Predicting the function of drug-like molecules methods and applications

Inaugural-Dissertation
to obtain the academic degree
Doctor rerum naturalium (Dr. rer. nat.)
submitted to the Department of Biology, Chemistry and Pharmacy
of Freie Universität Berlin

by
Hao Wang
from Liupanshui, Guizhou province, People's Republic of China

January 2017

Diese Arbeit wurde in dem Zeitraum von November 2012 bis Januar 2017 unter der Leitung von Prof. Dr. Ernst Walter Knapp am Institute für Chemie und Biochemie der Freien Universität Berlin im Fachbereich Biologie, Chemie und Pharmazie durchgeführt.

1. Gutacher: Prof. Dr. Ernst-Walter Knapp
2. Gutacher: Prof. Dr. Gerhard Wolber

Disputation am ___16. 3. 2017_____

Statutory Declaration

I hereby testify that this thesis is the result of my own work and research, except for any explicitly referenced material, whose source is listed in the bibliography. This work contains material that is the copyright property of others, which must not be reproduced without the permission of the copyright own.

Acknowledgement

First of all, I would like to thank Prof. Dr. Ernst Walter Knapp, China Scholarship Council, Free University of Berlin and Xiamen University for giving me this precious opportunity to study in Germany. I very much appreciate the supervision of Prof. Dr. Ernst Walter Knapp and valuable discussions during my doctoral work. In addition, I would also like to thank Dr. Bentzien Jörg and Dr. Ingo Mugge in Boehringer-Ingelheim group for their cooperation and valuable suggestions for my research projects. I am thankful for the support and great help of my colleagues in AG Knapp during my doctoral studies. Finally, I also thank Computational Systems Biology research training group (DFG - Graduiertenkolleg 1772) and Dahlem Research School of Free University of Berlin for their financial support and excellent program that allowed me to obtain useful academic skills and to extend my academic visions.

For the proofreading of this dissertation, I would like to thank Mr. Jovan Dragelj, Dr. Nadia Elgohobashi-Meinhardt, Dr. Cong Li, and Ms. Wendy Ashleigh Teo. Thanks to Dr. Florian Krull for translating my summary of dissertation into German.

在德国留学四年，生命中太多的事情在这里发生。似乎度过了一段远超过四年的时间。远离亲人，远离熟悉的环境。回首这四年，我需要感谢的人太多。在这里，感谢在柏林和我相遇的每一个朋友。和你们一起，收获了很多，也有了无数值得珍惜的时光。其中，可能也有一些在当时特别难过的时刻。但是，现在看来，那些难过都变成了快乐回忆的一部分。特别感谢远在中国的我的爸爸和妈妈，王海生先生和吴枚女士，还有我的其他亲人们。因为他们支持，我才能够如此安心的完成我的博士课题。最后，在这里也谢谢晓妤。谢谢你在博士课题完成前的最后一年，走进了我的生命，陪我度过了这特别难熬的最后一段。

选择读博是人生的历练，特别是在国外读博。完成这段修行，给了我一个不同的视角去审视人生。也希望这一段宝贵的人生财富能为我未来人生路带来更多的启示。

The table of contents

1. Introduction 1
1.1 Computer-Aided Drug Design 3
1.2 Machine Learning Techniques 6
1.2.1 Decision Tree 7
1.2.2 Random Forest 9
1.2.3 Artificial Neural Network 10
1.2.4 Naïve Bayesian Classifier 13

2 Methods 16
2.1 Dataset 16
2.2 Features 17
2.3 Cross-validation 18
2.4 Quality measures 19
2.5 Normalization 21
2.6 Linear classifier 22
2.6.1 Linear discriminate function 22
2.6.2 Training 25
2.6.3 Objective function 27
2.6.3.1 Loss function 27
2.7 Gradient descent algorithm 28
2.8 Regularization 31
2.8.1 Lasso Regularization 31
2.8.2 Ridge regression Regularization 32
2.9 DemPred 32
2.10 DemFeature 33
2.10.1 DemFeature-1 34
2.10.2 DemFeature-2 35
2.11 Quadratic features 36
2.12 Comparison of the performance between two models 37
2.13 Confidence measure 38

3 Applications 40
3.1 Project 1: Kaggle™ competition 40
3.1.1 Dataset 42
3.1.2 Results 43
3.1.2.1 Prediction results of DemPred 44
3.1.2.1.1 Linear features prediction results 45
3.1.2.1.2 Prediction results with quadratic features 49
3.1.2.2 Prediction results of DemFeature 53
3.1.2.2.1 Prediction results of DemFeature-1 53
3.1.2.2.2 Prediction result of DemFeature-2 57
3.1.2.3 Comparison with results from the Kaggle™ competition 62
3.1.2.4 *P*-values to compare different models 63
3.1.2.5 Measure confidence of prediction 65
3.1.3 Discussion 66
3.1.4 Conclusion 70
3.2 Project 2: Drug-induced phospholipidosis prediction 71
3.2.1 in silico methods assessing the potential of drug inducing PDL 76
3.2.1.1 Ploemen model for predicting PLD 77
3.2.1.2 Pelletier model: modified Ploemen model 77

3.2.1.3 Tomizawa model 78
3.2.1.4 Hanumegowda model 78
3.2.1.5 SMARTS models 79
3.2.2 Phospholipidosis database 81
3.2.2.1 Goracci phospholipidosis database 81
3.2.2.2 Independent phospholipidosis test dataset 84
3.2.3 Molecular descriptors for PLD 85
3.2.4 Results 86
3.2.4.1 Prediction of Goracci database 87
3.2.4.1.1 The prediction results of DemPred 87
3.2.4.1.2 The prediction results of DemFeature-1 91
3.2.4.1.3 The prediction results of TEM-confirmed compounds 95
3.2.4.1.4 Measurement of predictive confidence 98
3.2.4.1.5 PLD prediction performance for different models 99
3.2.4.1.6 Comparing prediction models by *P*-values 101
3.2.4.2 Prediction results of independent test set 101
3.2.5 Discussion 106
3.2.6 Conclusion 110
4 Summary 112
4 Zusammenfassung 114
Appendix 1. 116
Appendix 2. 120
Appendix 3. 122
References 131

