm of p(**x**) for the test dataset.

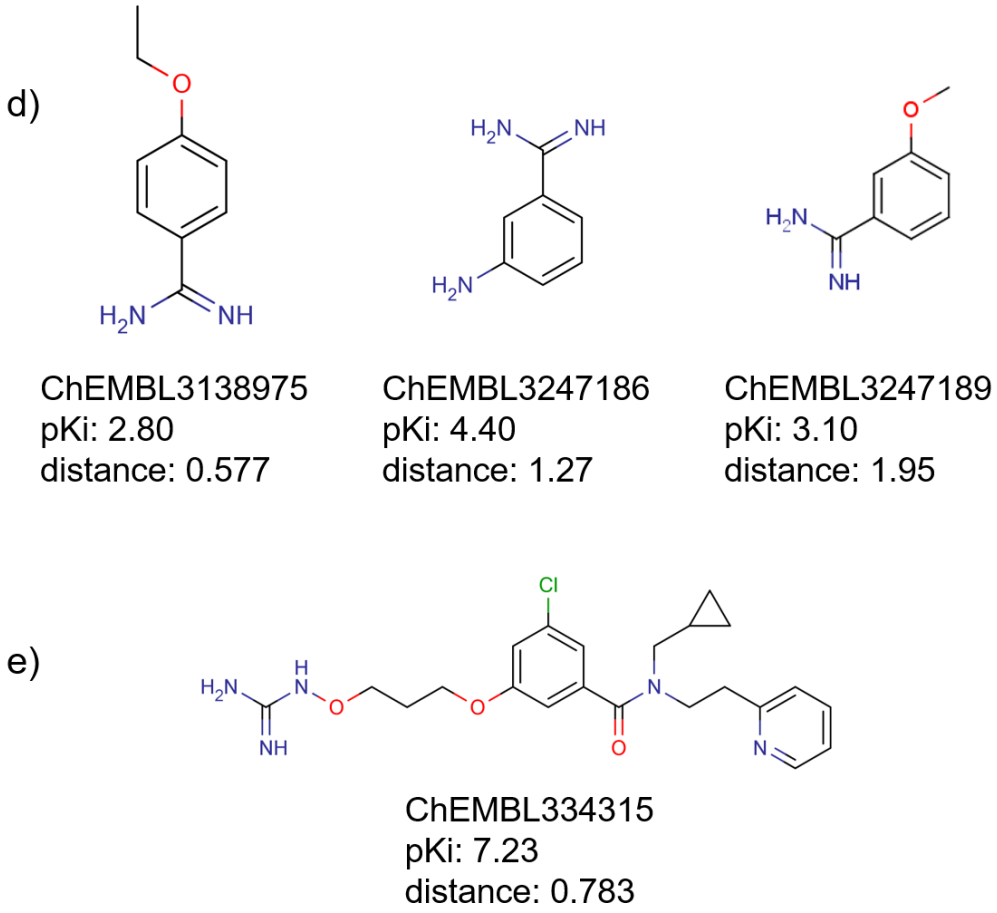


Figure 4-7 Nearest neighbors in the training dataset for outlier d and e in **Figure 4-5**. For compound d, the three nearest neighbors are shown. For compound e, the nearest neighbor is shown.

4-3-4 Inverse Analysis for De Novo Structure Design

As explained in the previous section, the constructed GMMs/cMLR model had predictability at R_{pred}^2 0.425 for test dataset. It might not be sufficient for conducting inverse QSAR analysis. For further analysis, both datasets were combined to form a training dataset. The validity of the workflow of the proposed methodology is independently scrutinized using the external dataset (test dataset). The results of this validation analysis is explained in Appendix G.

In this analysis with all 1,705 compounds, the same descriptors as those determined by AIC-based variable selection using training dataset were employed (**Table 4-2**). As a result of training a GMM, 8 Gaussians having the value of BIC at -42,817 were optimized. The parameter for a covariance matrix in *mclust* was "VEV", which is the same parameter as the one determined by training dataset. The Coordinates of prior Gaussian centers are on Table F-1 in Appendix F.

A MLR model was constructed for each Gaussian. R^2 and RMSE 0.656 and 0.993, respectively. F statistics (p-value) of the sub-models are C1 17.26 ($< 2.2\times10^{-16}$), C2 3.124 (6.1×10^{-6}), C3 3.405 (3.8×10^{-5}), C4 8.263 ($< 2.2\times10^{-16}$), C5 13.3 ($< 2.2\times10^{-16}$), C6 24.11 ($< 2.2\times10^{-16}$), C7 7.62 (1.1×10^{-12}), and C8 12.32 ($< 2.2\times10^{-16}$). Predicted value is plotted against the observed one in **Figure 4-8**.

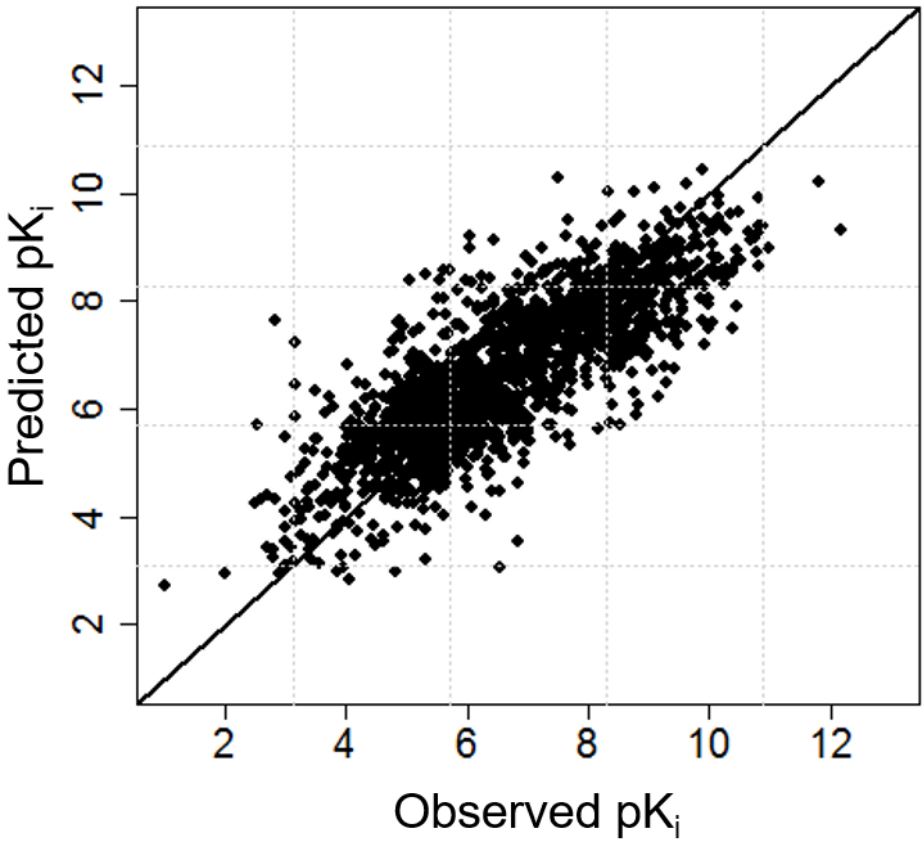


Figure 4-8 Predicted value is plotted against the observed one with the GMMs/MLR model using 1,705 samples.

As inverse analysis, objective variable values are set from 2 to 12 by 1. Consequently, by applying the formula derived in section 3-2-3 , 11 p(**x**|y) distributions could be retrieved. In order to visualize the posterior distributions (*i.e.* p(**x**|y)), which are distributions in high-dimensional space, a two-dimensional map with generative topographic mapping (GTM)[157] was employed. GTM is a nonlinear dimensionality reduction technique, which can convert data distribution in high-dimensional space into ones in lower dimension space. For visualization purposes, the dimension of the lower space is set two. A map by GTM was constructed with the training data using the package of GTM Toolbox[158] developed by Svensen.

Average Euclidean distance in the dataset between a coordinate in the original space (i.e. 27 dimension) and the corresponding projected point on the map was the criterion for

determining hyper-parameters responsible for constructing a map. The smaller the distance is, the better the map represents data distribution. In order to avoid overfitting, 3-fold cross validation was conducted when evaluating the distance. There are four hyper-parameters in GTM: mapsize (Mapsizes), number of radial basis functions (RBFs) (nRBF), the regularization term (Reg), and the variance of the Gaussians in RBFs (wRBF). The searched parameters were nRBF = {49, 64, 81, 100}, Reg = {0.001, 0.01}, and wRBF = {0.125, 0.25, 0.5, 1}. Mapsize was fixed at 400 based on the number of training samples. As a result of optimization, the optimal set of parameters was (nRBF, Reg, wRBF, Mapsizse) = (100, 0.01, 0.5, 400). The corresponding RMSE by cross-validation was 2.386. The final map for visualization was re-trained with all the training data and the determined set of hyper-parameters. **Figure 4-9** shows the map on which all samples are projected annotated with pK_i values. In this study, projection on the GTM map is based on the mean criterion. A sample in the high-dimensional descriptor space is projected on a set of two-dimensional coordinates. The coordinates are the mean of all the grids on the map weighted by the posterior density of the grid given the sample in high-dimensional space.

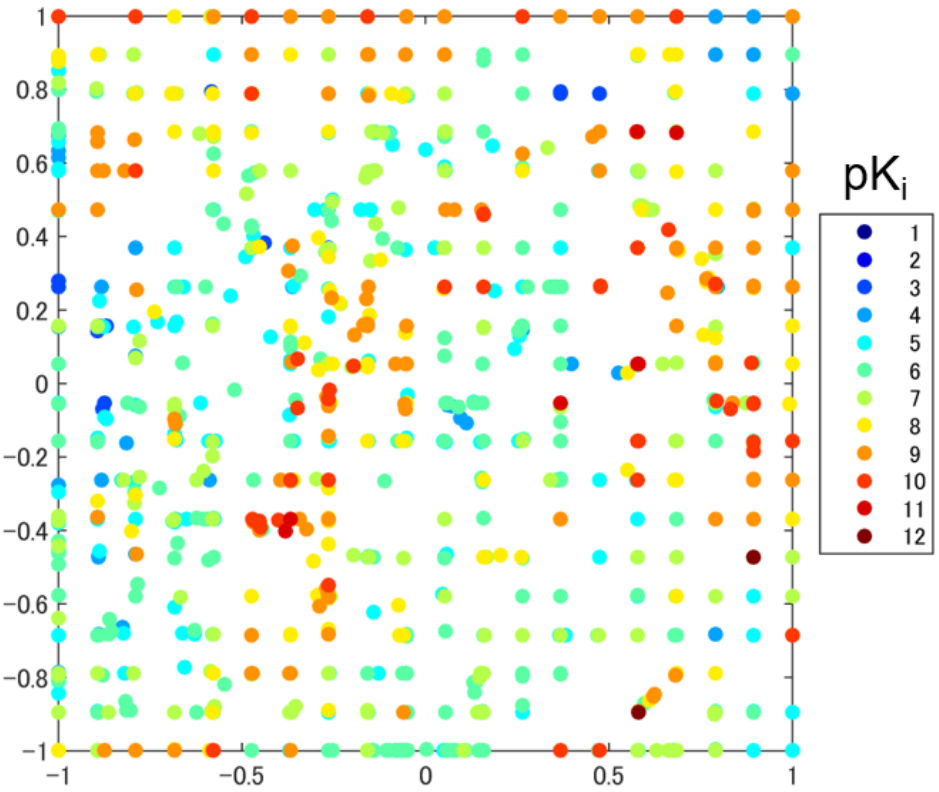


Figure 4-9 Map by GTM with the optimized set of hyper-parameters. Every sample, which is annotated with measured pK_i value, was projected on the map.

Every grid point on the GTM map has the corresponding coordinates in the original descriptor space (27 dimension). The GMM density at the coordinates in the original space is projected on the corresponding grid on the map. That is how PDF in the original descriptor

space is visualized onto the map. The acquired 11 projection of p(**x**|y) on the map for posterior density are shown on Figure F-1 in Appendix F. **Figure 4-10** shows the GTM map on which the density of p(**x**) is projected. On the map, 8 colorful dots represent the centers of Gaussians. High density area of p(**x**) is colored with dark gray, which is highly localized on the area around (-1, -0.6), regardless of the fact that 8 Gaussian centers are distributed on the whole map. This suggests that Gaussian density in high-dimensional space is highly sensitive to the difference of the coordinates. Therefore, even on a well-trained two-dimensional map, the density intensity drastically differs from place to place as explained in section 4-2 . The densities of the center of Gaussians are, C1 2.25×10^{3}, C2 7.49×10^{3}, C3 1.48×10^{2}, C4 9.10×10^{-1}, C5 1.99×10^{1}, C6 5.00×10^{2}, C7 1.67×10^{-6} and C8 2.84×10^{6}. The density of the 8th Gaussian is the highest of the Gaussians. Areas closed to the 8th Gaussian center seem to obtain higher density. As results of inverse analysis of the constructed GMMs/cMLR model, posterior distributions with various target y values are visualized in the same way as p(**x**). **Figure 4-11** shows 11 pictures corresponding to the different target y values (from 2 to 12). The centers of 8 Gaussians are also projected on the GTM map using the same colored label as on the map with p(**x**). These pictures tell that the center of Gaussians moves drastically, and that density on the map differs among p(**x**|y) with different y values. There is a tendency that the higher the y value is the lesser density the map grids possess.

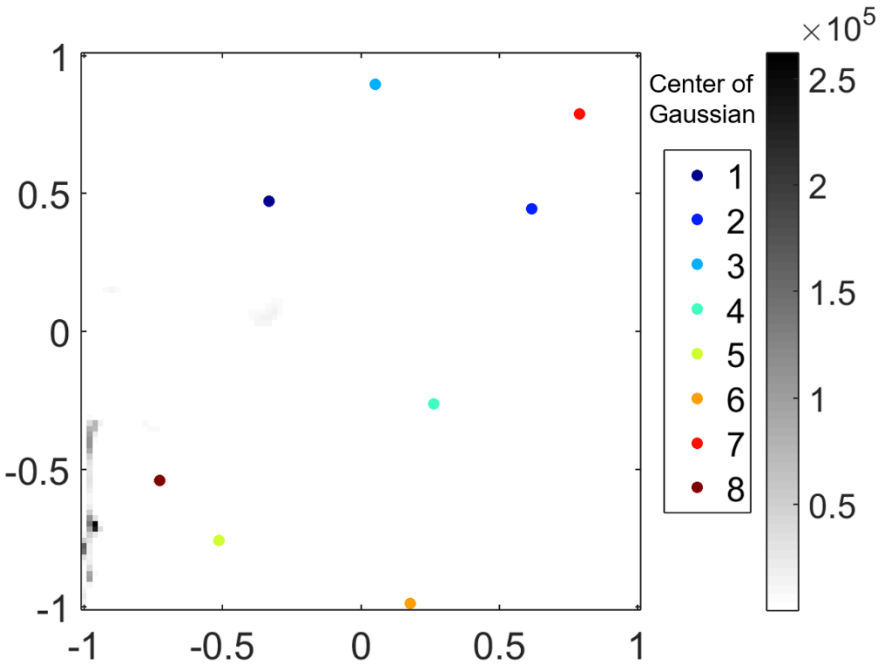
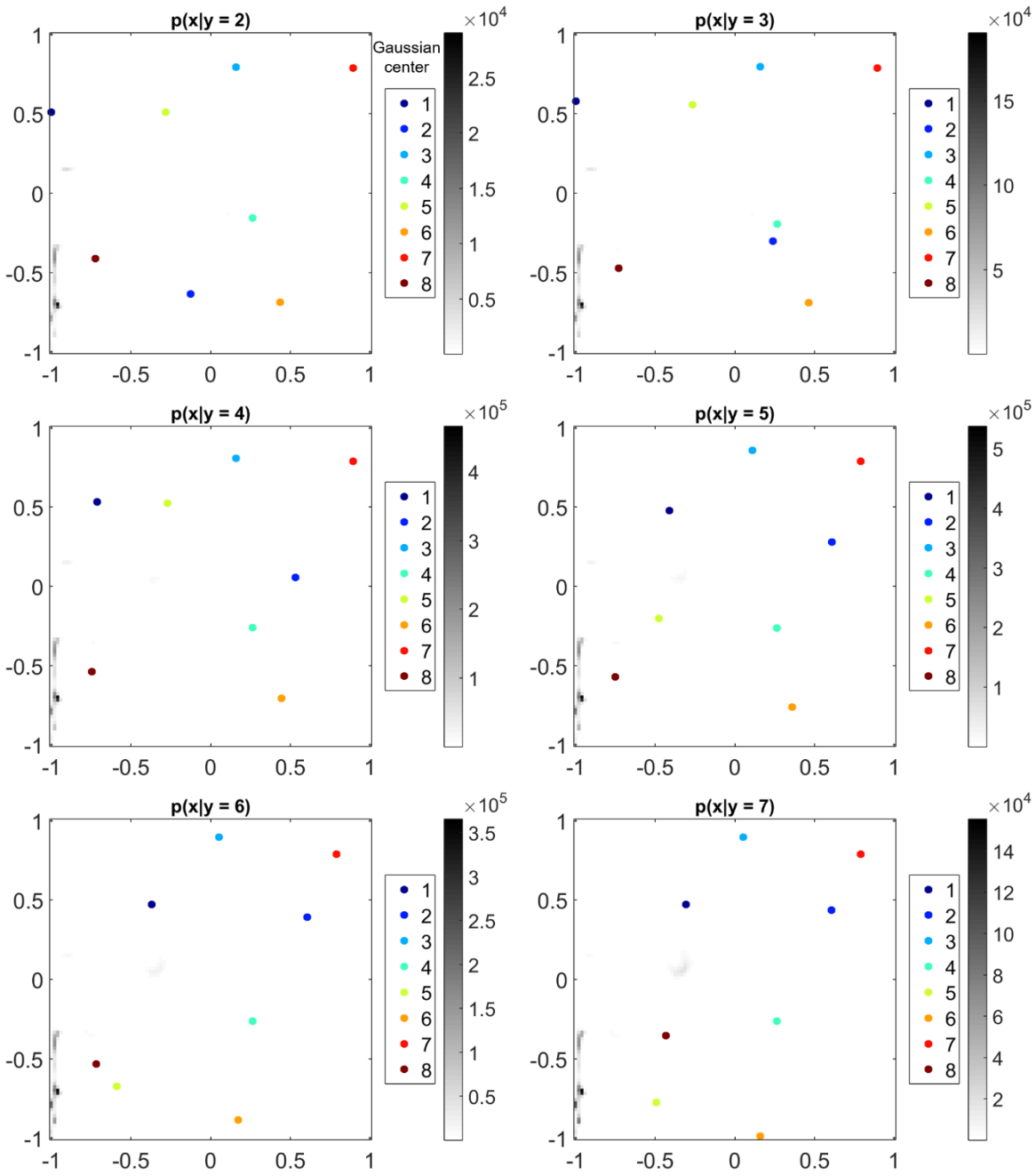


Figure 4-10 Map by GTM. The density of p(x) is projected on the map. The centers of 8 Gaussians are also projected on the map. Grayscale represents the density of a grid.


p(x|y = 2)
p(x|y = 3)
p(x|y = 4)
p(x|y = 5)
p(x|y = 6)
p(x|y = 7)
Gaussian center
1
2
3
4
5
6
7
8