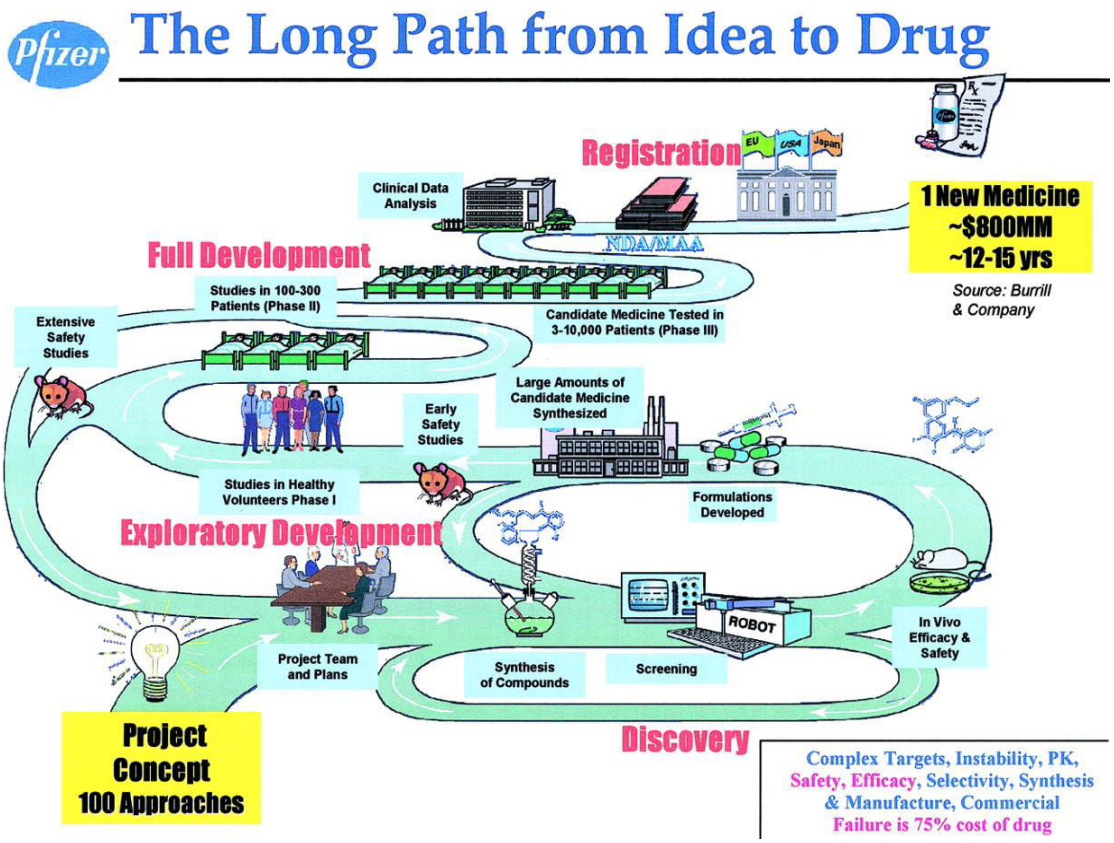

Figure 1.1: The process of drug development. The original figure by Pfizer Inc.

1. Introduction

The process of modern drug development mainly includes four stages[1,2], drug discovery, pre-clinical research, clinical trials and the approval of local FDA with a new drug application. After a new pharmaceutical drug is brought to market, it still needs to be supervised by regulation authorities for the drug safety.

In the stage of drug discovery, researchers focused on looking for key targets (e.g. a protein or a nuclear acid) of a disease. According to the information of the target discovered, chemical compounds designed for binding to the target, hoping that the binding event can stop or reverse the effects of the considered disease. At this stage, usually, thousands of compound candidates are prepared for screening. Test information needs to be gathered for evaluating whether a chemical entity is promising and should be further studied. Important information involves ADME of the compounds (i.e. Absorption, Distribution, Metabolization and Excretion); potential benefits and related mechanisms; recommended dosage; the way of taking drug (e.g. by mouth or by injection); potential side effects; the drug may affect different groups of people (e.g. gender and race) differently; potential interactions with other drugs and metabolism; the comparison of effectiveness with other similar drugs and more aspects.

Prior to human clinical trials stage, researchers need to carefully detect potential drug toxicities for excluding the possibilities of causing serious harm for people. During pre-clinical research, researchers would perform *in vitro* and *in vivo* experiments on microorganism and animal to further assess the dosing and toxicity levels. For those *in vivo* and *in vitro* experiments, U.S. FDA requires researchers to use the good laboratory practices following the minimum basic requirements.

Based on the information collected from the first two stages, researchers can submit an Investigation New Durg (IND) application to local regulatory authorities (e.g. FDA in U.S.). If an IND application is approved, the clinical trials can be performed on humans. Clinical trials usually involve four steps[2]: clinical phase 1 trials are mainly on healthy volunteers (20-100 people). In this phase, the goal of trials is to determine the safety and dosing; clinical phase 2 trials are used to get an initial reading of efficacy and further explore safety in small number of sick patients (up to several hundred people); in the clinical trials phase 3, large-scale trials are made for determining safety and efficacy in sufficiently large numbers of patients (300-3000 people). This is a pivotal step to evaluate the potential drug molecule.
Clinical trials phase 4 is a post-market surveillance study. Usually, after phase 3, a drug can receive the permission to market. But, it is still necessary to observe the behavior of the drug for a longer time period and a much larger patient population to detect rare or long-term adverse effects. During this phase, any discovered harmful effects may cause a drug being no longer sold, or being restricted to certain users[3]. The aim of the clinical trial is to test long-term or chronic toxicities of a lead compound. On the other side, the desired effectiveness of a drug also needs to be demonstrated in the clinical trials.

Overall, *de novo* drug design is a slow and complex process. Although the time and funds used for bringing a new drug to market are increasing, the number of new chemical entities approved is decreasing year by year. On average, a *de nove* drug from idea to going to market may take up to 18 years[4], of which the clinic trials may take around 6 years. To bring a new drug molecule from lab to market typically costs hundreds of millions or billions U.S. dollars[5,6]. Sometimes, developing a new drug may take a maximum of 2000 million dollars[6]. Apart from high cost and spending much time, the successful rate of chemical compounds investigated going into the market is quite low. According to the report of Kola *et al.*[7], for all compounds that enter the clinical trials, 30% compounds tested may fail owing to the lack of efficiency: 30% failures come from toxicological and safety tests and only 11% can pass the trials. A report from Stratmann *et al.* mentioned after pre-clinical stage, only about 10 drug

candidates can be qualified for clinical trials on human. Between 2007 to 2010, nearly 50% of drug candidates either failed during the clinical phase 3 trial or were rejected by the national regulatory agency[8].

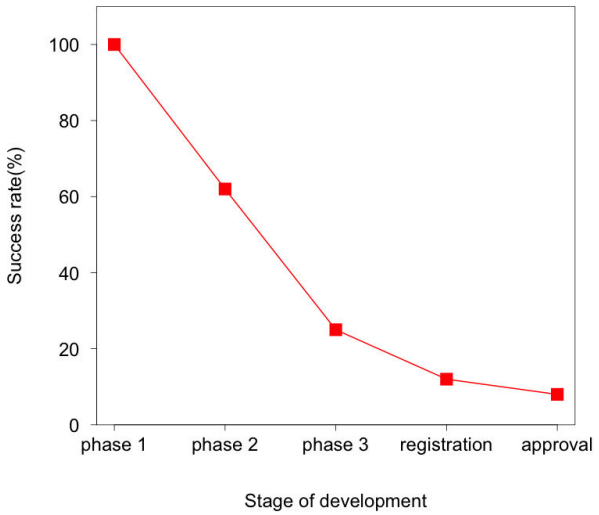

Figure 1.2: After entering the stage of clinical trials, the rate of success of compounds on the subsequent development phases. Data originated from the report written by Kola *et al.*[7] This figure was made by R v3.1.3.

Since bringing new drugs on the market has become so difficult, new techniques are demanded to accelerate and enhance the efficiency of the development of a new drug. The combination of computer techniques with pharmacological knowledge provides a cheaper and more efficient procedure for drug development, which makes pharmaceutical companies and research institutes greatly benefit from effectively accelerating drug development.

1.1 Computer-Aided Drug Design

The path of drug development is lengthy and complicated. To explore the behavior of a drug in a biological system using wet lab techniques is very time-consuming and intensively laborious task. With the development of computer science, there is a rapidly growing effort to apply computation techniques to the chemical and biological field in order to streamline drug discovery, design, development and optimization. Computer-aided or *in silico* design is being utilized to expedite and facilitate hit identification, hit-to-lead selection, optimize the absorption, distribution, metabolism, excretion and toxicity profile and avoid safety issues[9]. In this way effectiveness and efficiency of the drug discovery process will improve. Furthermore, *in silico* design can be an alternative method for decreasing the use of animals referring to ethical reasons. It can enhance the speed of development and save money. So far, the

computer-based drug development techniques have successfully provided assistance for the development of several important drugs, such as losartan (antihypertensive drug), ritonavir (antiviral drug), indinavir (antiviral), donepezil (anti-Alzheimer's disease) and more [10].

Before going to the clinical stage of drug development, the research focuses on drug design. The most fundamental goal in drug design is to predict whether a lead compound will bind to a biological target and the activity of compound can modulate this biological target and related biological responses. If the answer is yes, one needs to know how strong it is. Ligand-based drug design and structure-based design are common computational approaches applied in drug development. Ligand-based drug design is an indirect method to design a drug, not considering the target molecule explicitly. This method mainly relies on the knowledge of known molecules binding to the biological target of interest. These known molecules can be used to derive a pharmacophore model that defines the specific requirement of a molecule that can bind to the target[11]. In turn, this model may be used to design new molecule entities that interact with the target. Structure-based drug design, also called direct drug design, depends on the knowledge of the three-dimensional structure of the biological target obtained by X-ray crystallography or NMR spectroscopy. Based on the structure of the biological target, using interactive graphics and the knowledge of medicinal chemists, the candidate drugs that bind to the biological target with high affinity may be designed. Currently, various computational procedures can be used to design new drug candidates automatically[12].

(Q)SAR model, (Quantitive) Structure-Activity Relationships is an extensively used computer-aided drug development method[13]. Actually, (Q)SAR is a ligand-based method, which originated from Hansch model[14,15]. The basic hypothesis of (Q)SAR is that the structure of compounds determines their physical, chemical and biologic properties. This method can be applied as regression or classification model. For the regression task, the (Q)SAR model relates a set of "predictor" variables (X) to the potency of the response variable (Y). The classification (Q)SAR model relates the predictor variables to a categorical value of the response variable. For drug development with (Q)SAR modeling, the predictors normally consist of physicochemical properties, molecular descriptors, molecular finger prints and so on. The response variable of (Q)SAR is usually the biological activity of the drug. During the modeling process, first the relationship between chemical structure and biological activities is summarized by model, before new molecules can be predicted by this model.
The original Hansch model involves a linear function correlating a biological property to be predicted with steric, electronic and hydrophobic indices characterizing the chemical

architecture[16]. In the following years, based on this concept, a wealth of molecular descriptors and algorithms have been developed to build (Q)SAR models. Until now, there are several available commercial powerful (Q)SAR software packages to predict toxicities for aiding the drug development such as Derek for windows[17], MULTICASE[18] and TOPKAT[19].

- Derek for windows (Lhash Ltd., UK): a knowledge-based expert system. Its toxicity evaluation partly depends on alerts and chemical features associated with toxicity. All alerts or chemical features are based either on hypotheses relating to mechanisms of action of a chemical class or on observed empirical relationships. If a compound was associated with a level of likelihood of "equivocal or higher", it would be considered as toxicity positive. Otherwise, it would be negative.
- MC4PC (MULTICASE Inc.) is a molecular fragment-based approach. In the program, the predictive model can be automatically generated from the datasets provided by users. It reduces training set chemical structures to smaller chemical fragments (2- to 10-atom fragments) and then it identifies those fragments primarily associated with active compounds responsible for a biological target. Such fragments are called biophores. All compounds containing a specific biophore are removed from the training set. Then the next biophore would be identified based on the remaining part of the training set. Moreover, for each set of compounds sharing a specific biophore, the system would generate more molecular properties, which are defined as modulators in the program. Those molecular properties correlate with enhanced or diminished activity of a biophore (e.g. activating and inactivating fragments). The combination of these data is used to develop a QSAR model for estimating the potential toxicity of compounds to be tested.
- TOPKAT (Accelrys Inc.) predicts a range of toxicological endpoints. It includes three QSAR models. Each model can be applied to a specific class of chemicals. A submitted chemical structure would be given a probability being a developmental toxicant in rats. If the probability is below 0.3, it indicates no potential for developmental toxicity, if it is larger than 0.7, it signifies developmental toxicity potential. The probability range between 0.3 and 0.7 refers to indeterminate zone.