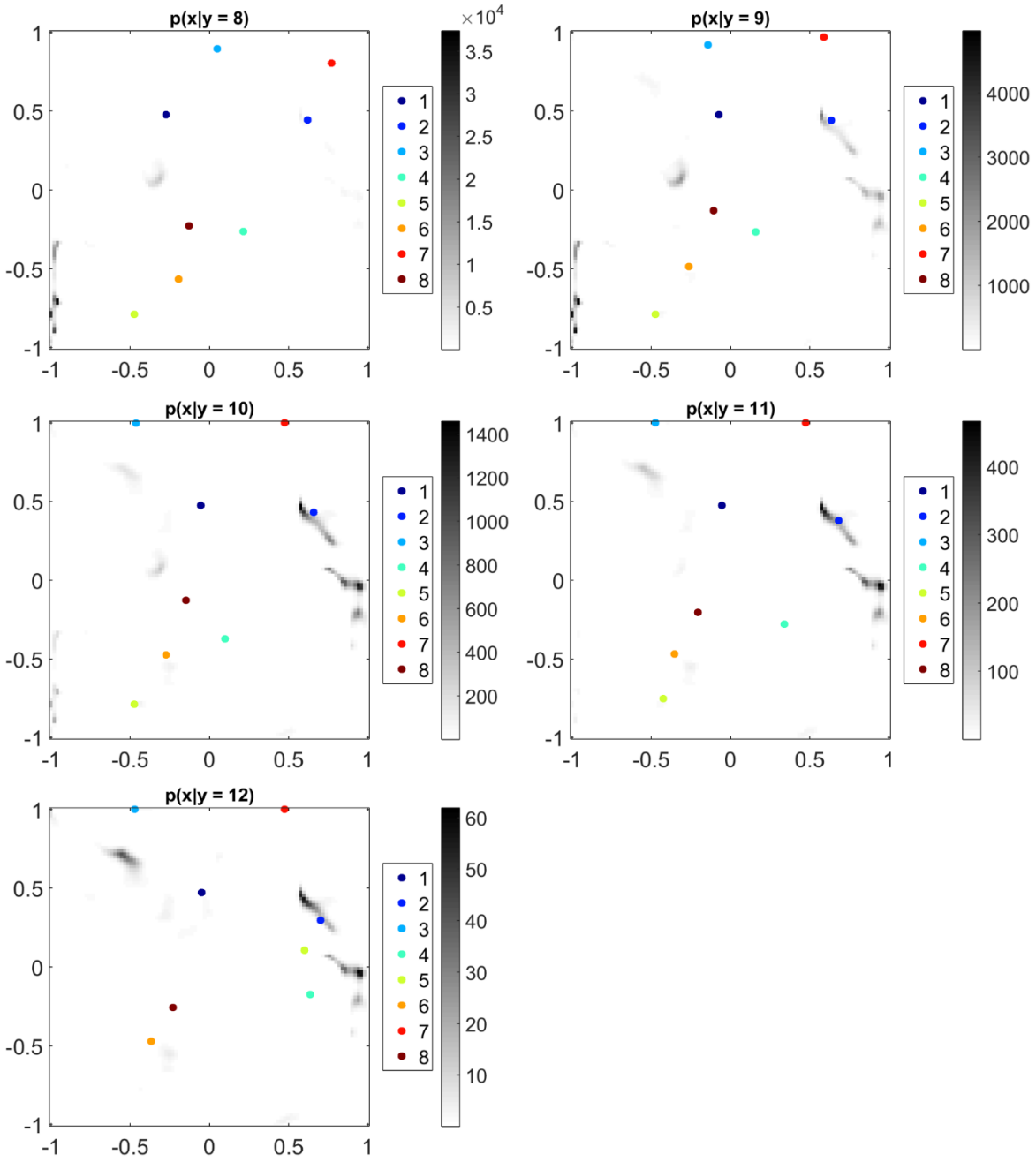
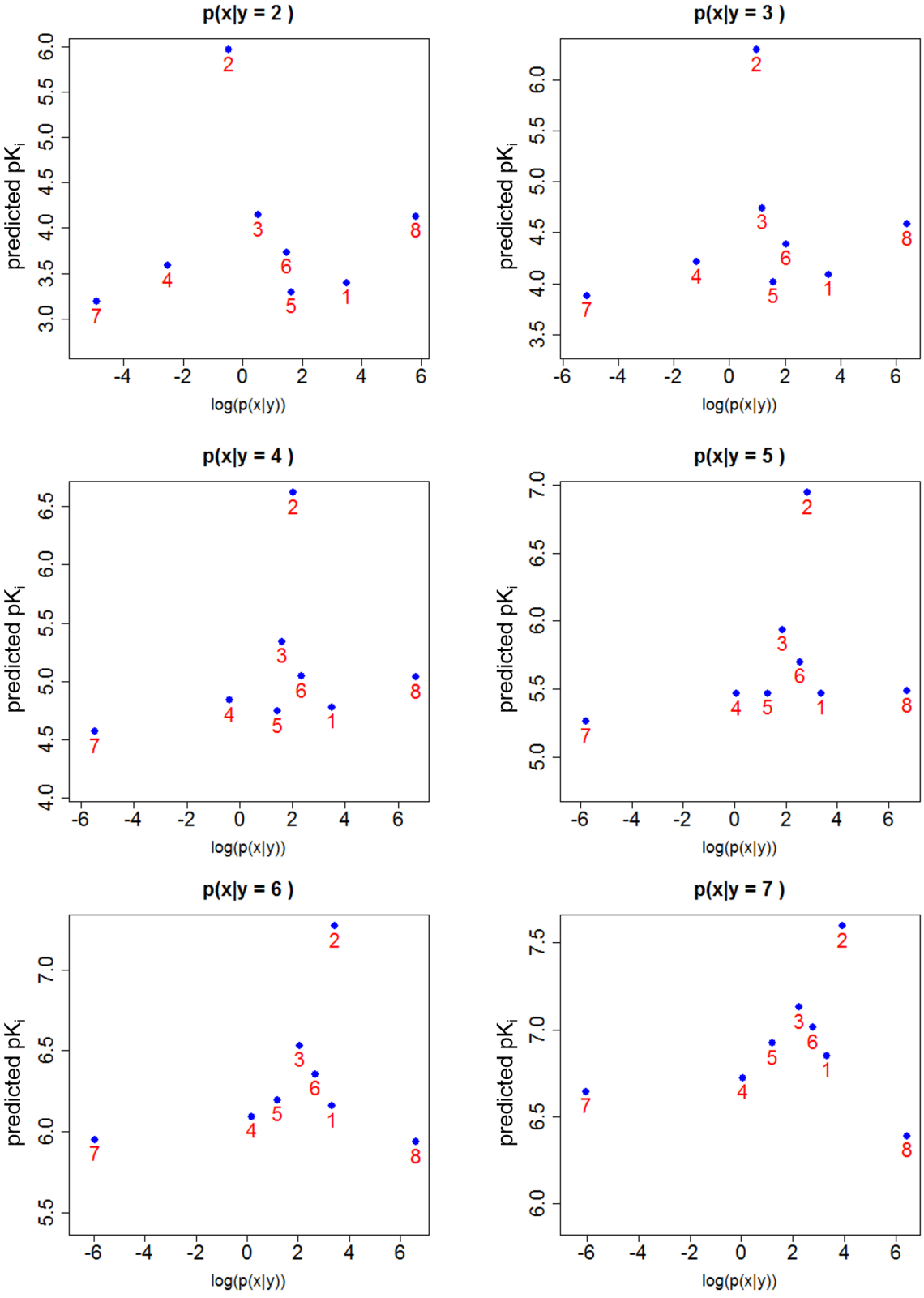


Figure 4-11 Maps by GTM. The density of p(**x**|y) is projected on the maps for various y values. The center of each Gaussian is also projected on the map.

One of the goals in this case study is to propose *de novo* structures that exhibit high y values. These structures should have higher density of p(**x**|y). As discussed in section 3-5-3 , it is usually difficult to determine a threshold for p(**x**|y), because p(**x**|y) inherits both p(**x**) and the set y value. The determination of **x** having high p(**x**|y) and the predicted value close to y is required. By using one of the Gaussian centers, p(**x**|y) is expected to be high. Therefore, a target y value and the Gaussian that is aimed at should be determined based on both factors. For this purpose, predicted y values of Gaussian centers are plotted against the corresponding

densities of p(**x**|y) in **Figure 4-12**. Based on these pictures, target Gaussians were selected as C1 and C8 with the y value is 11 and 9, respectively (i.e. p(**x**|y=9) and p(**x**|y=11)), since C8 has the highest density and C1 seems to have a good balance between density and the predicted y value. In the dataset for constructing the models, there were 366 samples having pK_i values between 8 and 10, 39 samples between 10 and 12 (**Figure 4-3**). Therefore, y = 9 in addition 11 were chosen in this case study.


p(x|y = 2)
predicted pK$_i$
log(p(x|y))
p(x|y = 3)
predicted pK$_i$
log(p(x|y))
p(x|y = 4)
predicted pK$_i$
log(p(x|y))
p(x|y = 5)
predicted pK$_i$
log(p(x|y))
p(x|y = 6)
predicted pK$_i$
log(p(x|y))
p(x|y = 7)
predicted pK$_i$
log(p(x|y))

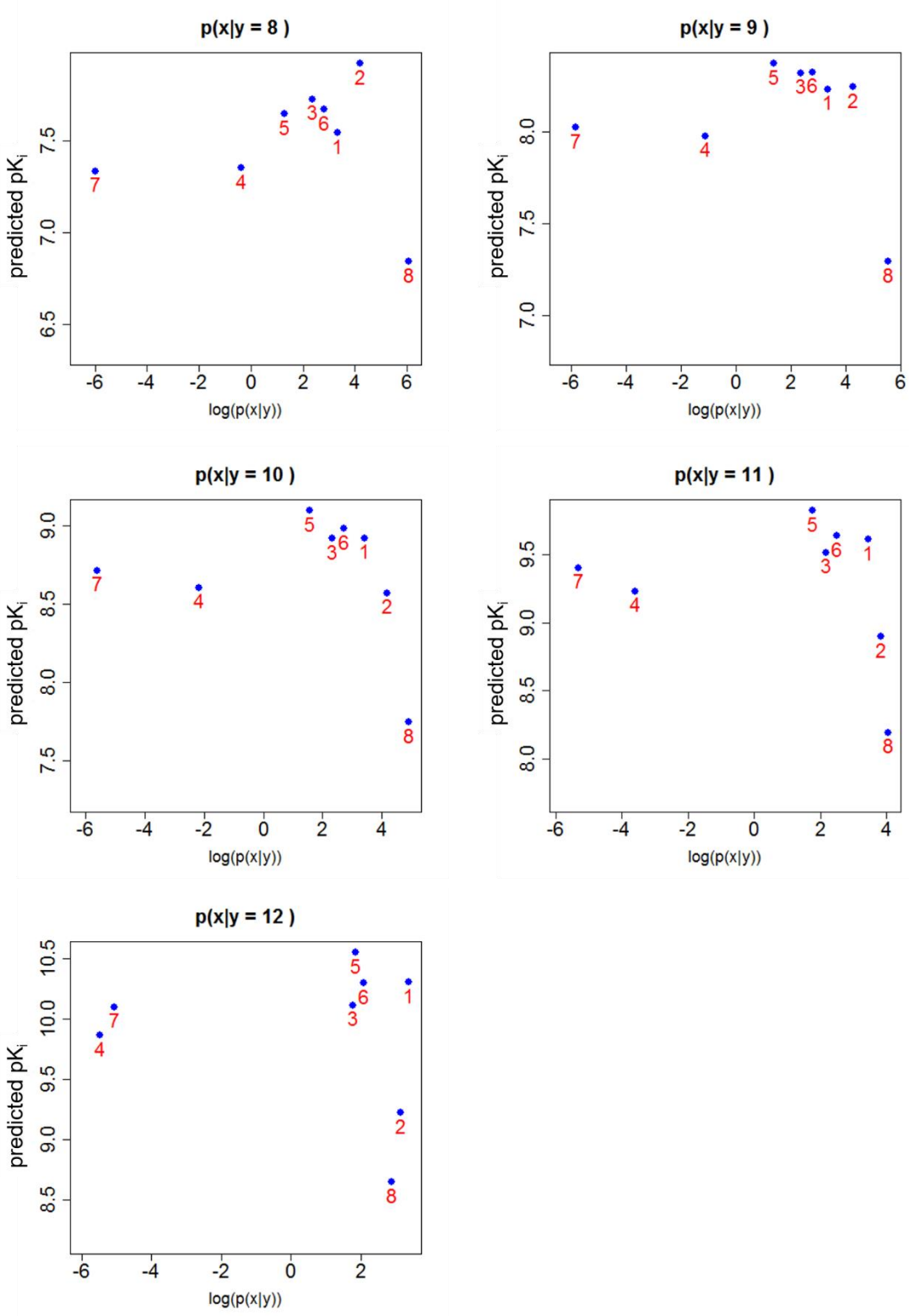


Figure 4-12 Predicted pK_i value against p(**x**|y) with different y values. Numbers on the pictures represent the corresponding Gaussians.

The coordinate sets of the two Gaussians are shown in **Table 4-5**. The densities and predicted pK_i values for these Gaussians are (2.84×10^3, 9.61) for C1 (y =11) and (3.52×10^5, 7.29) for C8 (y = 9).

Table 4-5 Coordinates of the posterior Gaussian centers in C1 (y =11) and C8 (y = 9).

	C1 (y = 11)	C8 (y = 9)
CIC	3.2	3.7
R05	0.5	0.6
aR	2.1	3.6
ZMIV	553.8	466.8
nBM	17.6	22
nHAccLipin	10.1	7.1
nCH2R2	3.6	1.3
nCHR3	0.5	0.0
nCH3R	1.3	1.1
nCH3X	0.2	0.2
nOH	0.1	0.0
n=O	2.1	1.4
ArNR2	0.0	0.0
nArCO	0.4	0.0
TPSA	150.1	106.7
LL	110.3	22.7
LD	223.9	66.8
LP	121.7	25.1
AA	135.3	68.1
AP	102.5	38.9
AN	-0.1	0.0
DD	42.1	27.4
RL	64.1	53
RA	64.9	84.3
RD	67.0	71.2
RP	31.4	25.3
RR	5.0	22.3

Although the densities of the two Gaussian centers on **Table 4-5** are high, they become low once one rounds them. All the descriptors employed here except TPSA cannot take real values, they only take integer values. For example, there is no chemical structure having 3.2 rings. Therefore, accurate density was determined only after conducting rounding operation to these discrete variables. The density for C1 (y = 11) decreased from 2.84×10^3 to 8.56, and for C8 (y =9) from 3.52×10^5 to 1.29×10^5. In order to look for their neighbor integer grid points that exhibit higher density than the ones obtained by rounding operation, the author used eigenvectors of the covariance matrix of the Gaussian. The real number grid points were searched from the center of a Gaussian in the direction of the eigenvectors having large

eigenvalues. Covariance matrix represents the correlation among variables in nature. In order to minimize the reduction of density for searched grid points, considering correlation seems like a good strategy. Eigenvectors of a covariance matrix tell the shape of the Gaussian along with the intensity represented by the corresponding eigenvalues. Therefore, along these eigenvectors, candidate grid points were searched. The criterion for choosing one of the points was the maximum difference of each variable between a point before and after rounding. Only one grid point having the minimum value of the criterion was selected, meaning the coordinates of the searched candidate point change the least after rounding. The procedure of finding the new coordinates is illustrated in **Figure 4-13**. For C1, three eigenvectors having the three highest eigenvalues were used, and for C8 only one eigenvector having the highest eigenvalue was used based on the contribution of eigenvectors. In C1, the contributions of the eigenvectors, which is the ratio of the squared eigenvalue to the summation of all the squared ones, were (48.9%, 25.7%, 12.4%). In C8, the contribution was 59.1%. The obtained two coordinates are on **Table 4-6**. The densities and predicted pK_i values for these newly acquired coordinates were (8.0, 8.97) for C1 (y =11). For C8 (y=9), the center of the Gaussian itself was recognized as the closet point to integer coordinates along the eigenvector, leaving the coordinates unchanged (1.28×10^5, 7.47). For C1, newly acquired density as well as the predicted pK_i value decreased from those of the center of C1 (y =11) after rounding to integer. Nevertheless, every variable of the newly determined coordinate is smaller than that of the center of C1 after rounding. Using small descriptor values as constraints is preferable when considering structure generation, because the proposed structure generator combines building blocks in every possible way until violating one of the upper bounds of the constraints. Since the upper bounds of the constraints are determined based on the coordinates of a Gaussian, smaller coordinates help to prevent the structure generation from combinatorial explosion.

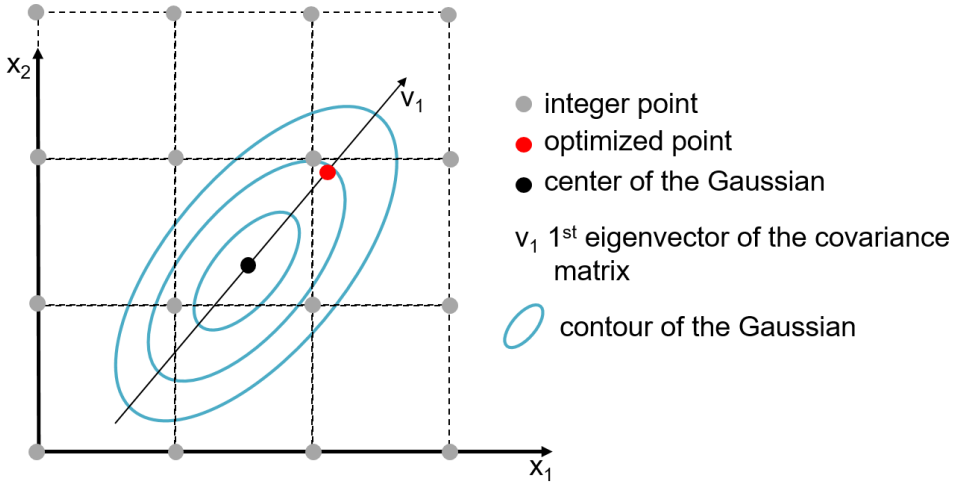


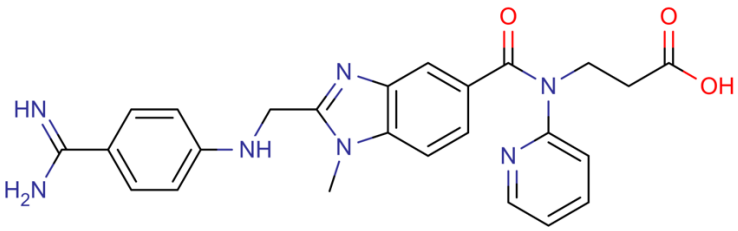
Figure 4-13 Illustration of finding coordinates that are close to integer point along eigenvectors corresponding with high eigenvalues in two-dimensional space (x_1 and x_2).

Table 4-6 Obtained coordinates by searching integer grid pints for the two Gaussians on **Table 4-5**.

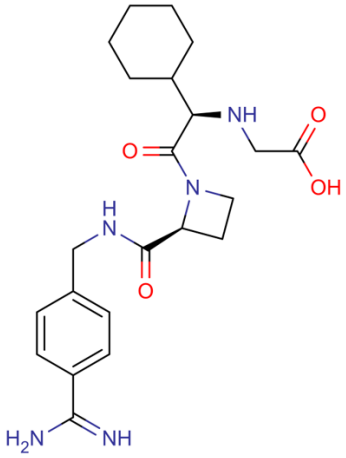
	C1 (y = 11)	C8 (y = 9)
CIC	2	4
R05	1	1
aR	1	4
ZMIV	438	467
nBM	9	22
nHAccLipin	8	7
nCH2R2	4	1
nCHR3	1	0
nCH3R	2	1
nCH3X	0	0
nOH	0	0
n=O	2	1
ArNR2	0	0
nArCO	0	0
TPSA	112	107
LL	125	23
LD	130	67
LP	86	25
AA	122	68
AP	82	39
AN	0	0
DD	1	27
RL	48	53
RA	45	84
RD	24	71
RP	14	25
RR	0	22

4-3-4-1 Drugs as Direct Thrombin Inhibitors

Before describing the results of *de novo* structure generation for these two sets of coordinates, two drugs as direct thrombin inhibitors were analyzed with the constructed inverse QSAR model. The two drugs were dabigatran and melagatran (**Figure 4-14**). Both are direct thrombin inhibitors, however, melagatran has been discontinued its distribution due to hepatotoxicity reported in 2006[159]. Before conducting the analysis, it should be noted that dabigatran was in the training dataset whereas melagatran was not. Reported pK_i of dabigatran to thrombin was 8.3 (K_i: 4.5 nM) and that of melagatran was 8.8 (K_i: 1.7 nM) in the ChEMBL database.



a) dabigatran



b) melagatran

Figure 4-14 Dabigatran and melagatran

The results of calculating the posterior densities of dabigatran and melagatran are shown on **Table 4-7**. For both cases, p(**x**|y) captured the feature of the closeness to the predicted y values for them. p(**x**) of dabigatran was 5.64E-08, and that of melagatran was 6.58E-04. It is fair to say dabigatran was out of AD, but we could not conclude that melagatran was also out of AD because of the p(**x**) value. The predicted pK_i for dabigatran was 7.66 and that for melagatran was 5.47. Based on the measured values of these two structures, the prediction for dabigatran succeeded whereas that for melagatran failed. Although prediction for dabigatran worked well, in inverse analysis it is not possible to reconstruct dabigatran based on the proposed workflow because p(**x**|y) given the pK_i at 11 is still small compared to other promising sets of coordinates shown in **Figure 4-12** (i.e. centers of Gaussian).

Table 4-7 Prediction results for dabigatran and melagatran

Target y	Dabigatran		Melagatran	
	p(**x**|y)	p(**x**|y)/p(**x**)	p(**x**|y)	p(**x**|y)/p(**x**)
2	1.55E-13	2.75E-06	2.67E-05	4.06E-02
3	3.15E-11	5.58E-04	1.93E-04	2.93E-01
4	1.41E-09	2.51E-02	6.03E-04	9.17E-01
5	1.77E-08	3.13E-01	1.03E-03	1.56E+00
6	7.69E-08	1.36E+00	1.19E-03	1.81E+00
7	1.24E-07	2.19E+00	9.96E-04	1.51E+00
8	6.65E-08	1.18E+00	5.45E-04	8.27E-01
9	1.17E-08	2.07E-01	1.90E-04	2.89E-01
10	6.99E-10	1.24E-02	4.43E-05	6.73E-02
11	1.25E-11	2.21E-04	6.02E-06	9.15E-03

The fact that predicted error for melagatran exhibits 3 times higher than the RMSE of the QSAR model indicates a limitation of this two-dimensional-based QSAR model. The 5 closest structures for melagatran in the training dataset were shown in **Figure 4-15**. The two closest structures (ChEMBL 81056 and ChEMBL 81844) to melagatran are identical in their two-dimensional forms (i.e. elimination of stereo information), but exhibit significant difference in pK_i. Although the proposed MCDs succeeded in capturing structural similarity between melagatran and these structures, the current proposed methodology cannot take the three-dimensional structural difference into account when predicting affinity.

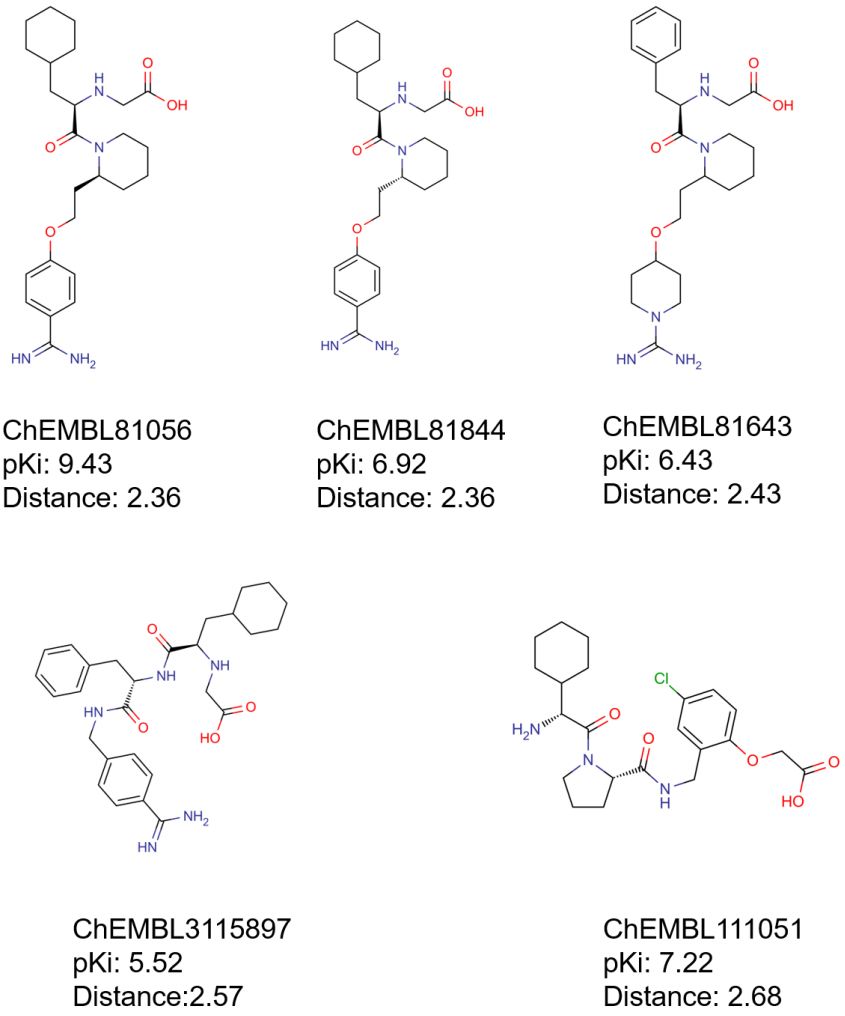


Figure 4-15 The five closest compounds to melagatran in the training dataset. pK_i is a measured value. Distance is Euclidean distance to the melagatran in 27 descriptor space after scaling.