In more recent times, machine learning techniques have been introduced in the field of drug development[20,21]. Powerful statistical algorithms such as Random Forest, Artificial Neural Network, SVM and similar algorithms can efficiently extract rules and functions or procedures from large training data to build the correlation with biological activities. (Q)SAR models based on those machine learning algorithms can be used to optimize the biological

activities, target selectivity, physicochemical and other biological properties of selected chemical compounds. In the pharmaceutical field, machine learning models also can be used to eliminate chemical compounds that have undesirable effects, such as mutagens, carcinogens, teratogens and other toxic compounds.

1.2 Machine Learning Techniques

Machine Learning Techniques (MLT) are powerful drug design tools that can be used to construct QSAR models. MLT applications in drug development are growing rapidly[22]. The models built by MLT relate chemical structure with biological activity. They are useful in elucidating the mechanisms of the chemical-biological interaction. One of the most important properties of those statistical models is their high predictive power. This feature is crucial in modern drug development, which efficiently guides drug discovery research.

In the past decades, due to the rapid development in the artificial intelligence field, several statistical methods have enlarged the arsenal of drug development tools[23-25]. A number of MLTs have been proved to be quite useful for the construction of (Q)SAR models. These are Random Forest (RF), Support Vector Machine (SVM), Artificial Neural Network(ANN) and other MTL approaches, which have been recognized as important tools in drug discovery[26-30]. Usually, the MLTs involve three categories. Those can be classified as supervised learning methods (e.g. SVM, RF, Bayesian Network), unsupervised learning methods (e.g. Self-Organizing Maps, Clustering Algorithm) and hybrid methods (e.g. Counter Propagation Neural Network). Hybrid models possess advantages of supervised and unsupervised learning techniques. MLTs are well suited for (Q)SAR studies if a set of compounds with known biological activities is available for learning to construct a model. Besides, for each compound, a number of molecular descriptors or features with different contributions can be used to describe chemical compounds.

In the following, several typical MLT algorithms will be introduced including Decision Tree, Random Forest, Artifical Neural Network and Naïve Bayesian Classifier. In addition, it needs to be mentioned that SVM[31-33] is also a powerful and widely applied algorithm. However, the core properties of SVM are closely related to DemPred and DemFeature, developed in our group. Those three algorithms belong to the linear classifier types. The basic theories of the two linear classifiers DemPred and DemFeature used in the present work is outlined in Chapter 2.

1.2.1 Decision Tree

Decision trees (DT) are one of the popular machine learning methods. It was ranked No.1 in the *Top 10 Algorithms in Data Mining* published by Springer LNCS in 2008[34]. A decision tree possesses a tree-like structure and grows from a root. It comprises nodes, branches and leaves. Each internal node of decision tree represents a "test" on an attribute; each branch represents the result of the test on a node; the leaf of decision tree represents classification labels (Fig. 1.3).

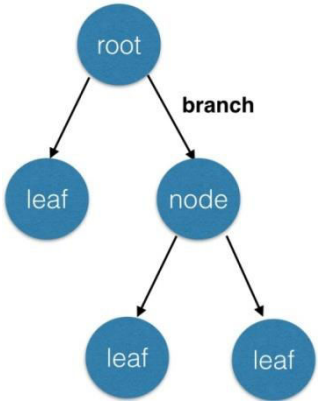

Figure 1.3: Tree-like structure of a decision tree model.

In 1987, J.R. Quinlan first invented the algorithm "iterative Dichotomiser 3" (ID3)[35], which was used to generate a decision tree from a dataset. In 1993, Quinlan upgraded this algorithm with an improved version C4.5[36]. Both algorithms, ID3 and C4.5 use the statistical calculation of information gain from a single attribute (feature, descriptor) to build a decision tree. Based on the concept of information gain, among all attributes of a training dataset, the attribute with maximum information gain is selected. The next step is from the remaining dataset to select the attribute with the most information gain to split. The process recurs on the smaller subsets. For each of the new subsets, the machine works in the same way until at each leaf all input samples belong to the same class. However, in this way it is easy to run into a problem over-fitting during modeling. The C4.5 algorithm addresses this problem by using a tree pruning technique. The procedure of constructing a decision tree is summarized below.

1. A dataset $\boldsymbol{C}$ involves N_C objects (molecules) $x \in \boldsymbol{C}$, which are assigned to two different classes $\boldsymbol{U}$ and $\boldsymbol{V}$ containing u and v objects, respectively. The information necessary for a complete classification of the whole dataset is

$$I(u, v) = -u/(u+v)\ log_2[u/(u+v)] - v/(u+v)\ log_2[v/(u+v)]. \quad (1.1)$$

2. An object $x \in \boldsymbol{C}$ can be assigned to one of the two classes $\boldsymbol{U}$ and $\boldsymbol{V}$ using the attribute (feature) A, which adopts the following value $A(x) \in \{A_i \mid i = 1,. . . . . N_A\}$. Using the values of the attribute, the objects x in $\boldsymbol{C}$ can be assigned (classified) to different subsets $\boldsymbol{C}_i$, $\boldsymbol{C} \equiv \{\boldsymbol{C}_i \mid i = 1,. . . . . N_A\}$, $\boldsymbol{C}_i \subset \boldsymbol{T}$. The individual subsets $\boldsymbol{C}_i$ contain u_i and v_i objects $N_C = \sum_i (u_i+v_i)$ belonging to the classes $\boldsymbol{U}$ and $\boldsymbol{V}$, respectively. The information necessary to classify all objects in the subset $\boldsymbol{C}_i$ is $I(u_i, v_i)$. The expected information necessary to further classify all objects in $\boldsymbol{C}$ using the subsets obtained with the attribute A is the weighted average

$$E(A) = \sum_{i=1}^{N_A} \frac{u_i + v_i}{u + v} I(u_i, v_i)\,. \tag{1.2}$$

If the partitioning in subsets $\boldsymbol{C}_i$ obtained with the attribute A was most successful the value of $E(A)$ should be as small as possible.

3. The information gained by using a classification based on the attribute A is

$$\text{gain}(A) = I(u, v) - E(A). \tag{1.3}$$

The strategy is to chose the attribute for classification, which yields the largest gain in information. $\boldsymbol{C}$ is the root and the subsets $\boldsymbol{C}_i$ are the first hierarchy of nodes of the growing tree.

4. For each subset $\boldsymbol{C}_i$ the above procedure (1, 2, 3) is repeated using the remaining attributes. Thus, from the root nodes grow branches and new nodes are created.

5. If all samples in a subset belong the same class, the corresponding node stops to grow branch and the model returns a leaf with this class. If on the other hand, the samples in this subset contain more than one class while this node stops, the model returns a leaf with the most frequently occurring class. If the algorithm does not reach a stop condition, new nodes are generated recursively.

Basically, the decision tree algorithm C4.5 builds a decision tree on the training dataset in the same way as ID3. However, C4.5 has been improved on several aspects. For example, C4.5 utilizes a threshold values to solve the problem of continuous and discrete attributes. Moreover, C4.5 allows attributes to be marked as a missing value. Thus, a missing value would not be used simply to calculate information gain and entropy. Besides, as mentioned above, C4.5 handles the over-fitting problem by removing branches, which are of little use, once the decision tree has been created.

Nowadays, the decision trees are a powerful classification tool, which are extensively applied by many researchers. Drug development is a very complicated task, but brings enormous benefits for mankind. Decision trees have been used by biologists, chemists and even

computer scientists to solve problems in drug development field. For example, Bach *et al.*[37] utilized decision tree technique to construct a model that assists applications for drug metabolism and kinetic studies, as well as for toxicological and pharmacological *in vivo* and *in vitro* testing. But, decision tree technique also contributed to solve the problem of antimicrobial resistance. For instance, Lira *et al.*[38] identified and synthesized two promising antimicrobial peptides through decision tree techniques and they successfully proved that both peptides have antimicrobial activity by *in vitro* experiments.

1.2.2 Random Forest

Random Forest (RF), as the name implies, is an ensemble algorithm, which is composed of many single decision trees. The output of an RF is the majority of votes of all decision trees trained for classification tasks or the average of the predicted values of all decision trees for a regression task. Leo Breiman[39] firstly developed the RF algorithm and used it as a trademark. More specifically the RF algorithm used nowadays combines the bagging method mentioned by Breiman and the idea of random selection of features introduced by Ho[40].

During the process of training of the RF, the Bagging or better say Bootstrap is the key technique of RF. Given a training dataset $\boldsymbol{D}$ including $\boldsymbol{X}$ and $\boldsymbol{Y}$. $\boldsymbol{X}=\{x_1,x_2,\ldots,x_n\}$ $\boldsymbol{n}$ is the total number of samples, where x_i denotes the sample and corresponding feature values. The response (expectation) values of the training data are $\boldsymbol{Y}=\{y_1,y_2,\ldots,y_n\}$. Bagging is repeated $\boldsymbol{B}$ times to select samples that constitute subsets of the training dataset with replacement. For each subset a decision tree is constructed.

The pseudo-code of training a single decision tree is shown below:

For $\boldsymbol{b}$=1,…, $\boldsymbol{B}$:

1. Pick up randomly with replacement $\boldsymbol{n}$ samples from the training dataset $\boldsymbol{D}(\boldsymbol{X},\boldsymbol{Y})$ and obtain a training subset, $\boldsymbol{D_b}(\boldsymbol{X_b}.\boldsymbol{Y_b})$.
2. Train a single decision tree $\boldsymbol{f_b}$ on $\boldsymbol{D_b}(\boldsymbol{X_b}.\boldsymbol{Y_b})$.

The trained RF model can be used to predict a new sample x_{new} by averaging the predicted values of the individual decision trees for a regression task or by taking the majority of votes of all individual decision trees for the classification task.

The technique of bagging effectively improves the prediction capacity of this algorithm, which decreases the bias of modeling. If only a single decision tree is used for prediction, the model is sensitive to noise in the training dataset. The average result of a number of decision trees can decrease this bias effectively. In the RF approach different decision trees are trained

using different subsets of the whole dataset available for training. Hence, not only different features but also different samples are used. In this way, avoids correlations among the decision trees are avoided.

Another important function of RF is the capacity to rank the feature importance. Feature ranking can be performed in two ways: Gini feature importance and permutation feature importance.[41] The Gini feature importance is based on the Decrease of Gini Impurity (DGI). For a single decision tree, during the training process, the important feature with a high DGI would be selected first when building a decision tree. Thus, when considering the decision trees, the important features possess high Gini importance.

Permutation features importance is calculated based on the prediction accuracy rather than working on the feature that is used to split the dataset into subsets. Before detailing the permutation importance computation, the Out-Of-Bag (OOB) error must be understood. For a single decision tree, OOB error is evaluated using the subsets of the training dataset, which are not used to construct the actual decision tree. The OOB error estimates the performance of the RF for optimizing parameters. The way of computing permutation feature importance is to compare the difference between OOB errors resulting from the dataset obtained from a random permutation of the targeted features and the OOB error resulting from the original dataset. Usually, during this process, the important features increase the OOB error.

As an outstanding algorithm, the RF can address high-dimensional problems in which the number of features is much larger than number of samples. It also performs well on coping with highly correlated dataset. Considering "omics" data in the biological field that are characterized by a high degree of complexity, RF approaches have many advantages in solving biological and medical problems. For example, Goldstein *et al.*[41] depicted the application of RF on genetic epidemiology in detail and Chen *et al.*[42] made an extensive overview of applications of RF to bioinformatics.

1.2.3 Artificial Neural Network

In the machine learning field, the Artificial Neural Network (ANN) is a mathematic algorithm inspired by features of biological neural networks. The ANN simulates in an abstract way how the human brain works. It actually presents a system of many interconnected nodes. These nodes can be viewed as neurons in the brain which exchange messages between each other. In 1943, a neurophysiologist, Warren McClloch and a logician, Walter Pitts designed a computational algorithm to simulate neural networks[43]. This mathematic algorithm is called threshold logic. Since this model was created, the research for neural network went in two

different directions. One is to simulate the biological processes in the brain and the other mainly focuses on developing artificial intelligence of neural networks.