4-3-5 Structure Generation

Two structure generation operations were conducted aiming to generate *de novo* structures having descriptor values similar to the ones on **Table 4-6**. The following two sub-sections are for the results from the two generation operations respectively (C8 (y =9) and C1 (y =11)). Structure generation was conducted on a CentOS6.5 personal computer with Intel Xeon E5-2680v2×2 and 64 GB RAM.

4-3-5-1 Generation for C8 (y = 9)

Ring systems were obtained by DecomposeRingFragments in Molgilla, which decomposed all the 1,705 molecules in the dataset according to the workflow in **Figure 4-2**. As a result, 301 unique ring systems, having the maximum number of access points 7, and the maximum contained heavy atoms 20, were obtained. Atom types of atom fragments were C, N, O, F, Cl, Br, and I with a single valence rule. Constraints for structure generation were on the **Table 4-8**. The ranges were determined based on the 0.2 times standard deviation of all the 1,705 molecules. Structure generation conditions by FragmentGenerator in Molgilla are as follows:

- The number of fragments to be combined to make a structure was from 2 to 15.
- The number of maximum ring systems in a structure was 5.
- Probability with which stochastic generation is conducted was 0.2.
- Used substructures on the taboo list were X_X, =C=, Rs_Rs, MR, TMC, RDBO, C__N, C__OX, CH2X, CH__O, and AlC_OAl (Table D1 in Appendix D).

Three trials were conducted with different initial conditions for the random number generator.

Table 4-8 Descriptor constraints for structure generation aiming at C8 (y=9). Lower-upper bounds are listed

Constraint*	Lower bound	Upper bound
CIC	4	4
R05	1	1
aR	4	4
ZMIV	442	492
nBM	21	23
nHAccLipin	7	7
nCH2R2	0	2
nCHR3	0	0
nCH3R	1	1
nCH3X	0	1
nOH	0	0
n=O	1	1
ArNR2	0	0
nArCO	0	0
TPSA	99	115
LL	0	56
LD	31	103
LP	10	40
AA	43	93
AP	27	51
AN	0	3
DD	12	42
RL	42	64
RA	68	100
RD	62	80
RP	21	29
RR	19	25

* MCDs as constraints are defined in Appendix E.

The results of structure generation were shown on **Table 4-9**. Without introducing stochastic generation, the number of searched structures during structure generation would be over 4×10^{16}. Therefore, in this case study, exhaustive generation would be impossible with FragmentGenerator in Molgilla.

Table 4-9 Generation results for C8 (y=9) in three trials.

	trial 1	trial 2	trial 3
Nodes[*1]	8.55E+09	7.98E+09	8.29E+09
Structures[*2]	45	26	15
Expected nodes[*3]	4.88E+16	4.46E+16	4.94E+16
Expected structures[*4]	3.44E+09	5.94E+08	9.24E+08
Time [s]	4.56E+04	4.03E+04	3.89E+04

[*1] Number of structures during generation; [*2] number of structures satisfying the constraints; [*3] expected number of structures during generation; [*4] expected number of structures satisfying the constraints.

Next step is to analyze the generated structures in terms of posterior distribution as well as predicted pK_i values. All the generated structures were confirmed to be unique based on string comparison among canonical representations of the generated structures (86 unique structures).

The relation between predicted pK_i values and $p(\mathbf{x}|y=9)$ of the generated 86 structures are shown in **Figure 4-16**. All 86 structures are successfully categorized in the C8 cluster. The Pareto optimal solutions are a and b on **Figure 4-16**. Corresponding chemical structures for a and b are shown in **Figure 4-17**. These two structures imply the limitation of the current version of *FragmentGenerator* in *Molgilla*. In this case study, constraints for the structure generation required the generated structure possessing positively charged points. The only atom fragment that is recognized as positively charged point is NH_2. Therefore, these structures contain one NH_2. This simplified rule ignores the functional groups in a molecule. In structure b, NH_2 forms an amide, meaning the NH_2 is not protonated. In structure a, NH_2 appears in the 2-aminopyridine. This NH_2 or the nitrogen atom in pyridine may be positively charged because of taking two tautomeric forms despite the fact that the lone pair on NH_2 is delocalized on pyridine. However, making positively charged point in this way is not what *FragmentGenerator* works. Moreover, the program wrongly recognizes anilines as positively charged points. Adopting more precise definition of PPPs is possible in exchange for computational efficiency.

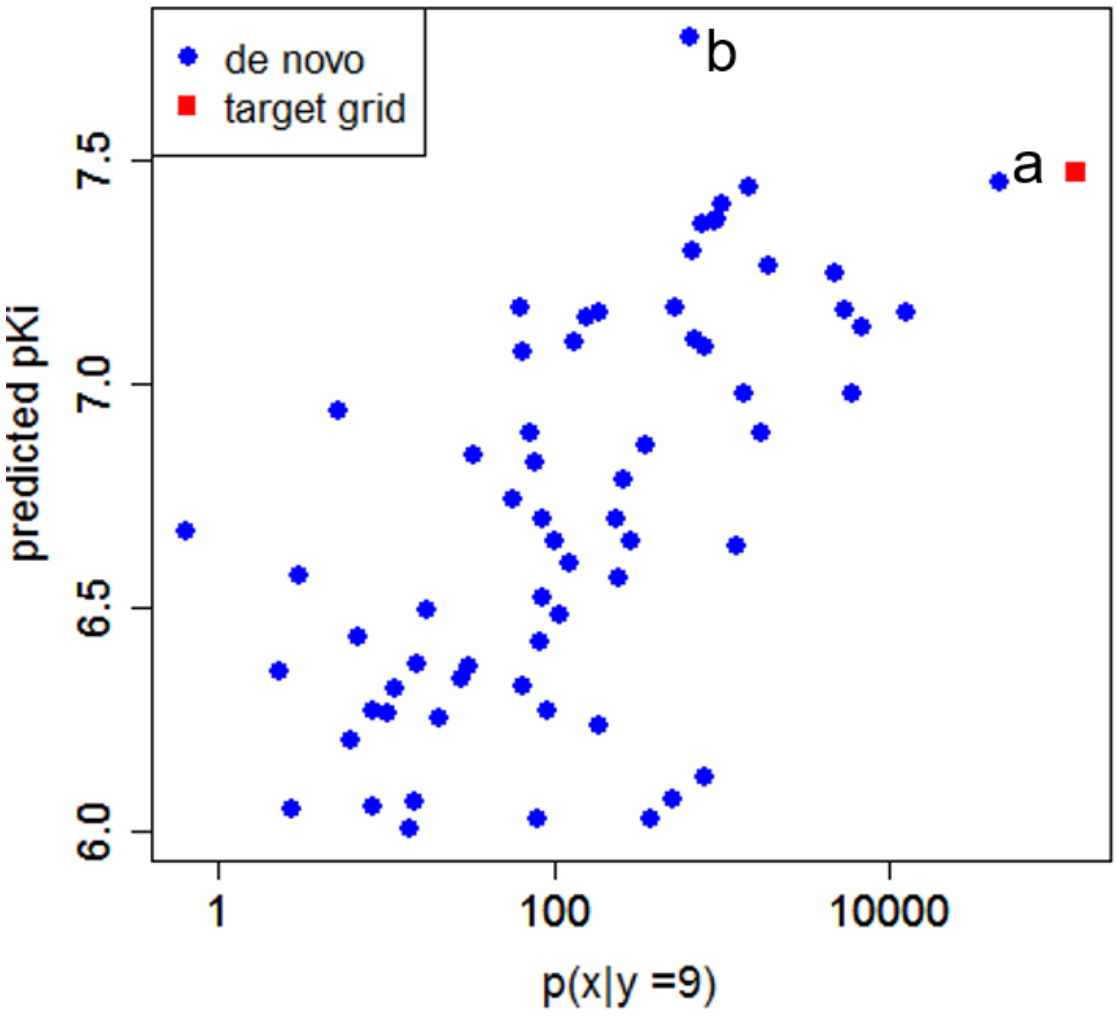


Figure 4-16 Predicted pK_i against p(**x**|y=9) of the generated structures (blue dots) and the target grid point mentioned on **Table 4-6** (a red dot).

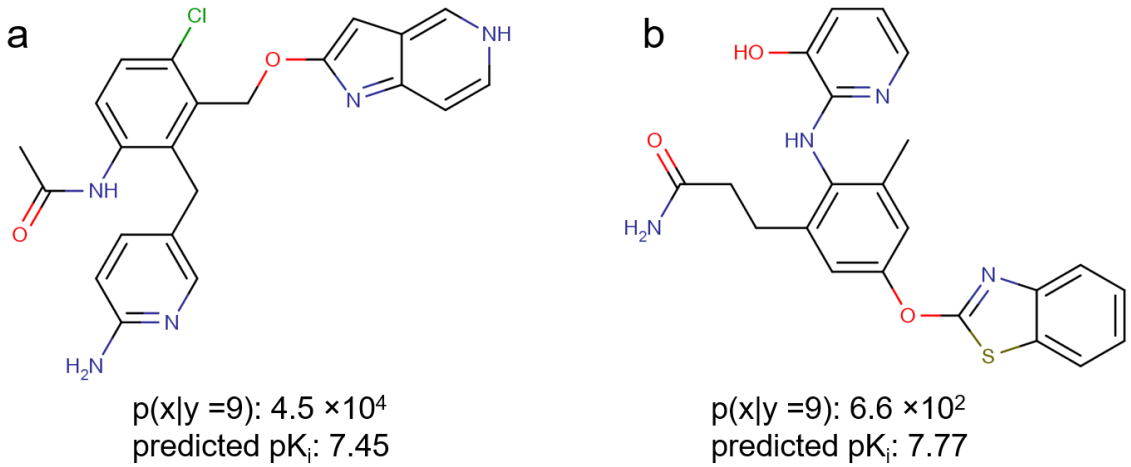


Figure 4-17 Chemical structures existing at the Pareto solutions between pK_i and p(**x**|y).

4-3-5-2 Generation for C1 (y = 11)

DecomposeRingFragments in Molgilla made 289 ring systems. In contrast to the previous case study (*i.e.* C8 (y = 9)), the maximum number of access points in a ring system was set to four. The maximum number of heavy atoms in a ring system is still set 20. Atom types of atom fragments were C, N, O, F, Cl, Br, and I with a single valence rule. Constraints for structure generation were on the **Table 4-10**. The ranges were determined based on the 0.2 times standard deviation of all the 1,705 molecules. Structure generation conditions by FragmentGenerator in Molgilla are as follows:

- The number of fragments to be combined to make a structure was from 2 to 15.
- The number of maximum ring systems in a structure was 5.
- Probability with which stochastic generation is conducted was 0.4.
- Used substructures on the taboo list were X_X, =C=, Rs_Rs, MR, TMC, RDBO, C__N, C__OX, CH2X, CH__O, AlC_OAl, Csp3, CHsp3_3Rings, and Nsp3_3Rings. (Table D1 in Appendix D).

Table 4-10 Descriptor constraints for structure generation aiming at C1 (y=11). Lower-upper bounds are listed

Constraint*	Lower bound	Upper bound
CIC	2	3
R05	1	1
aR	1	1
ZMIV	413	463
nBM	8	10
nHAccLipin	7	9
nCH2R2	2	5
nCHR3	1	1
nCH3R	1	2
nCH3X	0	0
nOH	0	0
n=O	1	2
ArNR2	0	0
nArCO	0	0
TPSA	104	120
LL	92	158
LD	94	166
LP	71	101
AA	97	147
AP	70	94
AN	0	0
DD	0	16
RL	37	59
RA	29	61
RD	15	33
RP	10	18
RR	0	0

* MCDs as constraints are defined in Appendix E.

The results of structure generation were shown on **Table 4-11**. Without introducing stochastic generation, the number of searched structures during structure generation would be over 6×10^{15}. In this case, the expected number of structures satisfying the constraints would reach 1.41×10^{8}. With the current computational power, we could not conduct more computational-power consuming analysis, such as docking simulation, for all structures. Therefore, in this case study, exhaustive generation should be avoided at this current computational level.

Table 4-11 Generation results for C1 (y=11) in three trials.

Nodes[*1]	4.25E+11
Structures[*2]	1739
Expected nodes[*3]	6.19E+15
Expected structures[*4]	1.41E+08
Time [s]	7.32E+05

[*1] Number of structures during generation; [*2] number of structures satisfying the constraints; [*3] expected number of structures during generation; [*4] expected number of structures satisfying the constraints.

Next step is to analyze the generated structures in terms of posterior distribution as well as predicted pK_i values. The relation between predicted pK_i values and $p(\mathbf{x}|y=11)$ of the generated 1,739 structures are plotted. All 1,739 structures were successfully categorized in the C1 cluster. The Pareto optimal solutions, including the target grid (red square), are from a to e on **Figure 4-18**. Chemical structures are shown in **Figure 4-19**. All structures contain two =Os. b and d have a 2,3-piperidinedione substructure. In order to check the effect of the =O variable, the histograms of $p(\mathbf{x}|y=11)$ is shown in **Figure 4-20** based on the value. Although there are less structures having two for =O, these structures show high density than those having one for =O.

Another feature of the generated structures shown in **Figure 4-19** is that the proposed structures all have two NH_2 atom fragments in order to obtain relatively large values for LP, AP, and RP. Conformation of ligands has been well studied for thrombin antagonists[160,161]. The Asp189 in the S1 pocket of thrombin is negatively charged. Ionic interaction between that residue and the ligands is said to be responsible for high affinity. Therefore, positively charged motifs, such as guanidine or benzamidine, have been brought. The lower bounds of the three STDPs (LP, AP, and RP) are 71, 70, and 10, respectively. Therefore, the majority of the generated structures have two NH_2. When looking for the structures having one NH_2 in the generated structure pool, 70 were found. Top three structures based on either predicted pK_i or $p(\mathbf{x}|y)$ are shown in **Figure 4-21**. These structures might become candidates for further analysis. On the top row in **Figure 4-21**, structure a and b are identical in terms of descriptors for constructing models. The only difference between them is the type of halogen atoms (*i.e.* a has an iodine, whereas b has bromine). Both iodine and bromine are recognized as lipophilic points. Other descriptors, such as MW, must be installed for distinguishing them. MW was not chosen for this analysis as a result of variable selection. Determining a proper set of descriptors is one of future challenge in this study.

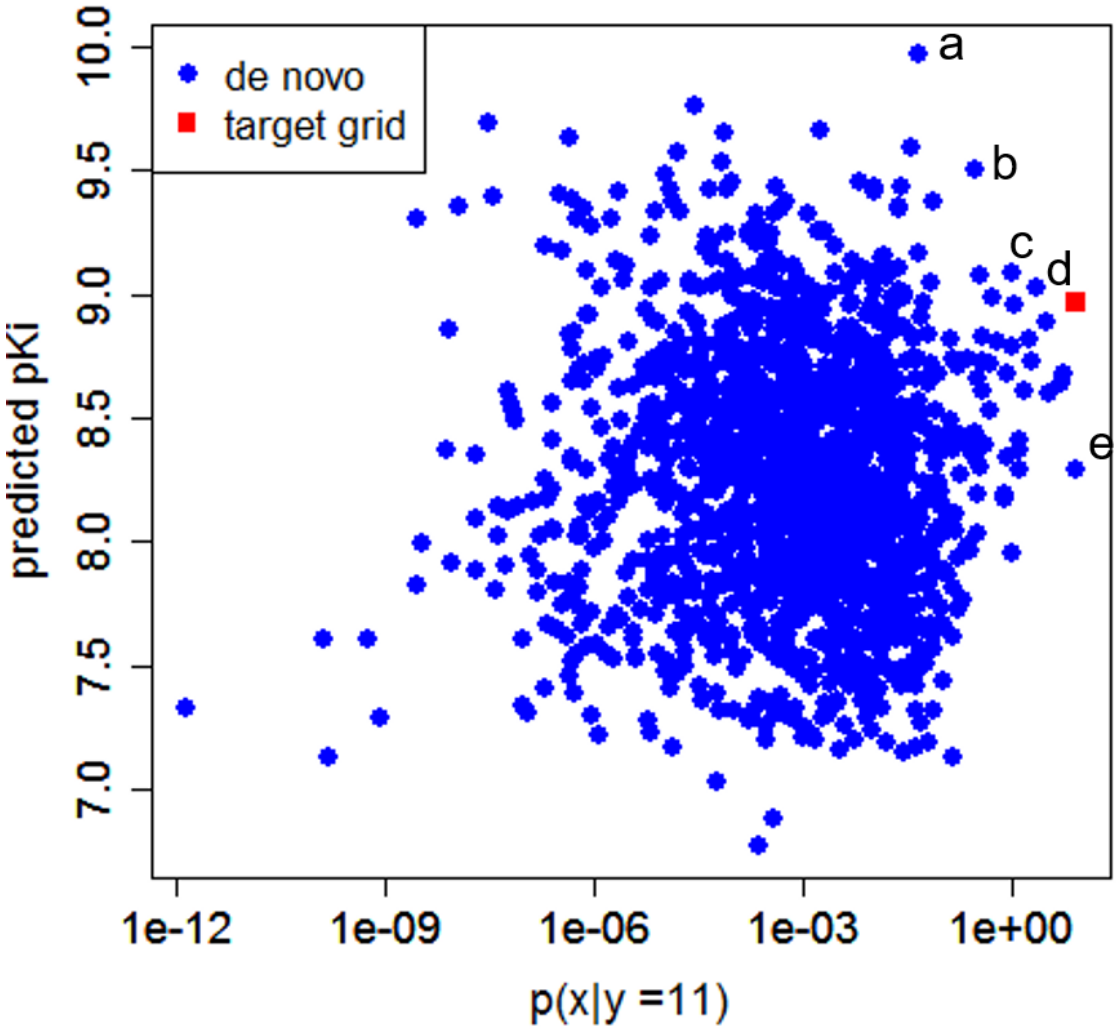


Figure 4-18 Predicted pK_i against p(**x**|y=11) of the generated structures (blue circles) and the target grid point mentioned on **Table 4-6** (a red square). Marked dots are Pareto solutions.

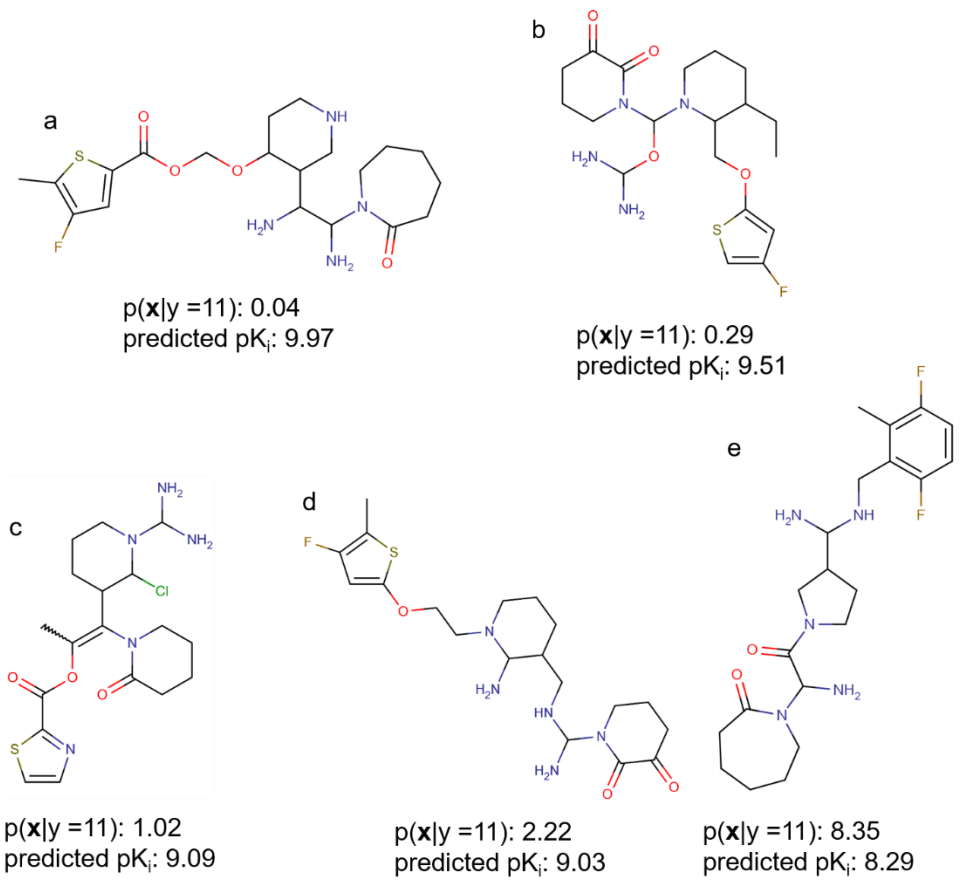


Figure 4-19 Chemical structures corresponding to the Pareto solutions in **Figure 4-18**.

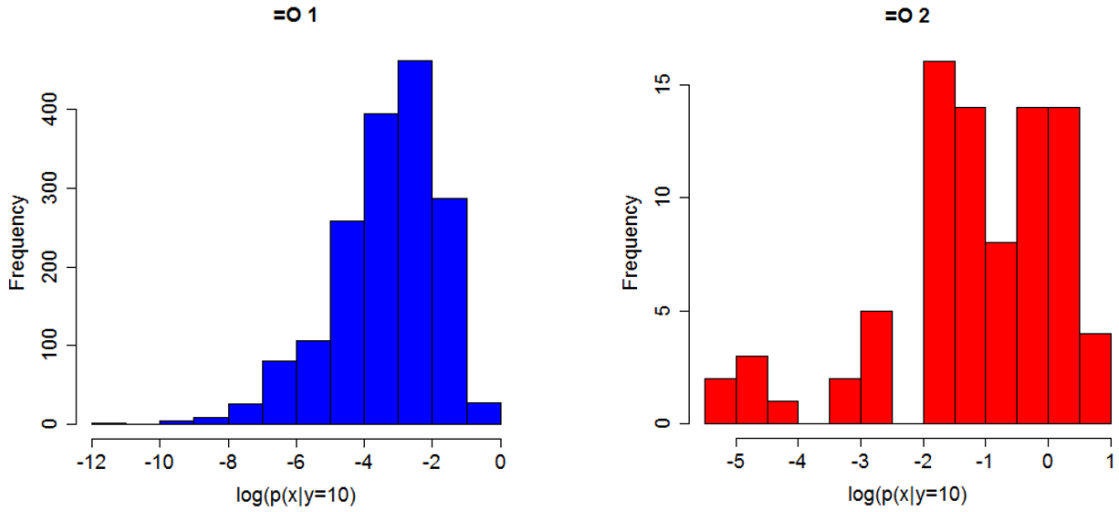


Figure 4-20 Histograms of the posterior densities p(**x**|y = 9) of generated structures depending on the =O value (1 left, 2 right).