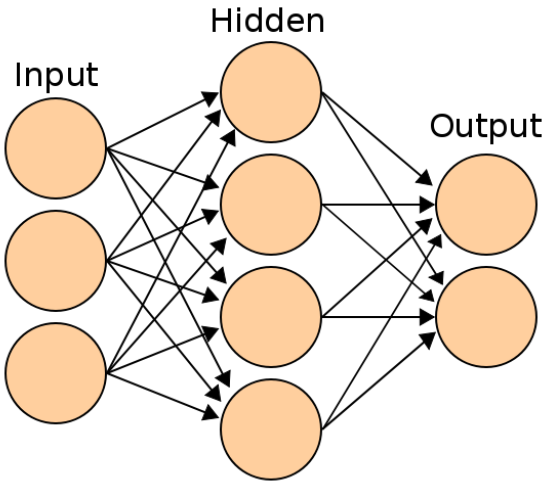

Figure 1.4: The basic architecture of an Artificial Neural Network. The original figure is taken fromWikipedia.https://commons.wikimedia.org/wiki/File:Artificial_neural_network.svg.

ANN comprises a large family of machine learning algorithms of different types. These are for instance, back propagation neural network[44], radial basis function network[45], recurrent neural network[46] and more. Generally, all types of ANN have a basic topology consisting of neurons in an input layer, one or more hidden layers and an output layer. There can be more than one hidden layer, which mainly depends on the complexity of neural network. The multi-layer perceptron (MLP) is the most common ANN algorithm. MLP includes several layers of neurons working as basic ANN structure. The neurons of the input layer adopt the values of the attributes of the samples from the dataset to be analyzed. The value x_i of the input neuron i is transferred to a neuron k of the next layer multiplied with a specific weight w_{ik}. The sum of these products merging at the neuron k of the next layer

$$v_k = \sum_{i=1}^{n} w_{ki} x_i \tag{1.4}$$

is processed by an activation function ϕ that yields the value γ_k

$$\gamma_k = \phi(v_k + v_{k0}) \tag{1.5}$$

placed in the neuron k of the next layer. In Equation(1.5) v_{k0} is a bias term. There are activation functions ϕ for different tasks depending on the specific situation (e.g. sigmoid function and hyperbolic tangent function). The aim of activation function is to scale and normalize the input for the neurons in the next layer.

Commonly used activation functions are the sigmoid, Equation(1.6), and the hyperbolic tangent function, Equation(1.7)

$$\Phi(v) = \frac{1}{1+e^{-v}} \tag{1.6}$$

$$\Phi(v) = \frac{e^{v} - e^{-v}}{e^{v} + e^{-v}}. \tag{1.7}$$

For each connection between two neurons of subsequent layers, the activation function transforms the input information into next layer until the output layer is reached. In the learning process the numerical values of the weights are adjusted to yield desired values for the neurons of the next layer, which may be a hidden layer or the output layer. The final value of the output layer is the predicted value of entire ANN model.

For predicting an output, the ANN needs to be trained efficiently based on the known dataset. Under the supervised training paradigm, the process of supervised learning needs the dataset with the known response for each observation. The initially the weights have random values between -1 and 1. After completion of a training cycle, the ANN model calculates a preliminary output value as the prediction, which is compared with the true output. The difference is the error in the prediction. In the ANN algorithm this error is feed back through the whole network. The values of the weights are adjusted to lower this error. The rule of error adjustment follows below

$$w_{ki}(a) = \delta_k x_i \eta . \tag{1.8}$$

In eq. 1.8, $w_{ki}(a)$ is the updated weight at a^{th} training cycle, which is in proportion to the input value x_i, to which the weight is applied, the error δ_k and the learning rate η. The influence of an input observation on the error is in proportion to the weight change of a neuron. The training speed is determined by the learning rate η. With increasing learning rate η, the training speed increases. However, a large value could lead to non-convergence of the model. For each cycle, the weight would be modified slightly in the direction going to a smaller error, until a target error is reached or no improvement of the error is observed. On another hand, if the training rate is too small, the process of training will be slow[47]. Choosing a value of the learning rate depends on the specific problem.

ANN is a high-throughput technique. With the advent of this technique, ANN facilitates to solve many biological problems, especially on the genomics or proteomics fields. Khan *et al.*[48] use ANN to deal with the complex genomic datasets. In their study, ANN was employed to classify 88 round blue-cells tumors into four diagnostic categories based on cDNA micro-

array analysis of 6000 genes. In proteomics, Rogers *et al.*[49] used ANN as the technique to detect early onset of renal cancer from the dataset generated by SELDI-TOF mass spectrometry.

The application of ANN is general. It can be used to process data containing complex relationships and interactions, especially for the non-linear dependencies between data and outcome, which is usually difficult to interpret. More importantly, for datasets with noisy information the ANN algorithm is particularly suitable. However, the ANN also has its limitations. Although the ANN can be used to solve many complex problems, its results cannot be logically analyzed. The reason is the inherent the "black box" effect of ANN. With hidden layers the accuracy of prediction can be improved but it also obviously decreases the speed of computation. Therefore, usually only ANN with one or two hidden layers are employed.

1.2.4 Naïve Bayesian Classifier

Bayes theorem is named after the Reverend Thomas Bayes (1702-1761), who studied how to compute a distribution for the probability parameter of a binomial distribution. The Naïve Bayesian Classifier (NBC) is the algorithm for constructing classifiers based on this theorem. In this algorithm, the features are considered to be independent of each other (the strong independence assumption). This is the reason why the algorithm is called "naive". Naive Bayesian classifier has been developed more than 60 years. On the text categorization field, especially on the spam process, it is still the very popular algorithm.

The principle assumption is that the value of a particular feature is independent of the value of any other features. For example, if we want to identify the gender of a person, the data includes three features: height, weight and foot size. The model of Naïve Bayesian Classifier considers each feature contributes independently to the probability that the person is a male or a female. Possible correlations among features are ignored.

Although the NBC algorithm is based on such a simplifying assumption, it can be trained easily to solve complex problems of supervised learning. An analysis written by Zhang *et al.*[50] showed an optimistic ability of Naïve Bayesian. In 2006, Caruana *et al.*[51] made a comprehensive comparison of NBC with other classification algorithms. Among several algorithms considered in this comparison, the NBC algorithm was outperformed by other popular machine learning algorithms.

Given a dataset with class variable $\boldsymbol{y}$ and the vector $\boldsymbol{x}$=(x_1,…, x_n) representing n independent features, the formulation of conditional probability based on the Bayes theorem can be written as follows

$$P(y_k \mid x_1,....,x_n) = \frac{P(y_k)P(x_1,...x_n \mid y_k)}{P(x_1,...x_n)}. \tag{1.9}$$

Using the relation

$$P(y_k, x_1,....,x_n) = P(y_k \mid x_1,....,x_n) * P(x_1,....,x_n), \tag{1.10}$$

We can recursively expanded $P(y_k, x_1,\ldots, x_n)$ as

$$\begin{aligned} P(y_k, x_1,....,x_n) &= P(x_1,....,x_n, y_k) \\ &= P(x_1 \mid x_2,....,x_n, y_k) * P(x_2,....,x_n, y_k) \\ &= P(x_1 \mid x_2,....,x_n, y_k) * P(x_2 \mid x_3,....,x_n, y_k) * P(x_3,....,x_n, y_k) \\ &= .... \\ &= P(x_1 \mid x_2,....,x_n, y_k) * P(x_2 \mid x_3,....,x_n, y_k) \ldots P(x_n \mid y_k) * P(y_k). \end{aligned} \tag{1.11}$$

Furthermore, assuming independence among all features x_i for a given category y_k we can write

$$P(x_i \mid x_{1+1},...,x_n, y_k) = P(x_i \mid y_k), \tag{1.12}$$

for i=1, 2,…, n yielding

$$P(y_k, x_1,...x_n) = P(y_k)\prod\nolimits_{i=1}^{n} P(x_i \mid y_k). \tag{1.13}$$

Since the features do not obey the laws of probability $P(x_1,\ldots, x_n)$ in Equation(1.10) is constant Z yielding from Equation (1.10) and (1.11)

$$P(y_k \mid x_1,...,x_n) = \frac{1}{Z} P(y_k)\prod_{i=1}^{n} P(x_i \mid y_k). \tag{1.14}$$

Predicting the expectation value $y^{(j)}$ for the sample (molecule) j, we take the value y_k for which $P(y_k \mid x_1^{(j)},\ldots, x_n^{(j)})$, Equation (1.14) assumes the maximum

$$y^{(j)} = \underset{k\in\{1,...,K\}}{\arg\max}\, P(y_k)\prod_{i=1}^{n} P(x_i^{(j)} \mid y_k), \tag{1.15}$$

where $x_i^{(j)}$ are the features of the sample j. This is known as the Maximum A Posteriori (MAP) decision rule. In Equation (1.15), $P(y_k)$ is the relative frequency that the expectation value y_k occurs in the training set and $P(x_i \mid y_k)$ is the frequency to find the feature value x_i for a given expectation value y_k referring to class k in the training set.

There are a several NBC algorithms in use. The difference between them mainly depends on the type of distribution used for $P(x_i|y_k)$. In real-world problems, we often meet continuous data. When dealing with continuous values of the features, a typical assumption is that

continuous values associated with each class are distributed according to the Gaussian distribution. If feature x_i is continuous, the training data would be segmented by the class and then calculate the mean value $\mu(y_k)$ and variance $\sigma(y_k)$ of x_i in each class. Thus the probability distribution of x_i associated with class y_k can be expressed as

$$P(x_i \mid y_k) = \frac{1}{\sqrt{2\sigma^2(y_k)\pi}} \exp\left(-\frac{(x_i - \mu(y_k))^2}{2\sigma^2(y_k)}\right). \quad (1.16)$$

Bernoulli NBC is another variant, where the $P(x_i|y_k)$ obey multivariate Bernoulli distributions. It is applicable for data where the features assume binary values, i.e. $x_i \in \{0, 1\}$. Therefore, this algorithm requires samples to be represented by binary-valued feature vectors. The decision rule for Bernoulli Naïve Bayes procedure is based on the distribution

$$P(x_i \mid y_k) = P(i \mid y_k)x_i + (1 - P(i \mid y_k))(1 - x_i) \quad (1.17)$$

This expression penalizes the non-occurrence of feature i that is an indicator for class y_k. The Bernoulli NBC is usually used in text classification. During the process of text classification, word occurrence is taken as a feature vector.

Although the Bernoulli NBC is often used to solve problems of text classification, it can also be used to solve biological problems such as classification of sequence data. Qiong Wang *et al*. developed a program, Ribosomal Database Project (RDP), which is based on the Bernoulli NBC technique. RDP can rapidly and accurately classify bacterial 16S RNA sequences into the new higher-order taxonomy and is suitable for the analysis of single rRNA sequence and for the analysis of libraries of thousands of sequences.

2 Methods

Utilizing machine learning techniques in drug discovery would result in effectively saving a significant amount of money and time. Since the extensive use of artificial intelligence in biological and medical fields has become common, an immense improvement has been made in drug development, greatly simplifying and accelerating drug development, especially in the early stages. However, the design of machine learning algorithms is also a complicated process. In general, the modeling for solving biological problems needs to consider several important elements, which consist of collecting reliable datasets, developing algorithms, confidence measure, and testing. Each of these elements decides on the robustness of the model obtained. In this chapter, the components used to construct our model are introduced in detail and the principles of the algorithms used for modeling are discussed.

2.1 Dataset

Generally, for machine learning tasks, the dataset contains two parts: a training set and a test set. The usage of the training set is the base for constructing the model, which can be used to discover a potentially predictive relationship through mathematic rules. Usually, the predictive model is adjusted to its task by optimization of its parameters. For example, a very simple model relating one feature $\boldsymbol{x}$ with the response value $\boldsymbol{y}$ in two dimensional space, $\boldsymbol{y' = f(x, \beta) = \beta_0 + \beta_1 x}$, involves two parameters, namely, bias, $\boldsymbol{\beta_0}$ and slope, $\boldsymbol{\beta_1}$, which are the parameters obtained by learning from a training set.