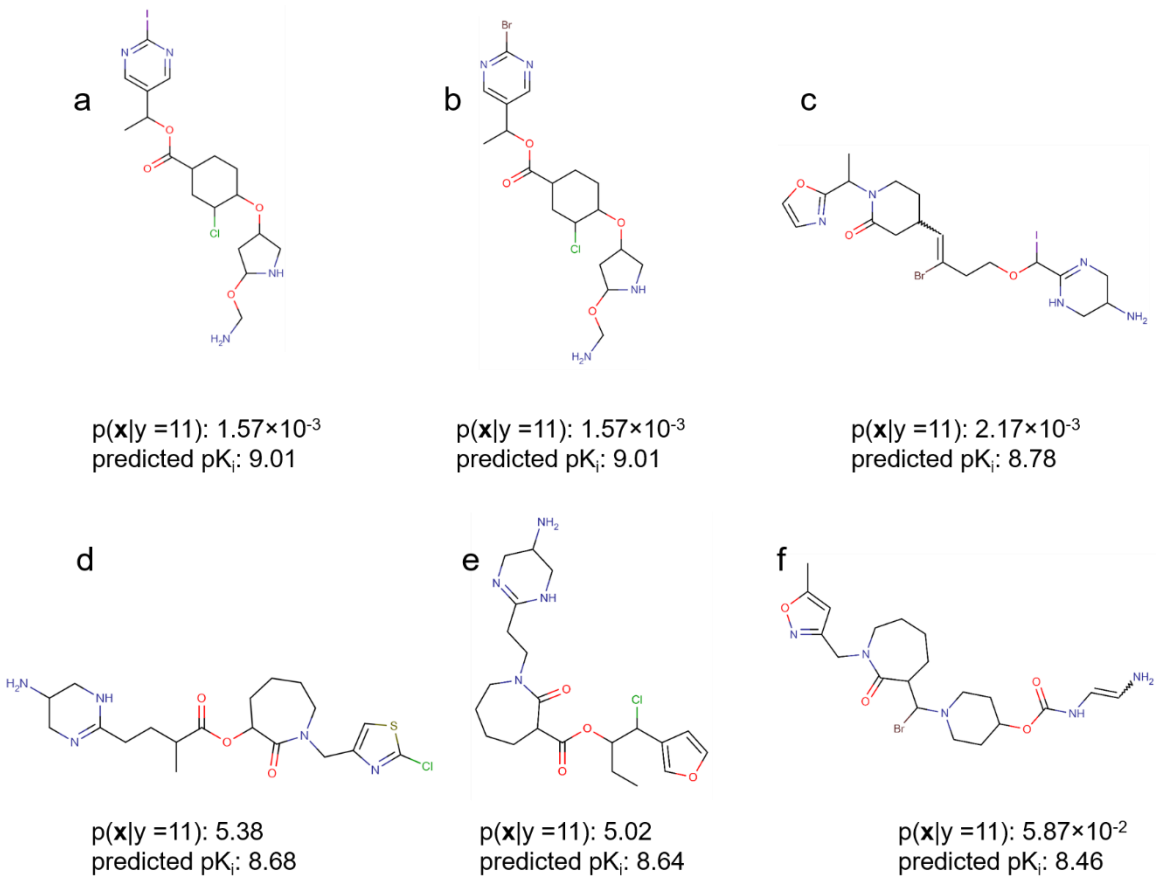


Figure 4-21 Selected structures having one NH_2 atom fragments. Upper: structures exhibiting the three highest pK_i values of the 70 structures having only one NH_2 atom fragments, Bottom: the three top structures based on p(**x**|y).

4-4 Conclusion

In this chapter, a chemical structure generation system based on inverse QSPR/QSAR has been proposed. The system consists of two independent parts: inverse analysis of a QSPR/QSAR model and structure generation from constraints. For the inverse analysis of a QSPR/QSAR model, GMMs/cMLR, which was introduced in CHAPTER 3, is employed. GMMs/cMLR is able to represent the nonlinear relationship between **x** and y by combining plural linear regressions. It can derive the posterior PDF of **x** given y as a closed-form solution (*i.e.* GMM). For the structure generation part, structure generator system *Molgilla*, which is explained in details in CHAPTER 2, can be employed. Structure generator *FragmentGenerator* in *Molgilla* constructs chemical graphs by combining ring systems and atom fragments exhaustively without making duplicates. It can prune a branch in a generation tree consisting of growing chemical graphs based on values of MCDs. The branch connects a parent to its child graphs. Introducing pruning to the branches contributes to the reduction of the possible solutions to be searched.

In order to combine these two methodologies into a system, which is the proposed structure generation system based on Inverse QSPR/QSAR, MCDs must be adopted for constructing a QSPR/QSAR model. Furthermore, structure generation constraints defined by the MCDs' ranges, which forms a hyper-rectangle in high-dimensional space, should be determined carefully. They should take both predicted values and AD into consideration. Inside the hyper-rectangle, only $p(\mathbf{x}|y)$ should exhibit high value, but still it is hard to determine the threshold for it because density of Gaussian distributions is sensitive to subtle changes of coordinates in high-dimensional space.

As a proof-of-concept case study, thrombin ligand design based on inverse QSAR was carried out. The proposed approach resulted in chemical structures having PDF values equivalent to that of the targeted coordinate derived from the center of a Gaussian. The Gaussian in the GMM for $p(\mathbf{x}|y)$ was determined based on predicted pK_i values and density of the coordinate. Because of the sensitivity of Gaussians mentioned in the paragraph above, the updated coordinates obtained by rounding to the original Gaussian center shows far smaller density. Therefore, dominating eigenvectors of the covariance matrix of a Gaussian guided another set of integer coordinates, which shows a relatively high density. *FragmentGenerator* succeeded in generating chemical structures satisfying the constraints for two regions (C8 (y=9) and C1 (y =11)) with the stochastic generation option. A handful of the generated structures were inspected. They show preferable $p(\mathbf{x}|y)$ as well as predicted pK_i values. This case study revealed at least three challenges for the current structure generation system: improper combination of MCD values, too many searched structures during generation and insufficient MCD description ability.

First, it is possible that there are no possible chemical structures satisfying the constraints determined, such as the number of aromatic rings being 2 and the number of rings 1. This type of contradiction among descriptors could happen. Other constraints considering the consistency of MCDs should be introduced, like Churchwell *et al.* introduced consistency equations in their proposed inverse analysis methodology[53]. It should be noted that the probability of such self-contradictions occurring is not high since the proposed methodology uses the prior distribution to derive posterior distributions.

Second, even though the number of structures satisfying constraints is not too big (over 10^9), exhaustive generation was intractable when the constraints could not contribute to the reduction of the number of nodes. Since the calculation time is proportional to the number of nodes during generation, it is important to eliminate as early as possible branches that cannot lead to structures satisfying the constraints. It is necessary to improve the generator regarding generation algorithms.

Third, as explained in section 4-3-5 , MCDs should be carefully chosen in order to distinguish different chemical structures if these structures need to be distinguished. Since descriptors are abstract representation of molecules, plural structures matching the same descriptor value exist. The important thing is to know well the object of an application.

Although there is room for improvement in the points mentioned above, the structure generation system based on inverse QSPR/QSAR might be a good tool for generating chemical graphs exhibiting a desired property/activity based on the model.

CHAPTER 5 Summary and Perspective

5-1 Summary

In this thesis, a chemical structure generation system based on inverse QSPR/QSAR analysis has been developed. Inverse QSPR/QSAR means analyzing pre-constructed QSPR/QSAR models inversely to obtain chemical structures exhibiting properties or activities that a chemist expects. Contrary to its simple definition, methodologies for inverse QSPR/QSAR are limited and difficult to develop because of complicated mapping relations both between **x** and y, and between chemical structures and **x**. Because pre-images for these relations cannot be precisely defined, methodologies to approximate the relations are necessary. A proposed system for solving inverse QSPR/QSAR problems consists of two parts, tackling the two mapping problems one by one. The first part is to acquire **x** information from a y value, and the second one is to construct chemical structures based on constraints derived from the **x** information. These two parts are described in detail in CHAPTERS 2 and 3.

In CHAPTER 2, structure generation methodologies are described. Reduced graph-based generation is proposed to construct chemical structures by combining ring systems and atom fragments. Because no efficient algorithms exist to combine building blocks with arbitral symmetry, an algorithm tailored for this purpose is proposed in Section 2-5. A structure generator in which the proposed algorithm is implemented can more rapidly generate chemical structures by combining building blocks than a simple fragment-combined-based structure generator.

When considering structure generation in inverse QSPR/QSAR analysis, a structure generator has to take descriptor values into consideration during structure generation. In other words, the same descriptors should be used in both inverse QSPR/QSAR analysis and structure generation. To construct QSPR/QSAR models with high predictability, a wide range of descriptors should be available. For this purpose, monotonously changing descriptors (MCDs) were introduced. The motivation for introducing MCDs was to efficiently reduce the number of generated structures without sacrificing exhaustiveness in structure generation. In Section 2-6-3, the description ability (i.e., predictability) of MCDs is demonstrated by comparing them with Dragon descriptors from which three-dimensional (3D) descriptors are excluded. It is concluded that the model predictability with MCDs is slightly inferior to that with Dragon descriptors. Therefore, constructed QSPR/QSAR models with MCDs are expected to have acceptable predictability, which makes inverse QSPR/QSAR analysis practical.

Regarding the scenarios where exhaustive structure generation is not possible with current computational power, diversity-oriented generation or reducing the number of generated structures is necessary. In Section 2-7, two algorithms are provided: pseudo framework-based generation and stochastic generation. Pseudo framework-based generation makes use of atom-based frameworks. The proposed algorithm can generate one structure per pseudo-framework defined by the user. The diversity of the structures generated by the proposed

algorithm is higher than that of the corresponding exhaustive structures in terms of molecular access system keys. Stochastic generation can estimate the number of structures to be generated without actual exhaustive generation. It is also useful for reducing generated structures, which is important for practical applications.

In CHAPTER 3, an inverse analysis methodology for retrieving **x** information from y as a probability density function is described. The applicability domain (AD), which means areas in chemical space where prediction by a QSPR/QSAR model is reliable, should be considered when applying QSPR/QSAR models. Gaussian mixture models and cluster-wise multiple linear regression (GMMs/cMLR) is proposed for this purpose. It constructs a single multiple linear regression (MLR) model for each cluster determined by a Gaussian mixture model. The methodology is explained in Section 3-2, followed by three proof-of-concept case studies, which aim to demonstrate the ability of GMMs/cMLR in different aspects. Before constructing cluster-wise MLR models, density estimation for the training data should be carried out with GMMs. Owing to the features of the Gaussian function, the density model and regression models can easily be combined to derive various probability distributions, including $p(\mathbf{x}|y)$ which is proposed as a criterion for representing the AD considering a specific target y value. One of the goals of this chapter is to confirm whether $p(\mathbf{x}|y)$ is suitable for this criterion. The first case study involves construction of QSAR models for four alpha-adrenoceptors (Section 3-5-1). It shows that GMMs/cMLR not only has higher predictability than MLR, but also that GMMs/cMLR models can be interpreted the same way as other linear regression models. Linear regression can be understood by the intensities of the regression coefficients obtained. Because GMMs/cMLR constructs each MLR model with training data classified into a cluster by GMMs, it can be interpreted in the same way as MLR. In this case study, it should be noted that careful interpretation should be performed based on regression coefficients because they are sometimes not consistent among clusters. The second case study involves generating regression models given simulation data using GMMs/cMLR and MLR. The data was compiled to show a nonlinear relationship between **x** and y. Through this case study, GMMs/cMLR succeeded in capturing the nonlinearity in contrast to MLR. The last case study relies on the aqueous solubility dataset. The goal of studying this particular scenario is to validate whether the derived $p(\mathbf{x}|y)$ can be used as a criterion for the AD while holding information on the closeness to a target y value. From visual inspection of several $p(\mathbf{x}|y)$ and $p(\mathbf{x})$ plots, $p(\mathbf{x}|y)$ seems to fulfill the aforementioned purposes.

In CHAPTER 4, the proposed structure generation system is provided by combining the strategies introduced in CHAPTERS 2 and 3. When sequentially connecting the workflows between inverse analysis of the QSPR/QSAR model and structure generation from constraints, criteria for translating $p(\mathbf{x}|y)$ into constraints are required. Therefore, focusing only on Gaussian functions with good balance between $p(\mathbf{x}|y)$ values and predicted y values is proposed. By designing ligands of thrombin inhibitors, *de novo* chemical structure generation based on inverse QSAR was performed.

5-2 Contributions of the Thesis

There are two major contributions of this thesis. The first contribution is to introduce MCDs as descriptors in inverse analysis and develop efficient structure generation algorithms that make use of them. Previous research related to inverse QSPR/QSAR has focused on a certain type of descriptors, such as connectivity indices or *signatures*[53]. Introducing arbitrary descriptors in structure generation is not possible because inverse mapping between the chemical structures and descriptors is usually unsolvable. Using MCDs with the proposed structure generation algorithm enables a QSPR/QSAR model in inverse analysis to have high predictability, leading to inverse QSPR/QSAR approaches becoming practical.

The second contribution is to propose a methodology that obtains **x** information from a specific *y* value by a probability density function (PDF). So far, all of the methodologies related to inverse QSPR/QSAR have not taken the concept of the AD into account. Moreover, only one linear equation has been used for inversely analyzing a QSPR/QSAR model. The proposed methodology—GMMs/cMLR—overcomes these limitations to a certain extent. Considering the AD in inverse QSPR/QSAR analysis by introducing a PDF prevents inverse QSPR/QSAR analysis from becoming a mere mathematical transformation of a regression equation followed by reconstruction of chemical graphs. This methodology gives a statistical interpretation of inverse QSPR/QSAR analysis.

5-3 Remarks on Inverse QSPR/QSAR

This thesis focuses on solving inverse QSPR/QSAR problems considering the AD. The hypothesis in this thesis is that it is possible to generate exhaustive structures satisfying a specific property or activity based on a QSPR/QSAR model when considering the AD. It is concluded that in practical molecular design, where generating chemical structures with dozens of ring systems and atom fragments needs to be considered, the hypothesis is rejected. In other words, the proposed inverse QSPR/QSAR methodology cannot completely overcome combinatorial explosion. With current computational power, it is possible to treat at most around one billion chemical structures. Assuming that a structure requires one kilobyte of memory to store in SDFile format (benzene requires 700 bytes), for one billion structures, one terabyte of memory is required. Although one terabyte of data is manageable, any arbitrary calculation method cannot be applied to all of the structures because each record needs to be loaded in memory for the calculation. This number is the fundamental limitation of inverse QSPR/QSAR analysis and structure generation. In contrast to my expectation, the number of structures within a specific area in chemical space is so large that an exhaustive method cannot be used, even when the area is strictly limited by introducing many descriptors. Therefore, a stochastic generation strategy and a diversity-oriented generation algorithm have been developed to reduce the number of chemical structures to be generated. In the thrombin case study (Section 4-3-5), exhaustive structure generation was not tractable even though the expected number of chemical structures to be generated was less than one billion. This indicates that further improvement of the structure generation algorithm might be possible because only structures satisfying constraints should ideally be generated without generating any other structures. Consequently, the proposed methodology cannot generate

exhaustive structures exhibiting a specific *y* value based on a QSPR/QSAR model considering the AD. Although the methodology could not generate exhaustive structures in the thrombin case study, it may generate novel chemical structures exhibiting a specific *y* value based on a QSPR/QSAR model after consideration of the model's AD.

5-4 Challenges

Although the proposed system can produce chemical structures satisfying an objective variable value based on a QSPR/QSAR model, there are several challenges remaining to make the system more reliable:

1. The implemented descriptors should be thorough to construct better QSPR/QSAR models.
2. More substructures should be registered on a "taboo" list.
3. Self-contradictory constraints should be determined before performing structure generation.
4. Multiple QSPR/QSAR models should be combined to determine a narrow region in descriptor space that is truly useful.
5. Truly nonlinear regression methodologies should be used in the system to construct models with high predictability.
6. A methodology to determine a threshold for the PDF ($p(\mathbf{x}|y)$) should be developed.

As shown in Section 2-6-3, adequate descriptors can be used in the structure generation system. Nevertheless, implementation of the descriptors without reducing the generation speed is a challenge. The 51 descriptors mentioned in this thesis were carefully implemented. Data structures and algorithms for calculating descriptors were determined based on the descriptors' features. However, there are no general data structures or algorithms that can be applied to the calculation of all descriptors (1).

The above claim can be applied to substructures on the taboo list used during structure generation procedures. Ideally, a sophisticated list, such as pan assay interference compounds [19], and filters developed by the Schneider's group[18] should be implemented to generate only reliable structures from a medicinal chemistry point of view (2). Some pairs of constraints may show contradictory values, meaning that there are no chemical structures that satisfy these values. Early determination of this situation makes the proposed system more rigorous (3). In practical molecular design, only chemical structures satisfying multiple criteria are desired. Therefore, multiple QSPR/QSAR should be incorporated when applying the proposed methodology to real-world problems. Combining PDFs is easy only when these distributions are independent (non-correlation is sufficient for Gaussian functions). Therefore, a methodology is necessary to express a posterior PDF by combining many PDFs (4). To represent a nonlinear relationship between $\mathbf{x}$ and y, GMMs/cMLR is introduced in this thesis. The methodology, which combines MLR models, is a pseudo nonlinear regression methodology. When it encounters a problem where GMMs/cMLR models do not have sufficient predictability, a widely used nonlinear regression methodology should be used in the proposed inverse analysis framework. One promising candidate is the Gaussian

process[162]. A method to sample coordinates based on $p(\mathbf{x}|y)$ from a regression model with a Gaussian process is proposed. Unlike GMMs/cMLR, the methodology with a Gaussian process cannot determine a PDF. Thus, structure generation focusing on $\mathbf{x}$ with the highest $p(\mathbf{x}|y)$ would be a reasonable strategy (5). Finally, the most important part, which affects the proposed chemical structures, is how to determine a proper threshold for $p(\mathbf{x}|y)$. In the case study of thrombin inhibitors, a hyper-rectangle (constraints) was determined based on both $p(\mathbf{x}|y)$ and a predicted pK_i value. Because $p(\mathbf{x}|y)$ contains information for both predicted y values and the data density, determination of the generation constraints should solely depend on $p(\mathbf{x}|y)$. This process is not as straight forward as that shown in Section 3-5-3 because of the sensitivity of $p(\mathbf{x}|y)$ to the difference of the coordinates in high-dimensional space. By making use of dimensionality reduction techniques before constructing GMMs models ($p(\mathbf{x}|y)$), this challenge may be overcome (6).

Furthermore, to make the proposed system more feasible, as other *de novo* structure generators have done[73,163], proposing the synthesizability of the generated structures is necessary. This can be achieved by analyzing the retrosynthesis paths of the generated structures[164] or improving the structure generator, which might take this factor into account.

5-5 Perspectives

This thesis provides a methodology for constructing a structure generation system based on inverse QSPR/QSAR. In terms of practical applications, the proposed system can be applied to any *in silico* molecular design project where a QSPR/QSAR model plays a pivotal role.

In the field of drug discovery, QSAR and QSPR models are still important for the early stage of drug development.[165] They can handle a vast number of (virtual) compounds and select promising chemical structures. QSAR models are useful, particularly when the 3D structures of a target macromolecule are not known. Furthermore, QSAR models can predict activities when target macromolecules are unknown or the mechanism of action of drugs is not understood. As well as considering the activity (affinity) of a target macromolecule, drug-likeness features,[166] such as toxicity and metabolic stability, should be considered to reduce the failure rate in clinical trials, which are very time and cost consuming. Hence, *in silico* drug-likeness models based on QSPR analyses are also used in the early stage of drug discovery. To consider multiple features when designing molecules by the proposed system, which is necessary for practical applications, multiple posterior PDFs can be combined to determine promising (high dense) areas in chemical space. A possible scenario for using an extended proposed system in drug discovery is to propose lead structures *in silico* before performing *in vitro* assays. The extended system can be adopted after determining a target macromolecule, which is an input of the system. In the system, a QSAR model is constructed with experimental data. When the QSAR model has high predictability, inverse QSAR analysis is performed according to the procedure explained in this thesis. Furthermore, highly dense areas of the posterior density can be combined with areas showing drug-like features. These areas become constraints for structure generation in the system. The structure generator Molgilla can generate chemical structures *de novo*. The proposed system can generate millions of chemical structures exhibiting the desired parameters based on QSPR/QSAR models, and these structures can be scrutinized by *in silico* methodologies that

need more computational power[167,168], such as molecular dynamics and quantum mechanics, before performing *in vitro* assays.

The proposed methodology is also applicable in the field of material design focusing on organic compounds, because virtual screening based on QSPR models is also performed in this field[169]. For example, when considering designing organic semiconductors, the proposed system can generate novel chemical structures based on a pre-constructed QSPR model. Molecules for such a purpose usually have a narrow highest occupied molecular orbital–lowest unoccupied molecular orbital gap, meaning that a training dataset contains molecules with conjugated systems. Recently, the Harvard clean energy project has attracted attention[33]. The goal of this project is to propose promising lead compounds as organic semiconductors for photovoltaic applications[170]. In this project, descriptor-based and quantum mechanics-based screening were applied to chemical structures in their virtual library[169]. The structures in the library were generated by combining building blocks in advance. Therefore, in theory, the proposed system based on inverse QSPR can be applied to this project. When considering applying the system to this project, forcing the structure generator to generate conjugated structures seems to be necessary. This requirement insures that the generated structures are inside the universal AD. Material design seems to be more straightforward than drug design because it requires fewer properties. Nevertheless, designing novel materials is important in both industry and academia. As the experimental data and the importance of designing novel molecules increase, a methodology for designing novel chemical structures is required. Structure generation based on inverse QSPR is a promising way to propose chemical structures *de novo*.

Appendix A

How to systematically search for a proper reduced graph corresponding with a ring system is explained in the following pseudo code[97]. The list of templates assigned to each ring is also shown in Figure A-1.

Table A-1 Pseudo codes for constructing a reduced graph matching a ring system by applying templates (recursive procedure for each ring in a spiro ring system). This table was copied from the article by Miyao et al. [97] with permission of Springer.

line	pseudo codes
1	**procedure** *Assign Reduced Graph*(*R:ring system, RG: reduced graph, SI: Current Spiro index*)
2	**if** *SI* is the number of spiro of *R* **then**
3	**if** *R* and *RG* have the same set of automorphism **then**
4	**return** *true*
5	**else**
6	**return** *false*
7	**endif**
8	**else**
9	*RS* = the part of *R* corresponding *SI*
10	*nAps* = the number of access points of *RS*
11	**Case** based on nAps
12	**Case** == 1
13	append a dot template to *RG* and color *RG* based on orbits of the access points of *R*
14	**if** *Assign Reduced Graph*(*R, RG, SI*+1) **then**
15	**return** *true*
16	**endif**
17	**Case** == 2
18	append a line template to *RG* and color *RG*
19	**if** *Assign Reduced Graph*(*R, RG, SI*+1) **then**
20	**return** *true*
21	**endif**
22	Case == 3
23	append a triangle template to *RG* and color *RG*
24	**if** *Assign Reduced Graph*(*R, RG, SI*+1) **then**
25	return *true*
26	**else**
27	withdrew the triangle template from *RG*
28	append a benzene like template *to RG* and color *RG*
29	**if** *Assign Reduced Graph*(*R, RG, SI*+1) **then**

```
30                  return true
31                endif
32              endif
33              Case == 4
34                append a square template to RG and color RG
35                if Assign Reduced Graph(R, RG, SI+1) then
36                  return true
37                else
38                withdrew the triangle template from RG
39                append the template of a hexagonal with two dummy vertices to RG
      and color RG
40              if Assign Reduced Graph(R, RG, SI+1) then
41                return true
42              endif
43              Case == 5
44                append a pentagonal template to RG and color RG
45                if Assign Reduced Graph(R, RG, SI+1) then
46                  return true
47                endif
48              Case == 6
49                append a hexagonal template to RG and color RG
50                if Assign Reduced Graph(R, RG, SI+1) then
51                  return true
52              endif
53              Case == 7
54                append a heptagonal template to RG and color RG
55                if Assign Reduced Graph(R, RG, SI+1) then
56                  return true
57                endif
58              Default
59                return  Not found proper RG
60          EndCase
61      Endif
62  Endprocedure
```

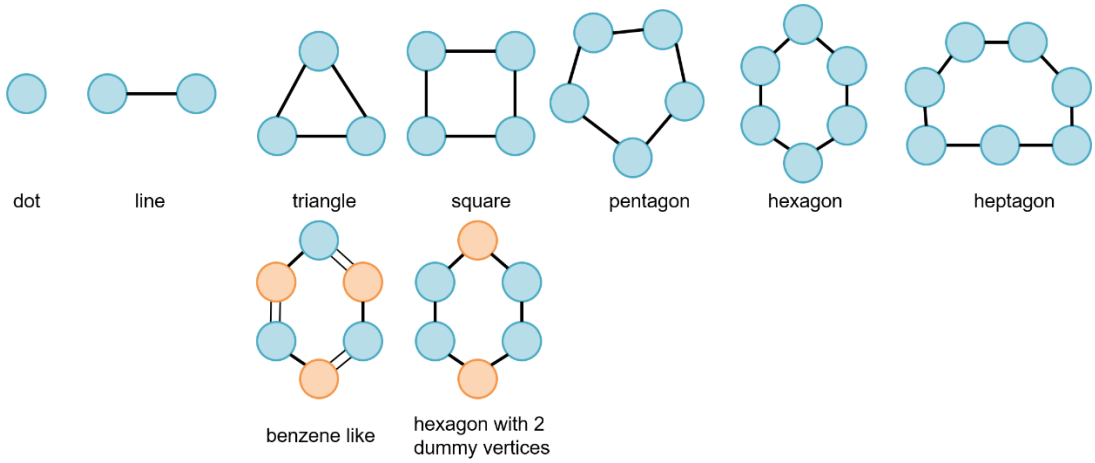


Figure A-1 Set of templates for representing reduced graphs. These templates correspond with those in **Table A-1**. Orange vertices in benzene like and hexagon with 2 dummy vertices are dummy vertices in order to show some topologies. This figure was copied from the article by Miyao et al. [97] with permission of Springer.

Appendix B

Complete list of MCDs used for testing MCDs' description ability. There were 409 MCDs selected by the author.

Table B1 MCDs categorized by the author in DRAGON 5 (0D, 1D, and 2D)

ID	Descriptor	Description
1	MW	molecular weight
2	Sv	sum of atomic van der Waals volumes (scaled on Carbon atom)
3	Se	sum of atomic Sanderson electronegativities (scaled on Carbon atom)
4	Sp	sum of atomic polarizabilities (scaled on Carbon atom)
5	nAT	number of atoms
6	nSK	number of non-H atoms
7	nBT	number of bonds
8	nBO	number of non-H bonds
9	nBM	number of multiple bonds
10	SCBO	sum of conventional bond orders (H-depleted)
11	nCIC	number of rings
12	nCIR	number of circuits
13	RBN	number of rotatable bonds
14	nDB	number of double bonds
15	nTB	number of triple bonds
16	nAB	number of aromatic bonds
17	nH	number of Hydrogen atoms
18	nC	number of Carbon atoms
19	nN	number of Nitrogen atoms
20	nO	number of Oxygen atoms
21	nS	number of Sulfur atoms
22	nF	number of Fluorine atoms
23	nCL	number of Chlorine atoms
24	nBR	number of Bromine atoms
25	nHM	number of heavy atoms
26	nX	number of halogen atoms
27	nR03	number of 3-membered rings
28	nR04	number of 4-membered rings
29	nR05	number of 5-membered rings
30	nR06	number of 6-membered rings
31	nR07	number of 7-membered rings
32	nR08	number of 8-membered rings
33	nR09	number of 9-membered rings

34	nR10	number of 10-membered rings
35	nR11	number of 11-membered rings
36	nR12	number of 12-membered rings
37	nBnz	number of benzene-like rings
38	ZM1	first Zagreb index M1
39	ZM1V	first Zagreb index by valence vertex degrees
40	ZM2	second Zagreb index M2
41	ZM2V	second Zagreb index by valence vertex degrees
42	SNar	Narumi simple topological index (log)
43	Xt	Total structure connectivity index
44	Dz	Pogliani index
45	Ram	ramification index
46	Pol	polarity number
47	LPRS	log of product of row sums (PRS)
48	SMTI	Schultz Molecular Topological Index (MTI)
49	SMTIV	Schultz MTI by valence vertex degrees
50	GMTI	Gutman Molecular Topological Index
51	GMTIV	Gutman MTI by valence vertex degrees
52	W	Wiener W index
53	Har	Harary H index
54	Har2	square reciprocal distance sum index
55	QW	quasi-Wiener index (Kirchhoff number)
56	HyDp	hyper-distance-path index
57	RHyDp	reciprocal hyper-distance-path index
58	w	Wiener W index
59	ww	hyper-detour index
60	Rww	reciprocal hyper-detour index
61	Wap	all-path Wiener index
62	WhetZ	Wiener-type index from Z weighted distance matrix (Barysz matrix)
63	Whetm	Wiener-type index from mass weighted distance matrix
64	Whetv	Wiener-type index from van der Waals weighted distance matrix
65	Whete	Wiener-type index from electronegativity weighted distance matrix
66	Whetp	Wiener-type index from polarizability weighted distance matrix
67	CSI	eccentric connectivity index
68	ECC	eccentricity
69	UNIP	unipolarity
70	BAC	Balaban centric index
71	T(N..N)	sum of topological distances between N..N
72	T(N..O)	sum of topological distances between N..O
73	T(N..S)	sum of topological distances between N..S
74	T(N..F)	sum of topological distances between N..F
75	T(N..Cl)	sum of topological distances between N..Cl

76	T(N..Br)	sum of topological distances between N..Br
77	T(O..O)	sum of topological distances between O..O
78	T(O..S)	sum of topological distances between O..S
79	T(O..F)	sum of topological distances between O..F
80	T(O..Cl)	sum of topological distances between O..Cl
81	T(O..Br)	sum of topological distances between O..Br
82	T(S..S)	sum of topological distances between S..S
83	T(S..F)	sum of topological distances between S..F
84	T(S..Cl)	sum of topological distances between S..Cl
85	T(S..Br)	sum of topological distances between S..Br
86	T(F..F)	sum of topological distances between F..F
87	T(F..Cl)	sum of topological distances between F..Cl
88	T(Cl..Cl)	sum of topological distances between Cl..Cl
89	T(Cl..Br)	sum of topological distances between Cl..Br
90	MWC01	molecular walk count of order 01 (number of non-H bonds, nBO)
91	MWC02	molecular walk count of order 02
92	MWC03	molecular walk count of order 03
93	MWC04	molecular walk count of order 04
94	MWC05	molecular walk count of order 05
95	MWC06	molecular walk count of order 06
96	MWC07	molecular walk count of order 07
97	MWC08	molecular walk count of order 08
98	MWC09	molecular walk count of order 09
99	MWC10	molecular walk count of order 10
100	TWC	total walk count
101	SRW01	self-returning walk count of order 01 (number of non-H atoms, nSK)
102	SRW02	self-returning walk count of order 02 (twice the number of non-H bonds)
103	SRW03	self-returning walk count of order 03
104	SRW04	self-returning walk count of order 04
105	SRW05	self-returning walk count of order 05
106	SRW06	self-returning walk count of order 06
107	SRW07	self-returning walk count of order 07
108	SRW08	self-returning walk count of order 08
109	SRW09	self-returning walk count of order 09
110	SRW10	self-returning walk count of order 10
111	MPC01	molecular path count of order 01 (number of non-H bonds, nBO)
112	MPC02	molecular path count of order 02 (Gordon-Scantlebury index)
113	MPC03	molecular path count of order 03
114	MPC04	molecular path count of order 04
115	MPC05	molecular path count of order 05
116	MPC06	molecular path count of order 06
117	MPC07	molecular path count of order 07

118	MPC08	molecular path count of order 08
119	MPC09	molecular path count of order 09
120	MPC10	molecular path count of order 10
121	piPC01	molecular multiple path count of order 01 (sum of conventional bond orders, SCBO)
122	piPC02	molecular multiple path count of order 02
123	piPC03	molecular multiple path count of order 03
124	piPC04	molecular multiple path count of order 04
125	piPC05	molecular multiple path count of order 05
126	piPC06	molecular multiple path count of order 06
127	piPC07	molecular multiple path count of order 07
128	piPC08	molecular multiple path count of order 08
129	piPC09	molecular multiple path count of order 09
130	piPC10	molecular multiple path count of order 10
131	TPC	total path count
132	piID	conventional bond-order ID number
133	CID	Randic ID number
134	BID	Balaban ID number
135	X0	connectivity index chi-0
136	X1	connectivity index chi-1 (Randic connectivity index)
137	X2	connectivity index chi-2
138	X3	connectivity index chi-3
139	X4	connectivity index chi-4
140	X5	connectivity index chi-5
141	X0v	valence connectivity index chi-0
142	X1v	valence connectivity index chi-1
143	X2v	valence connectivity index chi-2
144	X3v	valence connectivity index chi-3
145	X4v	valence connectivity index chi-4
146	X5v	valence connectivity index chi-5
147	XMOD	modified Randic connectivity index
148	RDCHI	reciprocal distance Randic-type index
149	RDSQ	reciprocal distance squared Randic-type index
150	ISIZ	information index on molecular size
151	TIC0	total information content index (neighborhood symmetry of 0-order)
152	TIC1	total information content index (neighborhood symmetry of 1-order)
153	TIC2	total information content index (neighborhood symmetry of 2-order)
154	TIC3	total information content index (neighborhood symmetry of 3-order)
155	TIC4	total information content index (neighborhood symmetry of 4-order)
156	TIC5	total information content index (neighborhood symmetry of 5-order)
157	ATS1m	Broto-Moreau autocorrelation of a topological structure - lag 1 / weighted by atomic masses

158	ATS2m	Broto-Moreau autocorrelation of a topological structure - lag 2 / weighted by atomic masses
159	ATS3m	Broto-Moreau autocorrelation of a topological structure - lag 3 / weighted by atomic masses
160	ATS4m	Broto-Moreau autocorrelation of a topological structure - lag 4 / weighted by atomic masses
161	ATS5m	Broto-Moreau autocorrelation of a topological structure - lag 5 / weighted by atomic masses
162	ATS6m	Broto-Moreau autocorrelation of a topological structure - lag 6 / weighted by atomic masses
163	ATS7m	Broto-Moreau autocorrelation of a topological structure - lag 7 / weighted by atomic masses
164	ATS8m	Broto-Moreau autocorrelation of a topological structure - lag 8 / weighted by atomic masses
165	ATS1v	Broto-Moreau autocorrelation of a topological structure - lag 1 / weighted by atomic van der Waals volumes
166	ATS2v	Broto-Moreau autocorrelation of a topological structure - lag 2 / weighted by atomic van der Waals volumes
167	ATS3v	Broto-Moreau autocorrelation of a topological structure - lag 3 / weighted by atomic van der Waals volumes
168	ATS4v	Broto-Moreau autocorrelation of a topological structure - lag 4 / weighted by atomic van der Waals volumes
169	ATS5v	Broto-Moreau autocorrelation of a topological structure - lag 5 / weighted by atomic van der Waals volumes
170	ATS6v	Broto-Moreau autocorrelation of a topological structure - lag 6 / weighted by atomic van der Waals volumes
171	ATS7v	Broto-Moreau autocorrelation of a topological structure - lag 7 / weighted by atomic van der Waals volumes
172	ATS8v	Broto-Moreau autocorrelation of a topological structure - lag 8 / weighted by atomic van der Waals volumes
173	ATS1e	Broto-Moreau autocorrelation of a topological structure - lag 1 / weighted by atomic Sanderson electronegativities
174	ATS2e	Broto-Moreau autocorrelation of a topological structure - lag 2 / weighted by atomic Sanderson electronegativities
175	ATS3e	Broto-Moreau autocorrelation of a topological structure - lag 3 / weighted by atomic Sanderson electronegativities
176	ATS4e	Broto-Moreau autocorrelation of a topological structure - lag 4 / weighted by atomic Sanderson electronegativities
177	ATS5e	Broto-Moreau autocorrelation of a topological structure - lag 5 / weighted by atomic Sanderson electronegativities
178	ATS6e	Broto-Moreau autocorrelation of a topological structure - lag 6 / weighted by atomic Sanderson electronegativities
179	ATS7e	Broto-Moreau autocorrelation of a topological structure - lag 7 / weighted by atomic Sanderson electronegativities
180	ATS8e	Broto-Moreau autocorrelation of a topological structure - lag 8 / weighted by atomic Sanderson electronegativities

181	ATS1p	Broto-Moreau autocorrelation of a topological structure - lag 1 / weighted by atomic polarizabilities
182	ATS2p	Broto-Moreau autocorrelation of a topological structure - lag 2 / weighted by atomic polarizabilities
183	ATS3p	Broto-Moreau autocorrelation of a topological structure - lag 3 / weighted by atomic polarizabilities
184	ATS4p	Broto-Moreau autocorrelation of a topological structure - lag 4 / weighted by atomic polarizabilities
185	ATS5p	Broto-Moreau autocorrelation of a topological structure - lag 5 / weighted by atomic polarizabilities
186	ATS6p	Broto-Moreau autocorrelation of a topological structure - lag 6 / weighted by atomic polarizabilities
187	ATS7p	Broto-Moreau autocorrelation of a topological structure - lag 7 / weighted by atomic polarizabilities
188	ATS8p	Broto-Moreau autocorrelation of a topological structure - lag 8 / weighted by atomic polarizabilities
189	EPS0	edge connectivity index of order 0
190	EPS1	edge connectivity index of order 1
191	ESpm02u	Spectral moment 02 from edge adj. matrix
192	ESpm03u	Spectral moment 03 from edge adj. matrix
193	ESpm04u	Spectral moment 04 from edge adj. matrix
194	ESpm05u	Spectral moment 05 from edge adj. matrix
195	ESpm06u	Spectral moment 06 from edge adj. matrix
196	ESpm07u	Spectral moment 07 from edge adj. matrix
197	ESpm08u	Spectral moment 08 from edge adj. matrix
198	ESpm09u	Spectral moment 09 from edge adj. matrix
199	ESpm10u	Spectral moment 10 from edge adj. matrix
200	ESpm11u	Spectral moment 11 from edge adj. matrix
201	ESpm12u	Spectral moment 12 from edge adj. matrix
202	ESpm13u	Spectral moment 13 from edge adj. matrix
203	ESpm14u	Spectral moment 14 from edge adj. matrix
204	ESpm15u	Spectral moment 15 from edge adj. matrix
205	ESpm01x	Spectral moment 01 from edge adj. matrix weighted by edge degrees
206	ESpm02x	Spectral moment 02 from edge adj. matrix weighted by edge degrees
207	ESpm03x	Spectral moment 03 from edge adj. matrix weighted by edge degrees
208	ESpm04x	Spectral moment 04 from edge adj. matrix weighted by edge degrees
209	ESpm05x	Spectral moment 05 from edge adj. matrix weighted by edge degrees
210	ESpm06x	Spectral moment 06 from edge adj. matrix weighted by edge degrees
211	ESpm07x	Spectral moment 07 from edge adj. matrix weighted by edge degrees
212	ESpm08x	Spectral moment 08 from edge adj. matrix weighted by edge degrees
213	ESpm09x	Spectral moment 09 from edge adj. matrix weighted by edge degrees
214	ESpm10x	Spectral moment 10 from edge adj. matrix weighted by edge degrees
215	ESpm11x	Spectral moment 11 from edge adj. matrix weighted by edge degrees
216	ESpm12x	Spectral moment 12 from edge adj. matrix weighted by edge degrees

217	ESpm13x	Spectral moment 13 from edge adj. matrix weighted by edge degrees
218	ESpm14x	Spectral moment 14 from edge adj. matrix weighted by edge degrees
219	ESpm15x	Spectral moment 15 from edge adj. matrix weighted by edge degrees
220	ESpm01d	Spectral moment 01 from edge adj. matrix weighted by dipole moments
221	ESpm02d	Spectral moment 02 from edge adj. matrix weighted by dipole moments
222	ESpm03d	Spectral moment 03 from edge adj. matrix weighted by dipole moments
223	ESpm04d	Spectral moment 04 from edge adj. matrix weighted by dipole moments
224	ESpm05d	Spectral moment 05 from edge adj. matrix weighted by dipole moments
225	ESpm06d	Spectral moment 06 from edge adj. matrix weighted by dipole moments
226	ESpm07d	Spectral moment 07 from edge adj. matrix weighted by dipole moments
227	ESpm08d	Spectral moment 08 from edge adj. matrix weighted by dipole moments
228	ESpm09d	Spectral moment 09 from edge adj. matrix weighted by dipole moments
229	ESpm10d	Spectral moment 10 from edge adj. matrix weighted by dipole moments
230	ESpm11d	Spectral moment 11 from edge adj. matrix weighted by dipole moments
231	ESpm12d	Spectral moment 12 from edge adj. matrix weighted by dipole moments
232	ESpm13d	Spectral moment 13 from edge adj. matrix weighted by dipole moments
233	ESpm14d	Spectral moment 14 from edge adj. matrix weighted by dipole moments
234	ESpm15d	Spectral moment 15 from edge adj. matrix weighted by dipole moments
235	ESpm01r	Spectral moment 01 from edge adj. matrix weighted by resonance integrals
236	ESpm02r	Spectral moment 02 from edge adj. matrix weighted by resonance integrals
237	ESpm03r	Spectral moment 03 from edge adj. matrix weighted by resonance integrals
238	ESpm04r	Spectral moment 04 from edge adj. matrix weighted by resonance integrals
239	ESpm05r	Spectral moment 05 from edge adj. matrix weighted by resonance integrals
240	ESpm06r	Spectral moment 06 from edge adj. matrix weighted by resonance integrals
241	ESpm07r	Spectral moment 07 from edge adj. matrix weighted by resonance integrals
242	ESpm08r	Spectral moment 08 from edge adj. matrix weighted by resonance integrals
243	ESpm09r	Spectral moment 09 from edge adj. matrix weighted by resonance integrals
244	ESpm10r	Spectral moment 10 from edge adj. matrix weighted by resonance integrals
245	ESpm11r	Spectral moment 11 from edge adj. matrix weighted by resonance integrals
246	ESpm12r	Spectral moment 12 from edge adj. matrix weighted by resonance integrals
247	ESpm13r	Spectral moment 13 from edge adj. matrix weighted by resonance integrals

248	ESpm14r	Spectral moment 14 from edge adj. matrix weighted by resonance integrals
249	ESpm15r	Spectral moment 15 from edge adj. matrix weighted by resonance integrals
250	nCp	number of terminal primary C(sp3)
251	nCs	number of total secondary C(sp3)
252	nCt	number of total tertiary C(sp3)
253	nCq	number of total quaternary C(sp3)
254	nCrs	number of ring secondary C(sp3)
255	nCrt	number of ring tertiary C(sp3)
256	nCrq	number of ring quaternary C(sp3)
257	nCar	number of aromatic C(sp2)
258	nCbH	number of unsubstituted benzene C(sp2)
259	nCb-	number of substituted benzene C(sp2)
260	nCconj	number of non-aromatic conjugated C(sp2)
261	nR=Cp	number of terminal primary C(sp2)
262	nR=Cs	number of aliphatic secondary C(sp2)
263	nR=Ct	number of aliphatic tertiary C(sp2)
264	nRCOOH	number of carboxylic acids (aliphatic)
265	nRCOOR	number of esters (aliphatic)
266	nArCOOR	number of esters (aromatic)
267	nRCONH2	number of primary amides (aliphatic)
268	nArCONH2	number of primary amides (aromatic)
269	nRCONHR	number of secondary amides (aliphatic)
270	nArCONHR	number of secondary amides (aromatic)
271	nRCONR2	number of tertiary amides (aliphatic)
272	nArCONR2	number of tertiary amides (aromatic)
273	nROCON	number of (thio-) carbamates (aliphatic)
274	nArOCON	number of (thio-) carbamates (aromatic)
275	nRCO	number of ketones (aliphatic)
276	nArCO	number of ketones (aromatic)
277	nCONN	number of urea (-thio) derivatives
278	nN=C-N<	number of amidine derivatives
279	nC(=N)N2	number of guanidine derivatives
280	nRCN	number of nitriles (aliphatic)
281	nArCN	number of nitriles (aromatic)
282	nArCNO	number of oximes (aromatic)
283	nRNH2	number of primary amines (aliphatic)
284	nArNH2	number of primary amines (aromatic)
285	nRNHR	number of secondary amines (aliphatic)
286	nArNHR	number of secondary amines (aromatic)
287	nRNR2	number of tertiary amines (aliphatic)
288	nArNR2	number of tertiary amines (aromatic)

289	nN-N	number of N hydrazines
290	nRCN	number of nitriles (aliphatic)
291	nArCN	number of nitriles (aromatic)
292	nN+	number of positively charged N
293	nArNO2	number of nitro groups (aromatic)
294	nN(CO)2	number of imides (-thio)
295	nC=N-N<	number of hydrazones
296	nROH	number of hydroxyl groups
297	nArOH	number of aromatic hydroxyls
298	nOHp	number of primary alcohols
299	nOHs	number of secondary alcohols
300	nOHt	number of tertiary alcohols
301	nROR	number of ethers (aliphatic)
302	nArOR	number of ethers (aromatic)
303	nRSR	number of sulfides
304	nS(=O)2	number of sulfones
305	nSO3	number of sulfonates (thio-/dithio-)
306	nSO2N	number of sulfonamides (thio-/dithio-)
307	nCH2RX	number of CH2RX
308	nCHR2X	number of CHR2X
309	nCHRX2	number of CHRX2
310	nCR2X2	number of CR2X2
311	nCRX3	number of CRX3
312	nArX	number of X on aromatic ring
313	nCXr	number of X on ring C(sp3)
314	nCXr=	number of X on ring C(sp2)
315	nCconjX	number of X on exo-conjugated C
316	nPyrrolidines	number of Pyrrolidines
317	nOxolanes	number of Oxolanes
318	nPyrroles	number of Pyrroles
319	nPyrazoles	number of Pyrazoles
320	nImidazoles	number of Imidazoles
321	nFuranes	number of Furanes
322	nThiophenes	number of Thiophenes
323	nOxazoles	number of Oxazoles
324	nIsoxazoles	number of Isoxazoles
325	nThiazoles	number of Thiazoles
326	nPyridines	number of Pyridines
327	nPyridazines	number of Pyridazines
328	nPyrimidines	number of Pyrimidines
329	nPyrazines	number of Pyrazines
330	nHDon	number of donor atoms for H-bonds (N and O)

331	nHAcc	number of acceptor atoms for H-bonds (N,O,F)
332	C-001	CH3R / CH4
333	C-002	CH2R2
334	C-003	CHR3
335	C-004	CR4
336	C-005	CH3X
337	C-006	CH2RX
338	C-007	CH2X2
339	C-008	CHR2X
340	C-009	CHRX2
341	C-010	CHX3
342	C-011	CR3X
343	C-012	CR2X2
344	C-013	CRX3
345	C-014	CX4
346	C-015	=CH2
347	C-016	=CHR
348	C-017	=CR2
349	C-018	=CHX
350	C-019	=CRX
351	C-020	=CX2
352	C-024	R--CH--R
353	C-025	R--CR--R
354	C-026	R--CX--R
355	C-027	R--CH--X
356	C-028	R--CR--X
357	C-029	R--CX--X
358	C-031	X--CR--X
359	C-032	X--CX--X
360	C-033	R--CH..X
361	C-034	R--CR..X
362	C-035	R--CX..X
363	C-038	Al-C(=X)-Al
364	C-039	Ar-C(=X)-R
365	C-040	R-C(=X)-X / R-C#X / X=C=X
366	C-041	X-C(=X)-X
367	C-042	X--CH..X
368	C-043	X--CR..X
369	C-044	X--CX..X
370	H-046	H attached to C0(sp3) no X attached to next C
371	H-047	H attached to C1(sp3)/C0(sp2)
372	H-048	H attached to C2(sp3)/C1(sp2)/C0(sp)

373	H-049	H attached to C3(sp3)/C2(sp2)/C3(sp2)/C3(sp)
374	H-050	H attached to heteroatom
375	H-051	H attached to alpha-C
376	H-052	H attached to C0(sp3) with 1X attached to next C
377	H-053	H attached to C0(sp3) with 2X attached to next C
378	O-056	alcohol
379	O-057	phenol / enol / carboxyl OH
380	O-058	=O
381	O-059	Al-O-Al
382	O-060	Al-O-Ar / Ar-O-Ar / R..O..R / R-O-C=X
383	O-061	O--
384	N-066	Al-NH2
385	N-067	Al2-NH
386	N-068	Al3-N
387	N-069	Ar-NH2 / X-NH2
388	N-070	Ar-NH-Al
389	N-071	Ar-NAl2
390	N-072	RCO-N< / >N-X=X
391	N-073	Ar2NH / Ar3N / Ar2N-Al / R..N..R
392	N-074	R#N / R=N-
393	N-075	R--N--R / R--N--X
394	N-076	Ar-NO2 / R--N(--R)--O / RO-NO
395	N-079	N+ (positively charged)
396	F-081	F attached to C1(sp3)
397	F-082	F attached to C2(sp3)
398	F-083	F attached to C3(sp3)
399	F-084	F attached to C1(sp2)
400	F-085	F attached to C2(sp2)-C4(sp2)/C1(sp)/C4(sp3)/X
401	Cl-086	Cl attached to C1(sp3)
402	Cl-089	Cl attached to C1(sp2)
403	Cl-090	Cl attached to C2(sp2)-C4(sp2)/C1(sp)/C4(sp3)/X
404	Br-094	Br attached to C1(sp2)
405	Br-095	Br attached to C2(sp2)-C4(sp2)/C1(sp)/C4(sp3)/X
406	S-107	R2S / RS-SR
407	S-108	R=S
408	S-110	R-SO2-R
409	Si-111	>Si<

Appendix C

Calculating STDPs between hydrogen bond acceptor (A) and lipophilic point (L) is described in this section. The example as well as the pseudo code is extracted from the paper.

Since the important part is to calculate R(GF) without spending much calculation resource, only this part is highlighted here. The pseudo code is shown on Table 6 and 7. The function *differenceAL* takes four arguments: growing structure (X), fragment (F) that is to be attached to X, access point (APt) of F at which F is connected to X, and distance matrix (D). After the distance matrix is updated, STDP is updated among fragments, followed by the value update between the access point that becomes saturated and fragments. It should be noted that the word *target* in Table C-1 means the point at which F is attached to, such as *target* access point and *target* fragment in X.

Table C-1 Pseudo codes for calculating the difference between STDP(G-F) and STDP (G) in case of A and L.

line	pseudo code
1	**procedure** *differenceAL*(*X*: reduced graph, *F*: fragment, *APt*: access point, *D*: distance matrix)
2	D_{new} = *update distance matrix*(*D*)
3	access point APt_{target} of fragment F_{target} in *X* is the point to which *AP* of *F* is connected
4	**if** APt_{target} is saturated **then**
5	PPP_{target} = determine pharmacophoric type at APt_{target}
6	$AL_{inside,target}$ = *updateAL_inside_fragment*(F_{target}, APt_{target}, PPP_{target})
7	$AL_{outside\_target}$ = *updateAL_outside_fragment*(F_{target}, APt_{target}, PPP_{target}, *X*, *D*)
8	**Endif**
9	**if** *AP* is saturated **then**
10	*P* = determine pharmacophoric type at *APt*
11	$AL_{inside,F}$ = *updateAL_inside_fragment*(*F*, *APt*, *P*)
12	**Endif**
13	*nA* is the number of A in *F*
14	*nL* is the number of L in *F*
15	*sA* is the sum of distances of As in *F* to *APt*
16	*sL* is the sum of distances of Ls in *F* to *APt*
17	$STDP_{update}$ = 0
18	**for** each fragment *tF* in *X* **do**
19	nA_{tF} is the number of A in *tF*
20	nL_{tF} is the number of L in *tF*
21	AP_{tF} is the closest access point to *F* in *tF*
22	sA_{tF} is the sum of distances of As in tF to APt_{tF}
23	sL_{tF} is the sum of distances of Ls in tF to APt_{tF}
24	$STDP_{update}$ += (nA_{tF}·D_{new}(*tF*, *F*) + sA_{tF})·*nL* + (nL_{tF}·D_{new}(*tF*, *F*) + sL_{tF})·*nA* + *sA*·nL_{tF}+ *sL*·nA_{tF}
25	**Endfor**
26	**return** $AL_{inside,target}$ + $AL_{outside\_target}$ + $AL_{inside,F}$ + $STDP_{update}$
27	**Endprocedure**

Table C-2 Algorithm for updating A and L (hydrogen bond acceptors and lipophilicity) inside a fragment when the saturated access point emerges

line	pseudo code
1	**procedure** *updateAL_inside_fragment*(*F: fragment , APt: access point, P*: pharmacophore type of *APt*)
2	$AL_{inside} = 0$
3	**if** *P* is A **then**
4	AL_{inside} += sum of distances between *APt* and every other vertex having L in *F*
5	**Endif**
6	**if** *P* is L **then**
7	AL_{inside} += sum of distances between *APt* and every other vertex having A in *F*
8	**Endif**
9	**return** AL_{inside}
10	**Endprocedure**

Appendix D

Chemical structures containing unstable and reactive substructures should not be generated in structure generation. Furthermore, structures containing hard to synthesize should be eliminated. Table D-1 is the compiled taboo list and Figure D-1 shows corresponding example structures.

Table D-1 Taboo list implemented in Molgilla

Substructure	Definition and explanation
X_X	Direct connection between hetero atoms, which is usually reactive.
__C__	Allene-like motif, which have higher reactivity than alkene.
Rs_Rs	Direct connection between ring systems, making a steric complex structure.
MR	Multiple building blocks are connected at an access point in a ring system, making a steric complex structure.
TMC	Carbon atom located at the end of chemical bond with a double bond, which lacks pharmacophoric effect
C__N	Imine-like motif, which is reactive.
C__OX	Acyl halide-like structure, which is reactive. X is halogen.
CH2X	Alkyl halide-like structure, which is reactive. X is halogen.
CH__O	Aldehyde like motif (CH(=O)), which is reactive
AlC__OAl	Aliphatic ketone, which is reactive. Al is a non-aromatic carbon.
Csp3	Carbon atom with 4 atoms (sp3) except hydrogen atoms, which makes synthesis difficult.
CHsp3_R3	Carbon atom with 3 ring systems and one hydrogen atoms, which means both steric complicated and higher chance of being a chirality center.
Nsp3_R3	Nitrogen atom with 3 ring systems, which is steric complicated
3-membered O	Epoxides-like motif, which is reactive (implemented in *DecomposeRingFragment* module).
3-membered N	Aziridine-like motif, which is reactive (implemented in *DecomposeRingFragment* module).

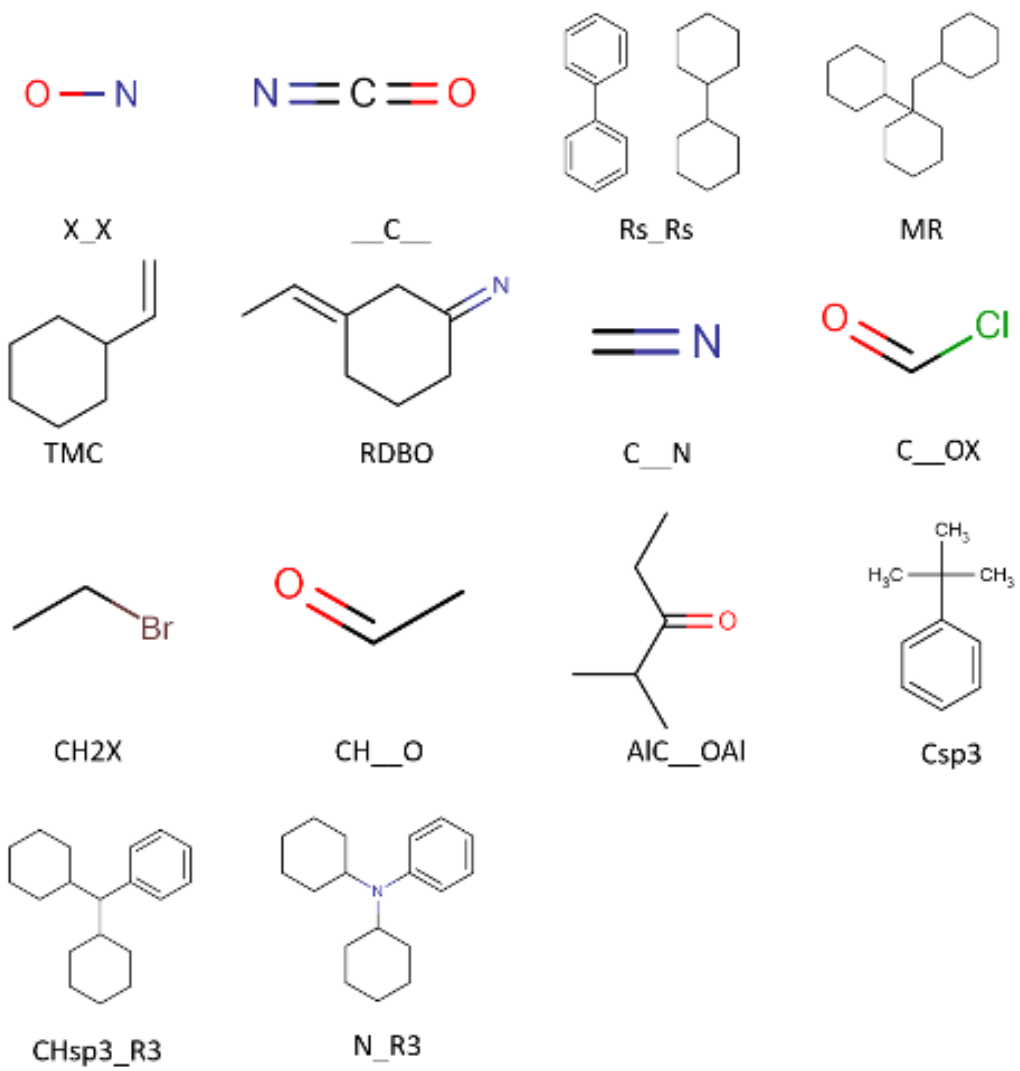


Figure D-1 Example structures containing substructures on the taboo list (Table D-1)

Appendix E

51 MCDs (including 21 STDPs) are currently implemented in Molgilla. They are listed in the Table E1.

Table E1 MCD list: L means lipophilic point, A means hydrogen bond acceptor point, D means hydrogen bond donor point, P means positively charged point, N means negatively charged point, and R means aromatic rings.

No.	MCD	Definition
1	CIC	Number of rings (SSSR)
2	R05	Number of 5-membered rings (SSSR)
3	aR	Number of aromatic rings (SSSR)
4	ZM1V	1st valence Zagrev index (summation of squared valence electrons except for hydrogen atom connections).
5	nHeavyAtom	Number of heavy atoms.
6	nBT	Number of bonds.
7	nBM	Number of weighted multiple bonds
8	nBR	Number of rotatable bonds. Rotatable bonds are single bonds that are not at an edge of a molecular graph (without hydrogen atoms), also do not participate in ring formation.
9	MW	Molecular weight.
10	H050	Number of hydrogens attached to hetero atoms.
11	nHBDLipin	Number of hydrogen bond donors defined by Lipinski [14](number of OHs and NHs).
12	nHAccLipin	Number of Hydrogen bond acceptors defined by Lipinski (number of Os and Ns).
13	nCH2R2	Number of substructures defined by the SMARTS query: [CH2]([C,c])[C,c].
14	nCH3R	Number of substructures CH3 fragments connected to a carbon atom.
15	nCH3X	Number of substructures CH3 fragments connected to a hetero atom.
16	nOH	Number of substructures defined by the SMARTS query: C[OH].
17	nO=	Number of Os with double bond.
18	nArNR2	Number of aromatic amines defined by the SMARTS query: aN(C)C.
19	nArCN	Number of aromatic cyanos defined by the SMARTS query: aC#N.
20	nArCO	Number of aromatic ketones defined by the SMARTS query: [a]C=O.
21	nRCOOH	Number of carboxylic acids defined by the SMARTS query: C(=O)[OH].

22	X1	Randic connectivity index.
23	TPSA	Topological polar surface area based on the atomic contribution method by Ertl et al.[171]
24	LL	Sum of topological distances between lipophilic points.
25	LA	Sum of topological distances between lipophilic and hydrogen bond acceptor points.
26	LD	Sum of topological distances between lipophilic and hydrogen bond donor points.
27	LN	Sum of topological distances between lipophilic and negatively charged points.
28	LP	Sum of topological distances between lipophilic and positively charged points.
30	LR	Sum of topological distances between lipophilic points and aromatic rings.
31	AA	Sum of topological distances between hydrogen bond acceptor points.
32	AD	Sum of topological distances between hydrogen bond acceptor and hydrogen bond donor points.
33	AN	Sum of topological distances between hydrogen bond acceptor and negatively charged points.
34	AP	Sum of topological distances between hydrogen bond acceptor and positively charged points.
35	AR	Sum of topological distances between hydrogen bond acceptor points and aromatic rings
36	DD	Sum of topological distances between hydrogen bond donor points.
37	DN	Sum of topological distances between hydrogen bond donor and negatively charged points.
38	DP	Sum of topological distances between hydrogen bond donor and positively charged points.
39	DR	Sum of topological distances between hydrogen bond donor points and aromatic rings
40	NN	Sum of topological distances between negatively charged points.
41	NP	Sum of topological distances between negatively charged and positively charged points.
42	NR	Sum of topological distances between negatively charged points and aromatic rings
43	PP	Sum of topological distances between positively charged points.
44	PR	Sum of topological distances between positively charged points and aromatic rings.
45	RR	Sum of topological distances between aromatic rings.
46	nL	Number of lipophilic points
47	nA	Number of hydrogen bond acceptor points
48	nD	Number of hydrogen bond donor points

49	nN	Number of negatively charged points
50	nP	Number of positively charged points
51	naR	Number of aromatic rings

Appendix F

In this chapter, supporting information related to the analysis in CHAPTER 4are shown. Table F-1 shows the coordinates of the centers of Gaussians for the prior distribution used in analysis of thrombin. Figure F-1 shows GTM maps with grayscale background representing the density of posterior distribution of **x** given y.

Table F-1 Centers (means) of Gaussians (8 Gaussians) of prior distribution.

	C1	C2	C3	C4	C5	C6	C7	C8
CIC	2.5	4.2	3.1	3.9	3.9	4.3	4.3	3.5
R05	0.4	1.5	0.7	0.3	0.9	1.4	0.8	0.6
aR	1.6	2.7	1.1	3	2.9	3.6	1.8	3.2
ZMIV	382.2	549.8	466.3	560.6	440	595.3	607.6	405.5
nBM	15.1	21.2	14	23.9	20.5	21.9	20.1	21.1
nHAccLipi n	6.9	9.9	10	9.4	7	8.1	13	6.5
nCH2R2	3	4.7	9.2	1.6	2.5	1.4	7.5	0.8
nCHR3	0.3	0.5	0.7	0.2	0.7	0.0	1.2	0.0
nCH3R	0.9	0.4	0.2	0.7	1.2	0.5	0.9	0.7
nCH3X	0.2	0.4	0.1	0.6	0.2	0.8	0.0	0.1
nOH	0.1	0.0	0.6	0.5	0.2	0.1	0.2	0.0
nO2	1.5	3.2	3.4	2.1	1.5	1.6	3.8	1
ArNR2	0.0	0.0	0.0	0.9	0.2	0.2	0.0	0.0
nArCO	0.2	0.7	0.1	0.5	0.3	0.9	0.4	0.3
TPSA	113.4	145.2	156.8	143.6	106.1	101	193	109
LL	75.5	115.1	327	26.1	77	41.1	428.8	11.2
LD	157.8	155.5	411.7	103.8	126.8	48.1	556.9	50.8
LP	76.4	73.7	158.3	41.7	46	6.5	200	19.3
AA	54.9	136.2	102.5	157.3	54.9	255.7	267.4	32.8
AP	52.8	83.2	83.2	92.4	37	24.9	179.6	34.5
AN	0.0	0.0	16.2	22	3.6	0.0	7.5	0.0
DD	36.3	33	79.6	60.8	33.3	11.2	205.5	44.8
RL	44.2	116.9	88.2	54.9	86.9	67	126.3	31.3
RA	32.4	114.6	41.3	115.8	65.7	168.2	148.3	50.8
RD	37.9	80	38	82.6	56.1	39	108.2	67.8
RP	17.5	36.9	15.2	32.3	18	8	38.7	25.1
RR	3.4	19.4	1	18	12.5	23.6	20.5	15.7

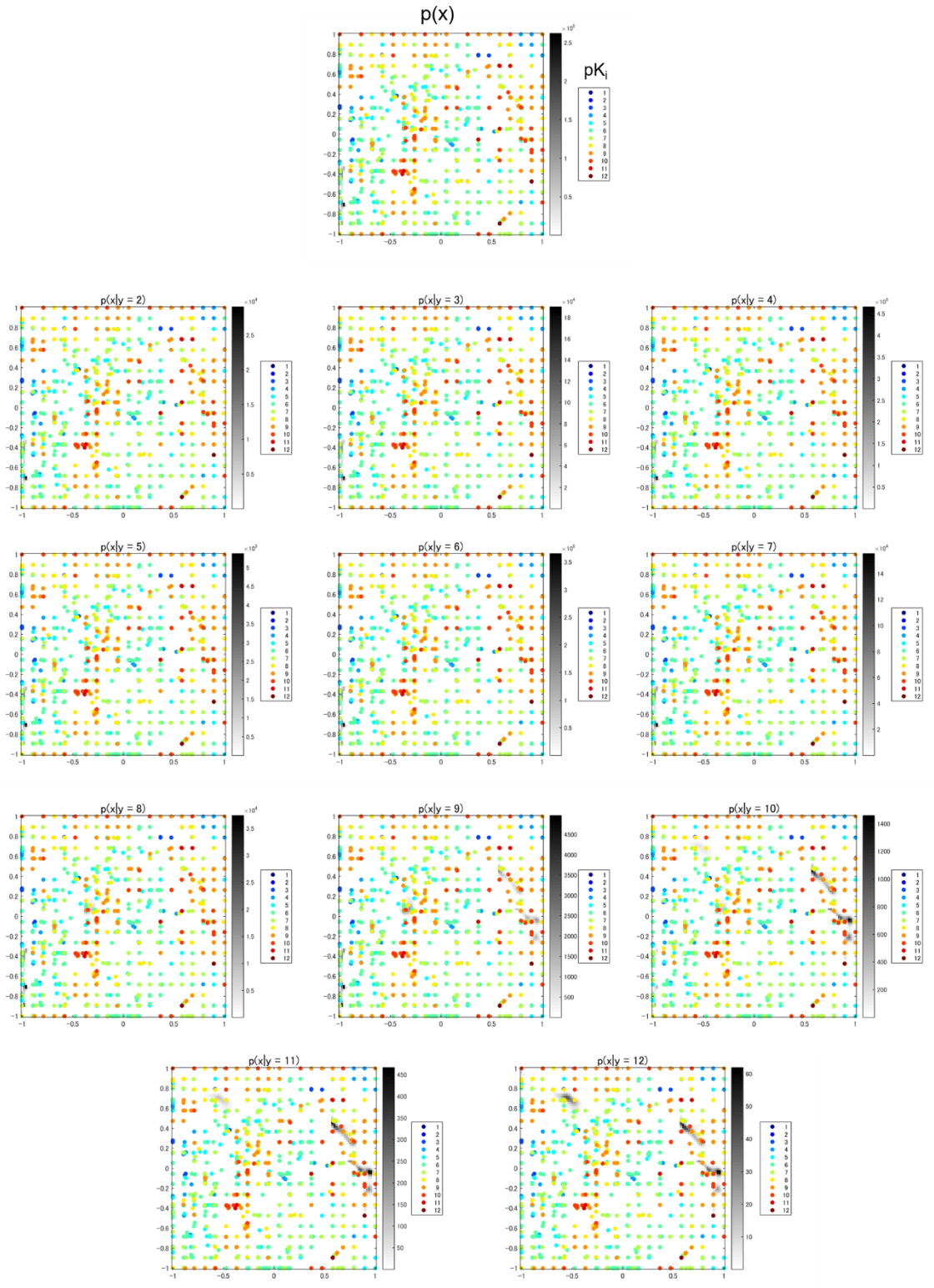


Figure F-1 Posterior densities projected on the GTM map. Hyper parameters for training GTM are described in section 4-3-4 . The top one is the prior distribution. The rest of the maps for the posterior distributions.

Appendix G

The proposed inverse QSAR methodology was applied to ligand design for thrombin. Here, validity of conducting the proposed methodology is demonstrated with the same test dataset as in section 4-3-3. In this Appendix, results of the same kind of analysis that has been conducted in section 3-5-3 are described. The number of training compounds was 1000, and that of the test compounds was 705, the number of Gaussians in a GMM is five. GMMs/cMLR exhibits that R^2 was 0.667 and RMSE 0.972 for the training data, and R_{pred}^2 was 0.425 and $RMSE_{pred}$ 1.294 for the test dataset.

First, in order to confirm that p(**x**|y) represents the closeness to the target y value as well as inherent features from p(**x**), p(**x**) and p(**x**|y) were compared with each other in various y values. p(**x**) is plotted against p(**x**|y) with different y values in Figure G-1 (for the test dataset). The used y values for inverse analysis were 9, 10, 11, 12. Color intensity represents the difference between a target y value and a measured one. Green colors mean the samples have large difference, and brown colors have small. In almost all pictures, the higher p(**x**|y) of a sample becomes, the lesser the absolute error between the measured y value and the target y value is shown. Furthermore, it can be seen that p(**x**|y) inherited the p(**x**) feature strongly. No matter how close the measured y value of a training sample is to the desired one (y), p(**x**|y) does not go across the diagonal line excessively. Therefore, p(**x**|y) represents the likelihood that **x** exhibits the y value after considering AD for training dataset. In contrast to the case study with aqueous solubility dataset in section 3-5-3, The range of both prior and posterior density is wider. In this study, 27 descriptors were employed whereas 6 descriptors were employed with the aqueous solubility dataset. Density of Gaussian in high dimensional space is sensitive to a subtle change of coordinates because of the curse of dimension. It might lead to a reasonable analysis using a set of coordinates obtained by some dimension reduction techniques, such as PCA.

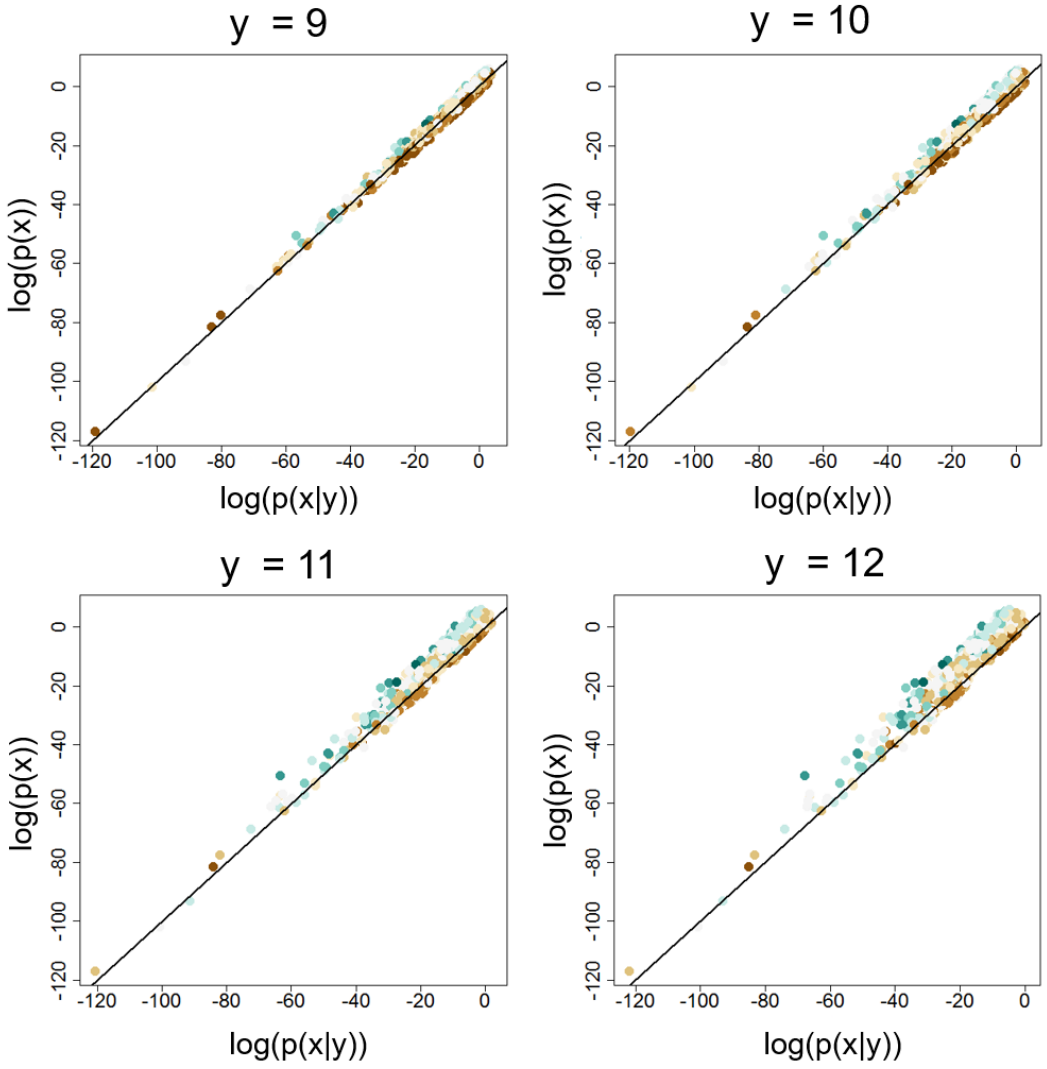


Figure G-1 p(**x**) is plotted against p(**x**|y) with a specific y value. The target y (pK_i) were 9, 10, 11, 12. Green colored dots have measured pK_i largely different from the target y value, whereas brown colored dots have measured pK_i that is small different from the target y value.

In order to validate how much the proposed methodology for structure generation is effective compared with a traditional screening one, the two strategies were applied to the test dataset: one is the selection of test set compounds based on predicted pKi values by the QSAR model, and the other is the proposed methodology shown in section 4-3-4, in which both predicted pKi values and p(**x**|y) given the specific target y value are taken into account. This comparison information can be inferred from Figure G-2. In this figure, black dots refer to selected compounds based solely on predicted pKi values by the model. In this case study, the threshold for determination of the sample selection was set to 2 based on the RMSE for the test dataset (1.294). For example, for y =11, compounds that have predicted pKi values from 9 to 13 were selected. There were 256, 124, 32, 5 compounds were selected for y = 9, 10, 11, and 12, respectively. These compounds are plotted on each corresponding picture. In

the figure G-2, absolute error means the difference between measured pKi and the target one. All pictures except the one for y = 12 show negative correlation between the absolute error and p(**x**|y). For y = 12, all 5 compounds have a small p(**x**|y), meaning those dots are out of AD. Therefore, introducing p(**x**|y) as a criterion for selecting chemical structures is expected to enhance the reliability of the screening, compared with a traditional QSAR-based screening.

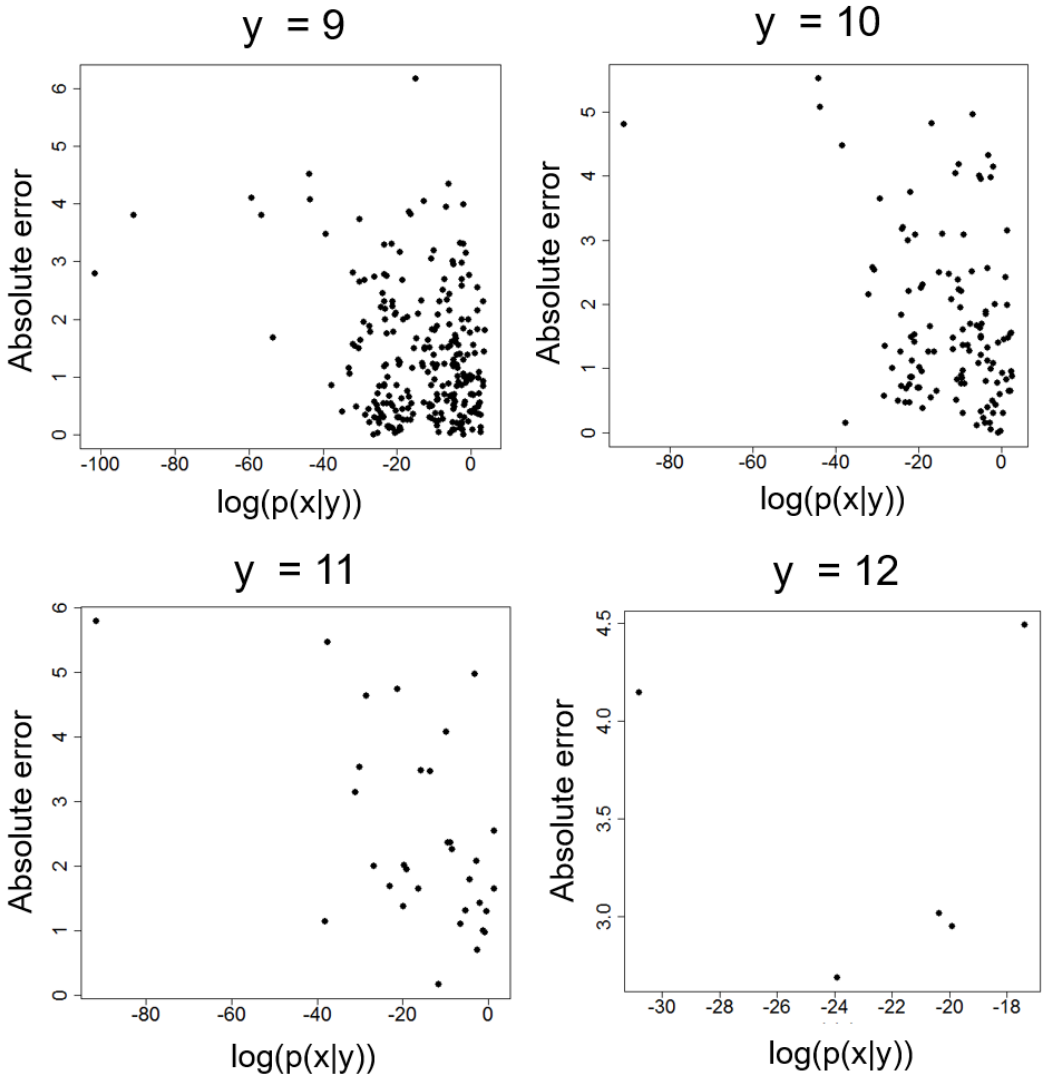


Figure G-2 Absolute error between measured y value and the target one is plotted against log(p(**x**|y)) with y = 9, 10, 11, 12. The scattered dots are compounds in the test dataset, dots which have the predicted y value are close to the target y value with error 2. For example, for y =11, compounds that have predicted pKi values from 9 to 13 were selected.

BIBLIOGRAPHY

1. Mitchell, J. B. O. Machine learning methods in chemoinformatics. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **4,** 468–481 (2014).
2. Wold, S., Sjöström, M. & Eriksson, L. PLS-regression: a basic tool of chemometrics. *Chemom. Intell. Lab. Syst.* **58,** 109–130 (2001).
3. Kholodovych, V. *et al.* Accurate predictions of cellular response using QSPR: a feasibility test of rational design of polymeric biomaterials. *Polymer (Guildf).* **45,** 7367–7379 (2004).
4. Verma, J., Khedkar, V. M. & Coutinho, E. C. 3D-QSAR in drug design--a review. *Curr. Top. Med. Chem.* **10,** 95–115 (2010).
5. Jobak, K. G. & Reid, A. R. C. ESTIMATION OF PURE-COMPONENT PROPERTIES FROM GROUP-CONTRIBUTIONS. *Chem. Eng. Commun.* **57,** 233–243 (1987).
6. Simamora, P., Miller, A. H. & Yalkowsky, S. H. Melting point and normal boiling point correlations: applications to rigid aromatic compounds. *J. Chem. Inf. Model.* **33,** 437–440 (1993).
7. Wakita, K., Yoshimoto, M., Miyamotomo, S. & Watanabe, H. A method for calculation of the aqueous solubility of organic compounds by using new fragment solubility constants. *Chem. Pharm. Bull. (Tokyo).* **34,** 4663–4681 (1986).
8. Klopman, G. & Zhu, H. Estimation of the Aqueous Solubility of Organic Molecules by the Group Contribution Approach. *J. Chem. Inf. Model.* **41,** 439–445 (2001).
9. Hou, T. J., Xia, K., Zhang, W. & Xu, X. J. ADME evaluation in drug discovery. 4. Prediction of aqueous solubility based on atom contribution approach. *J. Chem. Inf. Comput. Sci.* **44,** 266–75 (2004).
10. Leo, A., Jow, P. Y. C., Silipo, C. & Hansch, C. Calculation of hydrophobic constant (log P) from .pi. and f constants. *J. Med. Chem.* **18,** 865–868 (1975).
11. Rekker, R. F., Laak, A. M. Ter & Mannhold, R. On the Reliability of Calculated Log P-values: Rekker, Hansch/Leo and Suzuki Approach. *Quant. Struct. Relationships* **12,** 152–157 (1993).
12. Tetko, I. V. & Tanchuk, V. Y. Application of Associative Neural Networks for Prediction of Lipophilicity in ALOGPS 2.1 Program. *J. Chem. Inf. Model.* **42,** 1136–1145 (2002).
13. Jaworska, J., Nikolova-Jeliazkova, N. & Aldenberg, T. QSAR applicabilty domain estimation by projection of the training set descriptor space: a review. *Altern. Lab. Anim.* **33,** 445–59 (2005).
14. Lipinski, C. A. Drug-like properties and the causes of poor solubility and poor permeability. *J. Pharmacol. Toxicol. Methods* **44,** 235–249 (2000).
15. Morrill, J. A. & Byrd, E. F. C. Development of quantitative structure-property relationships for predictive modeling and design of energetic materials. *J. Mol. Graph. Model.* **27,** 349–55 (2008).
16. Lavecchia, A. & Di Giovanni, C. Virtual screening strategies in drug discovery: a critical review. *Curr. Med. Chem.* **20,** 2839–60 (2013).

17. Schneider, G. & Baringhaus, K.-H. in *De novo Molecular Design* 1–55 (Wiley-VCH Verlag GmbH & Co. KGaA, 2013). doi:10.1002/9783527677016.ch1
18. Roche, O. *et al.* Development of a Virtual Screening Method for Identification of 'Frequent Hitters' in Compound Libraries. *J. Med. Chem.* **45,** 137–142 (2002).
19. Baell, J. B. & Holloway, G. A. New substructure filters for removal of pan assay interference compounds (PAINS) from screening libraries and for their exclusion in bioassays. *J. Med. Chem.* **53,** 2719–40 (2010).
20. Ursu, O., Rayan, A., Goldblum, A. & Oprea, T. I. Understanding drug-likeness. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **1,** 760–781 (2011).
21. Oprea, T. I. Current trends in lead discovery: Are we looking for the appropriate properties? *J. Comput. Aided. Mol. Des.* **16,** 325–334
22. 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. *J. Chem. Inf. Comput. Sci.* **43,** 2048–56 (2003).
23. Liu, H. X. *et al.* The prediction of human oral absorption for diffusion rate-limited drugs based on heuristic method and support vector machine. *J. Comput. Aided. Mol. Des.* **19,** 33–46 (2005).
24. Bemis, G. W. & Murcko, M. A. Designing Libraries with CNS Activity. *J. Med. Chem.* **42,** 4942–4951 (1999).
25. Schneider, G. & Fechner, U. Computer-based de novo design of drug-like molecules. *Nat. Rev. Drug Discov.* **4,** 649–663 (2005).
26. Kirkpatrick, P. & Ellis, C. Chemical space. *Nature* **432,** 823–823 (2004).
27. Bohacek, R. S., McMartin, C. & Guida, W. C. The art and practice of structure-based drug design: a molecular modeling perspective. *Med. Res. Rev.* **16,** 3–50 (1996).
28. Reutlinger, M., Rodrigues, T., Schneider, P. & Schneider, G. Multi-objective molecular de novo design by adaptive fragment prioritization. *Angew. Chem. Int. Ed. Engl.* **53,** 4244–4248 (2014).
29. Nicolaou, C. A., Apostolakis, J. & Pattichis, C. S. De novo drug design using multiobjective evolutionary graphs. *J. Chem. Inf. Model.* **49,** 295–307 (2009).
30. Mishima, K., Kaneko, H. & Funatsu, K. Development of a New De Novo Design Algorithm for Exploring Chemical Space. *Mol. Inform.* **33,** 779–789 (2014).
31. Brown, N., McKay, B., Gilardoni, F. & Gasteiger, J. A graph-based genetic algorithm and its application to the multiobjective evolution of median molecules. *J. Chem. Inf. Comput. Sci.* **44,** 1079–87 (2004).
32. Venkatasubramanian, V., Chan, K. & Caruthers, J. M. Evolutionary Design of Molecules with Desired Properties Using the Genetic Algorithm. *J. Chem. Inf. Comput. Sci.* **35,** 188–195 (1995).
33. Hachmann, J. *et al.* Lead candidates for high-performance organic photovoltaics from high-throughput quantum chemistry – the Harvard Clean Energy Project. *Energy Environ. Sci.* **7,** 698–704 (2014).
34. Kawai, K., Nagata, N. & Takahashi, Y. De novo design of drug-like molecules by a fragment-based molecular evolutionary approach. *J. Chem. Inf. Model.* **54,** 49–56 (2014).
35. Morgan, H. L. The Generation of a Unique Machine Description for Chemical Structures-A Technique Developed at Chemical Abstracts Service. *J. Chem. Doc.* **5,**

107–113 (1965).
36. Weininger, D., Weininger, A. & Weininger, J. L. SMILES. 2. Algorithm for generation of unique SMILES notation. *J. Chem. Inf. Model.* **29,** 97–101 (1989).
37. Goldberg, D. E. Genetic Algorithms in Search, Optimization and Machine Learning. (1989). at <http://dl.acm.org/citation.cfm?id=534133>
38. Virshup, A. M., Contreras-García, J., Wipf, P., Yang, W. & Beratan, D. N. Stochastic voyages into uncharted chemical space produce a representative library of all possible drug-like compounds. *J. Am. Chem. Soc.* **135,** 7296–7303 (2013).
39. Brown, N., McKay, B. & Gasteiger, J. A novel workflow for the inverse QSPR problem using multiobjective optimization. *J. Comput. Aided. Mol. Des.* **20,** 333–41 (2006).
40. Gani, R., Nielsen, B. & Fredenslund, A. A group contribution approach to computer-aided molecular design. *AIChE J.* **37,** 1318–1332 (1991).
41. Arakawa, M., Yamada, Y. & Funatsu, K. Development of the computer software. *J. Comput. Aided Chem.* **6,** 90–96 (2005).
42. Miyao, T., Kaneko, H. & Funatsu, K. Ring-system-based Chemical Structure Enumeration for de Novo Design. *Yakugaku Zasshi* **136,** 101–106 (2016).
43. Kier, L. B., Hall, L. H. & Frazer, J. W. Design of molecules from quantitative structure-activity relationship models. 1. Information transfer between path and vertex degree counts. *J. Chem. Inf. Comput. Sci.* **33,** 143–147 (1993).
44. Hall, L. H., Kier, L. B. & Frazer, J. W. Design of molecules from quantitative structure-activity relationship models. 2. Derivation and proof of information transfer relating equations. *J. Chem. Inf. Comput. Sci.* **33,** 148–152 (1993).
45. Hall, L. H., Dailey, R. S. & Kier, L. B. Design of molecules from quantitative structure-activity relationship models. 3. Role of higher order path counts: Path 3. *J. Chem. Inf. Comput. Sci.* **33,** 598–603 (1993).
46. Skvortsova, M. I., Baskin, I. I., Slovokhotova, O. L., Palyulin, V. A. & Zefirov, N. S. Inverse problem in QSAR/QSPR studies for the case of topological indexes characterizing molecular shape (Kier indices). *J. Chem. Inf. Comput. Sci.* **33,** 630–634 (1993).
47. Skvortsova, M. I., Baskin, I. I., Slovokhotova, O. L., Palyulin, V. a & Zefirov, N. S. Inverse problem in QSAR/QSPR studies for the case of topological indexes characterizing molecular shape (Kier indices). *J. Chem. Inf. Comput. Sci.* **33,** 630–634 (1993).
48. Skvortsova, M., Fedyaev, K., Palyulin, V. & Zefirov, N. Inverse structure-property relationship problem for the case of a correlation equation containing the Hosoya index. *Dokl. Chem.* **379,** 191–195 (2001).
49. Randic, M. Characterization of molecular branching. *J. Am. Chem. Soc.* **97,** 6609–6615 (1975).
50. Kier, L. B. A Shape Index from Molecular Graphs. *Quant. Struct. Relationships* **4,** 109–116 (1985).
51. Faulon, J.-L., Churchwell, C. J. & Visco, D. P. The signature molecular descriptor. 2. Enumerating molecules from their extended valence sequences. *J. Chem. Inf. Comput. Sci.* **43,** 721–34 (2003).
52. Faulon, J.-L. On using graph-equivalent classes for the structure elucidation of large molecules. *J. Chem. Inf. Comput. Sci.* **32,** 338–348 (1992).
53. Churchwell, C. J. *et al.* The signature molecular descriptor. 3. Inverse-quantitative

structure-activity relationship of ICAM-1 inhibitory peptides. *J. Mol. Graph. Model.* **22,** 263–73 (2004).
54. Brown, W. M., Martin, S., Rintoul, M. D. & Faulon, J. L. Designing novel polymers with targeted properties using the signature molecular descriptor. *J. Chem. Inf. Model.* **46,** 826–835 (2006).
55. Contejean, E. & Devie, H. An Efficient Incremental Algorithm for Solving Systems of Linear Diophantine Equations. *Inf. Comput.* **113,** 143–172 (1994).
56. Wong, W. W. & Burkowski, F. J. A constructive approach for discovering new drug leads: Using a kernel methodology for the inverse-QSAR problem. *J. Cheminform.* **1,** 4 (2009).
57. Akutsu, T., Fukagawa, D., Jansson, J. & Sadakane, K. Inferring a graph from path frequency. *Discret. Appl. Math.* **160,** 1416–1428 (2012).
58. Akutsu, T. & Nagamochi, H. Comparison and enumeration of chemical graphs. *Comput. Struct. Biotechnol. J.* **5,** e201302004 (2013).
59. Rosipal, R. & Trejo, L. J. Kernel partial least squares regression in Reproducing Kernel Hilbert Space. *J. Mach. Learn. Res.* **2,** 97–123 (2002).
60. Smola, A. J. & Schölkopf, B. A tutorial on support vector regression. *Stat. Comput.* **14,** 199–222 (2004).
61. Kaneko, H. & Funatsu, K. Applicability domain based on ensemble learning in classification and regression analyses. *J. Chem. Inf. Model.* **54,** 2469–82 (2014).
62. Baskin, I. I., Kireeva, N. & Varnek, A. The One-Class Classification Approach to Data Description and to Models Applicability Domain. *Mol. Inform.* **29,** 581–587 (2010).
63. Seybold, P. G., May, M. & Bagal, U. A. Molecular structure: Property relationships. *J. Chem. Educ.* **64,** 575–581 (1987).
64. Bemis, G. W. & Murcko, M. A. The properties of known drugs. 1. Molecular frameworks. *J. Med. Chem.* **39,** 2887–2893 (1996).
65. Jochum, C. & Gasteiger, J. Canonical Numbering and Constitutional Symmetry. *J. Chem. Inf. Model.* **17,** 113–117 (1977).
66. Lindsay, R. K., Buchanan, B. G., Feigenbaum, E. A. & Lederberg, J. DENDRAL: A case study of the first expert system for scientific hypothesis formation. *Artif. Intell.* **61,** 209–261 (1993).
67. Benecke, C., Grüner, T., Kerber, A., Laue, R. & Wieland, T. MOLecular structure GENeration with MOLGEN, new features and future developments. *Fresenius. J. Anal. Chem.* **359,** 23–32 (1997).
68. Grüner, T., Laue, R. & Meringer, M. Algorithms for Group Actions: Homomorphism Principle and Orderly Generation Applied to Graphs. *DIMACS Ser. Discret. Math. Theor. Comput. Sci.* **28,** 113–122 (1997).
69. Colbourn, C. J. & Read, R. C. Orderly algorithms for graph generation. *Int. J. Comput. Math.* **7,** 167–172 (2007).
70. Sasaki, S. & Kudo, Y. Structure elucidation system using structural information from multisources: CHEMICS. *J. Chem. Inf. Comput. Sci.* **25,** 252–257 (1985).
71. Funatsu, K., Susuta, Y. & Sasaki, S. Introduction of two-dimensional NMR spectral information to an automated structure elucidation system, CHEMICS. Utilization of 2D-INADEQUATE information. *J. Chem. Inf. Comput. Sci.* **29,** 6–11 (1989).
72. Funatsu, K., Miyabayashi, N. & Sasaki, S. Further development of structure generation in the automated structure elucidation system CHEMICS. *J. Chem. Inf. Comput. Sci.*

28, 18–28 (1988).
73. Hartenfeller, M. *et al.* DOGS: reaction-driven de novo design of bioactive compounds. *PLoS Comput. Biol.* **8,** e1002380 (2012).
74. Rishton, G. M. Reactive compounds and in vitro false positives in HTS. *Drug Discov. Today* **2,** 382–384 (1997).
75. Kutchukian, P. S., Lou, D. & Shakhnovich, E. I. FOG: Fragment Optimized Growth algorithm for the de novo generation of molecules occupying druglike chemical space. *J. Chem. Inf. Model.* **49,** 1630–42 (2009).
76. Fink, T., Bruggesser, H. & Reymond, J.-L. Virtual exploration of the small-molecule chemical universe below 160 Daltons. *Angew. Chem. Int. Ed. Engl.* **44,** 1504–8 (2005).
77. Fink, T. & Reymond, J.-L. Virtual exploration of the chemical universe up to 11 atoms of C, N, O, F: assembly of 26.4 million structures (110.9 million stereoisomers) and analysis for new ring systems, stereochemistry, physicochemical properties, compound classes, and drug discove. *J. Chem. Inf. Model.* **47,** 342–353 (2007).
78. Ruddigkeit, L., van Deursen, R., Blum, L. C. & Reymond, J.-L. Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17. *J. Chem. Inf. Model.* **52,** 2864–2875 (2012).
79. Ruddigkeit, L., Blum, L. C. & Reymond, J.-L. Visualization and virtual screening of the chemical universe database GDB-17. *J. Chem. Inf. Model.* **53,** 56–65 (2013).
80. Bemis, G. W. & Murcko, M. A. Properties of Known Drugs. 2. Side Chains. *J. Med. Chem.* **42,** 5095–5099 (1999).
81. Taylor, R. D., MacCoss, M. & Lawson, A. D. G. Rings in drugs. *J. Med. Chem.* **57,** 5845–5859 (2014).
82. Lessel, U., Wellenzohn, B., Lilienthal, M. & Claussen, H. Searching Fragment Spaces with feature trees. *J. Chem. Inf. Model.* **49,** 270–279 (2009).
83. Gutman, I. & Polansky, O. E. *Mathematical Concepts in Organic Chemistry*. (Springer Berlin Heidelberg, 1986). doi:10.1007/978-3-642-70982-1
84. Shimizu, M., Nagamochi, H. & Akutsu, T. Enumerating tree-like chemical graphs with given upper and lower bounds on path frequencies. *BMC Bioinformatics* **12 Suppl 1,** S3 (2011).
85. Jindalertudomdee, J., Hayashida, M., Zhao, Y. & Akutsu, T. ベンゼン環を持つ木状化学構造の幅優先探索による列挙手法. in *第36回情報化学討論会講演要旨集* (2013).
86. Jindalertudomdee, J., Hayashida, M., Zhao, Y. & Akutsu, T. ナフタレン環を持つ木状化学構造の幅優先探索による列挙手法. in *第37回情報化学討論会講演要旨集* (2014).
87. Jindalertudomdee, J., Hayashida, M., Zhao, Y. & Akutsu, T. Enumeration method for tree-like chemical compounds with benzene rings and naphthalene rings by breadth-first search order. *BMC Bioinformatics* **17,** 113 (2016).
88. Zhao, Y., Hayashida, M., Jindalertudomdee, J., Nagamochi, H. & Akutsu, T. Breadth-first search approach to enumeration of tree-like chemical compounds. *J. Bioinform. Comput. Biol.* **11,** 1343007 (2013).
89. Randic, M., Mihalic, Z., Nikolic, S. & Trinajstic, N. Graphical bond orders: Novel structural descriptors. *J. Chem. Inf. Model.* **34,** 403–409 (1994).

90. McKay, B. D. Isomorph-Free Exhaustive Generation. *J. Algorithms* **26,** 306–324 (1998).
91. Nenand, T. *Chemical Graph Theory*. (CRC Press, 1992).
92. Oprea, T. I. On the information content of 2D and 3D descriptors for QSAR. *J. Braz. Chem. Soc.* **13,** 811–815 (2002).
93. Nettles, J. H. *et al.* Bridging chemical and biological space: 'target fishing' using 2D and 3D molecular descriptors. *J. Med. Chem.* **49,** 6802–6810 (2006).
94. 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. *J. Am. Chem. Soc.* **110,** 5959–67 (1988).
95. Rappe, A. K., Casewit, C. J., Colwell, K. S., Goddard, W. A. & Skiff, W. M. UFF, a full periodic table force field for molecular mechanics and molecular dynamics simulations. *J. Am. Chem. Soc.* **114,** 10024–10035 (1992).
96. Halgren, T. A. Merck molecular force field. I. Basis, form, scope, parameterization, and performance of MMFF94. *J. Comput. Chem.* **17,** 490–519 (1996).
97. Miyao, T., Kaneko, H. & Funatsu, K. Ring system-based chemical graph generation for de novo molecular design. *J. Comput. Aided. Mol. Des.* **30,** 425–446 (2016).
98. McKay, B. D. & Royle, G. F. *Constructing the Cubic Graphs on Up to 20 Vertices*. (1985). at <https://books.google.co.jp/books/about/Constructing_the_Cubic_Graphs_on_Up_to_2.html?id=MyaDuAAACAAJ&pgis=1>
99. Chemish: Chemometorics Software. at <http://www.cheminfonavi.co.jp/chemish>
100. Miyao, T., Arakawa, M. & Funatsu, K. Exhaustive Structure Generation for Inverse-QSPR/QSAR. *Mol. Inform.* **29,** 111–125 (2010).
101. Faulon, J.-L. & Bender, A. *Handbook of Chemoinformatics Algorithms*. (CRC Press, 2010). at <https://books.google.com/books?hl=en&lr=&id=O_9GU60TcGgC&pgis=1>
102. Miyao, T., Kaneko, H. & Funatsu, K. Ring-System-Based Exhaustive Structure Generation for Inverse-QSPR/QSAR. *Mol. Inform.* **33,** 764–778 (2014).
103. Miyao, T. 効率的な分子設計のためのInverse-QSPRを利用した構造生成システムの開発. (2010).
104. DRAGON for Windows (Software for Molecular Descriptor Calculation) version 5.4.
105. Water solublity (logS) database. at <http://modem.ucsd.edu/adme/databases/databases_logS.htm>
106. GVK database. at <http://www.gvkbio.com/>
107. Griffith, R., Luu, T. T. T., Garner, J. & Keller, P. A. Combining structure-based drug design and pharmacophores. *J. Mol. Graph. Model.* **23,** 439–446 (2005).
108. Martin, Y. C. *et al.* A fast new approach to pharmacophore mapping and its application to dopaminergic and benzodiazepine agonists. *J. Comput. Aided. Mol. Des.* **7,** 83–102 (1993).
109. Arakawa, M., Hasegawa, K. & Funatsu, K. Novel alignment method of small molecules using the Hopfield Neural Network. *J. Chem. Inf. Comput. Sci.* **43,** 1390–1395 (2003).
110. Reutlinger, M. *et al.* Chemically Advanced Template Search (CATS) for Scaffold-Hopping and Prospective Target Prediction for 'Orphan' Molecules. *Mol. Inform.* **32,** 133–138 (2013).

111. Schneider, G., Neidhart, W., Giller, T. & Schmid, G. 'Scaffold-Hopping' by Topological Pharmacophore Search: A Contribution to Virtual Screening. *Angew. Chem. Int. Ed. Engl.* **38,** 2894–2896 (1999).
112. Fechner, U., Franke, L., Renner, S., Schneider, P. & Schneider, G. Comparison of correlation vector methods for ligand-based similarity searching. *J. Comput. Aided. Mol. Des.* **17,** 687–698 (2003).
113. Miyao, T., Reker, D., Schneider, P., Funatsu, K. & Schneider, G. Chemography of natural product space. *Planta Med.* **81,** 429–435 (2015).
114. Reker, D. *et al.* Revealing the macromolecular targets of complex natural products. *Nat. Chem.* **6,** 1072–8 (2014).
115. Ashton, M. *et al.* Identification of Diverse Database Subsets using Property-Based and Fragment-Based Molecular Descriptions. *Quant. Struct. Relationships* **21,** 598–604 (2002).
116. Durant, J. L., Leland, B. A., Henry, D. R. & Nourse, J. G. Reoptimization of MDL keys for use in drug discovery. *J. Chem. Inf. Comput. Sci.* **42,** 1273–80
117. Bento, A. P. *et al.* The ChEMBL bioactivity database: an update. *Nucleic Acids Res.* **42,** D1083–1090 (2014).
118. Blum, L. C. & Reymond, J.-L. 970 million druglike small molecules for virtual screening in the chemical universe database GDB-13. *J. Am. Chem. Soc.* **131,** 8732–8733 (2009).
119. Rishton, G. M. Nonleadlikeness and leadlikeness in biochemical screening. *Drug Discov. Today* **8,** 86–96 (2003).
120. Landrum, G. RDKit: Open-source cheminformatics. at <http://www.rdkit.org>
121. boost C++ libraries.
122. Hakimi, S. L. On Realizability of a Set of Integers as Degrees of the Vertices of a Linear Graph. I. *J. Soc. Ind. Appl. Math.* **10,** 496–506 (1962).
123. Hosoya, H. Topological Index. A Newly Proposed Quantity Characterizing the Topological Nature of Structural Isomers of Saturated Hydrocarbons. *Bull. Chem. Soc. Jpn.* **44,** 2332–2339 (1971).
124. Skvortsova, M. I., Stankevich, I. V. & Zefirov, N. S. Generation of molecular structures of polycondensed benzenoid hydrocarbons using the randic index. *J. Struct. Chem.* **33,** 416–422 (1992).
125. White, D. & Wilson, R. C. Generative models for chemical structures. *J. Chem. Inf. Model.* **50,** 1257–1274 (2010).
126. Miyao, T., Kaneko, H. & Funatsu, K. Inverse QSPR/QSAR Analysis for Chemical Structure Generation (from y to x). *J. Chem. Inf. Model.* **56,** 286–299 (2016).
127. *Concepts and applications of molecular similarity /*. (Wiley, 1990).
128. Maggiora, G. M. On outliers and activity cliffs--why QSAR often disappoints. *J. Chem. Inf. Model.* **46,** 1535 (2006).
129. Bishop, C. *Pattern Recognition and Machine Learning*. (Springer-Verlag New York, 2006).
130. Dempster, a, Laird, N. & Rubin, D. Maximum Likelihood from Incomplete Data via the EM Algorithm. *J. R. Stat. Soc. Ser. B* **39,** 1–38 (1977).
131. Akaike, H. Factor analysis and AIC. *Psychometrika* **52,** 317–332 (1987).
132. Schwarz, G. Estimating the Dimension of a Model. *Ann. Stat.* **6,** 461–464 (1978).
133. DeSarbo, W. S. & Cron, W. L. A maximum likelihood methodology for clusterwise

linear regression. *J. Classif.* **5,** 249–282 (1988).
134. Viele, K. & Tong, B. Modeling with Mixtures of Linear Regressions. *Stat. Comput.* **12,** 315–330
135. Späth, H. A fast algorithm for clusterwise linear regression. *Computing* **29,** 175–181 (1982).
136. Manwani, N. & Sastry, P. S. K-plane regression. *Inf. Sci. (Ny).* **292,** 39–56 (2015).
137. Fraley, C., Raftery, a. E., Murphy, T. B. & Scrucca, L. MCLUST Version 4 for R: Normal Mixture Modeling for Model-Based Clustering, Classification, and Density Estimation. (2012).
138. Bishop, C. M. Novelty detection and neural network validation. *IEE Proceedings - Vision, Image, and Signal Processing* **141,** 217 (1994).
139. Fechner, N., Jahn, A., Hinselmann, G. & Zell, A. Estimation of the applicability domain of kernel-based machine learning models for virtual screening. *J. Cheminform.* **2,** 2 (2010).
140. Breiman, L. Random forests. *Mach. Learn.* **45,** 5–32 (2001).
141. Breiman, L. Bagging predictors. *Mach. Learn.* **24,** 123–140 (1996).
142. Weaver, S. & Gleeson, M. P. The importance of the domain of applicability in QSAR modeling. *J. Mol. Graph. Model.* **26,** 1315–1326 (2008).
143. Verleysen, M. & Francois, D. The curse of dimensionality in data mining and time series prediction. *Comput. Intell. BIOINSPIRED Syst. Proc.* **3512,** 758–770
144. ChemAxon. at <(http://www.chemaxon.com>
145. Costanzo, M. J. *et al.* In-depth study of tripeptide-based alpha-ketoheterocycles as inhibitors of thrombin. Effective utilization of the S1' subsite and its implications to structure-based drug design. *J. Med. Chem.* **48,** 1984–2008 (2005).
146. Boatman, P. D., Urban, J., Nguyen, M., Qabar, M. & Kahn, M. High-throughput synthesis and optimization of thrombin inhibitors via urazole α-addition and Michael addition. *Bioorg. Med. Chem. Lett.* **13,** 1445–1449 (2003).
147. Isaacs, R. C. A. *et al.* Structure-based design of novel groups for use in the P1 position of thrombin inhibitor scaffolds. Part 1: Weakly basic azoles. *Bioorg. Med. Chem. Lett.* **16,** 338–342 (2006).
148. Sanderson, P. E. J. *et al.* Azaindoles: moderately basic P1 groups for enhancing the selectivity of thrombin inhibitors. *Bioorg. Med. Chem. Lett.* **13,** 795–798 (2003).
149. Nantermet, P. G. *et al.* Design and synthesis of potent and selective macrocyclic thrombin inhibitors. *Bioorg. Med. Chem. Lett.* **13,** 2781–2784 (2003).
150. Sanderson, P. E. J. *et al.* Small, low nanomolar, noncovalent thrombin inhibitors lacking a group to fill the 'Distal binding pocket'. *Bioorg. Med. Chem. Lett.* **13,** 161–164 (2003).
151. Rai, R. *et al.* Development of potent and selective factor Xa inhibitors. *Bioorg. Med. Chem. Lett.* **11,** 1797–1800 (2001).
152. Pinto, D. J. P. *et al.* 1-[3-Aminobenzisoxazol-5'-yl]-3-trifluoromethyl-6-[2'-(3-(R)-hydroxy-N-pyrrolidinyl)methyl-[1,1']-biphen-4-yl]-1,4,5,6-tetrahydropyrazolo-[3,4-c]-pyridin-7-one (BMS-740808) a highly potent, selective, efficacious, and orally bioavailable inhibitor of bloo. *Bioorg. Med. Chem. Lett.* **16,** 4141–4147 (2006).
153. Ye, B. *et al.* Thiophene-anthranilamides as highly potent and orally available factor Xa inhibitors. *J. Med. Chem.* **50,** 2967–2980 (2007).
154. Venables, W. N. & Ripley, B. D. *Modern Applied Statistics with S*. (Springer New

York, 2002). doi:10.1007/978-0-387-21706-2
155. Blech, S., Ebner, T., Ludwig-Schwellinger, E., Stangier, J. & Roth, W. The metabolism and disposition of the oral direct thrombin inhibitor, dabigatran, in humans. *Drug Metab. Dispos.* **36,** 386–399 (2007).
156. Deng, J. Z. *et al.* Development of an oxazolopyridine series of dual thrombin/factor Xa inhibitors via structure-guided lead optimization. *Bioorg. Med. Chem. Lett.* **15,** 4411–6 (2005).
157. Bishop, C. M., Svensén, M. & Williams, C. K. I. GTM: The Generative Topographic Mapping. *Neural Comput.* **10,** 215–234 (1998).
158. Svensen, M. The GTM Toolbox. (1999).
159. McNaughton, R., Huet, G. & Shakir, S. An investigation into drug products withdrawn from the EU market between 2002 and 2011 for safety reasons and the evidence used to support the decision-making. *BMJ Open* **4,** e004221 (2014).
160. Lee, C. J. & Ansell, J. E. Direct thrombin inhibitors. *Br. J. Clin. Pharmacol.* **72,** 581–592 (2011).
161. Nilsson, M. & Ha, M. Compounds Binding to the S2 - S3 Pockets of Thrombin. **15,** 2708–2715 (2009).
162. Rasmussen, C. & Williams, C. Gaussian Processes for Machine Learning. *GAUSSIAN Process. Mach. Learn.* 1–247 at <http://apps.webofknowledge.com/full_record.do?product=UA&search_mode=GeneralSearch&qid=1&SID=W2TwPXVwPXXnadxl4v7&page=1&doc=1>
163. Vinkers, H. M. *et al.* SYNOPSIS: SYNthesize and OPtimize System in Silico. *J. Med. Chem.* **46,** 2765–2773 (2003).
164. Funatsu, K. & Sasaki, S.-I. Computer-assisted organic synthesis design and reaction prediction system, 'AIPHOS'. *Tetrahedron Comput. Methodol.* **1,** 27–37 (1988).
165. Bolten, B. M. & DeGregorio, T. From the analyst's couch. Trends in development cycles. *Nat. Rev. Drug Discov.* **1,** 335–336 (2002).
166. van de Waterbeemd, H. & Gifford, E. ADMET in silico modelling: towards prediction paradise? *Nat. Rev. Drug Discov.* **2,** 192–204 (2003).
167. Durrant, J. D. & McCammon, J. A. Molecular dynamics simulations and drug discovery. *BMC Biol.* **9,** 71 (2011).
168. Raha, K. *et al.* The role of quantum mechanics in structure-based drug design. *Drug Discov. Today* **12,** 725–731 (2007).
169. Olivares-Amaya, R. *et al.* Accelerated computational discovery of high-performance materials for organic photovoltaics by means of cheminformatics. *Energy Environ. Sci.* **4,** 4849 (2011).
170. Hachmann, J. *et al.* The Harvard Clean Energy Project: Large-Scale Computational Screening and Design of Organic Photovoltaics on the World Community Grid. *J. Phys. Chem. Lett.* **2,** 2241–2251 (2011).
171. Ertl, P., Rohde, B. & Selzer, P. Fast Calculation of Molecular Polar Surface Area as a Sum of Fragment-Based Contributions and Its Application to the Prediction of Drug Transport Properties. *J. Med. Chem.* **43,** 3714–3717 (2000).