For a machine learning task, after establishing the predictive relationship based on the training set, a test set is necessary for assessing the strength and usefulness of this predictive relationship. The test set must be strictly independent of the training set. It is strictly prohibited that a sample in the test set participates in training of the model. Moreover, a significant principle is that it must follow the same probability distribution as the training set. However, in addition to the training set and test set, a validation set is also necessary. During modeling, to properly select model parameters, the validation set can be used to optimize certain global parameters. For example, in the RF[39], both the number of single decision trees and the number of features for each tree need to be optimized through a validation set. Typically, the validation set is separated from the training set. A common proportion used to separate a validation set from the training set is 3:7[52].

In my doctoral dissertation, there are two datasets studied. One is the Kaggle competition contest launched by Boehringer Ingelheim Inc. in 2012[53]. This dataset provided a realistic up-

to-date prediction scenario for drug classification and the predictions submitted in this contest from different professional persons and groups that certainly come close to the theoretical limits of what can be achieved for this prediction task. Another dataset used in the thesis is the phospholipidosis dataset[54]. Phospholipidosis is a drug-induced side effect. The early prediction of this side effect can accelerate drug development and decrease the cost of research. Moreover, in recent years, this side effect has gained more and more interests from pharmaceutical communities, and has been extensively studied around the world so that we can reliably evaluate our model by comparing with other *in silico* methods to predict phospholipidosis.

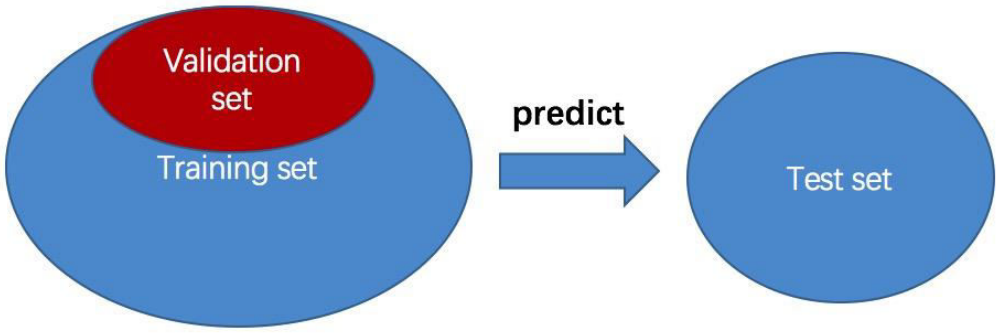

Figure 2.1: Dataset: training set, test set and validation set. The red part is the validation set separated from the training set.

The prediction tasks studied in this thesis are binary problems, which only include a positive set and a negative set. For a pharmaceutical problem, usually the positive set refers to the set of compounds causing a given biological phenomenon and the remaining compounds belong to the negative set.

2.2 Features

The first condition of using the computational method to explore drug-like compounds is that the compounds existing in the real world be transferred into the virtual space in the computer. A compound can be described quantitatively and qualitatively by its physicochemical properties, theoretical molecular properties and topological fingerprints. In bioinformatics, this step is a process of numerically vectorizing compounds. Thus, a compound vector can be input into computer for modeling or prediction.

Currently, several software packages are available to compute features that describe molecular compounds. Those features are also called molecular descriptors. Table 2.1 lists the information of several popular applied software packages. These packages provide thousands

of molecular descriptors enabling prediction models to consider a broad spectrum of pharmacodynamics, pharmacokinetic and toxicological properties, and others.

Since the number of molecular descriptors can be very large, it is necessary to provide tools that allow to reduce the number of features. Therefore, methods for selecting subsets of features that are relevant for the predicted property can reduce hidden dependencies among the features. This allows to interpret the results of predictions and saves CPU time. Datasets with noisy information could decrease prediction performance. In these cases, approaches that reduce redundant features using for instance mutual information[55], Pearson product-moment correlation coefficient[56] are most helpful. In this study, the Lasso method[57] was employed to perform feature selection based on the weights of the features. The Lasso method is an embedded method which performs feature selection as part of the model construction process.

Table 2.1: Overview of popular software packages for computing molecular descriptors

software	source or reference
DRAGON[58]	Todeschini et al., 2005
Molcom-Z[59]	Hall et al., 2002
JOELib[60]	Wegner, 2005
Xue descriptors set[61]	Xue et al., 2004
MODEL[62]	Li et al., 2007
CDK Development Kit[63]	Steinbeck et al., 2006
Daylight[64]	Daylight Inc.
Volsurf[65]	Molecular Discovery Ltd.
ChemDes[66]	Dong et al., 2015
MOE[67]	Chemical Computing Group Inc.

2.3 Cross-validation

Cross-validation is a technique used for model validation[68,69]. The aim of the method is to assess the model and optimize parameters based on a dataset with known biological response variable. This dataset is separated from the training set. In other words, it is the validation dataset mentioned in Chapter 2.1. Technically, cross-validation includes two types: exhaustive cross validation and non-exhaustive cross-validation.

The principle of exhaustive cross validation is to learn and test all possible ways of dividing the original training set into a sub-training set and a validation set. For example, Leave-one-out (LOO) cross validation uses only one sample as the validation set and the remaining samples are the sub-training set. If the original training set has $\boldsymbol{n}$ samples, this method requires learning and validating $\boldsymbol{n}$ times. However, in the case of a dataset having a large number of samples, this method is time-consuming. Non-exhaustive cross-validation does not

compute all ways of splitting the original training set. For example, 5 folds cross-validation is a non-exhaustive method. In 5 folds cross-validation, the original training set is randomly divided into 5 subsets. The size of each subset is nearly equal. Of these 5 subsets, there is one subset left as the validation set and the remaining 4 subsets are used as the sub-training set. This process is repeated 5 times to ensure each subset is used once exactly. Depending on the size of the dataset, sometimes, to improve the accuracy of validation, 10 folds cross-validation also can be used[70]. This kind of method is generally called ***p*** fold cross validation. ***p*** is an integer. When ***p*** is equal to the size of the dataset, then ***p*** folds exactly becomes exactly same as leave-one-out validation. In addition, depending on the specific situation, the ***p*** folds cross-validation can be repeated for several rounds.

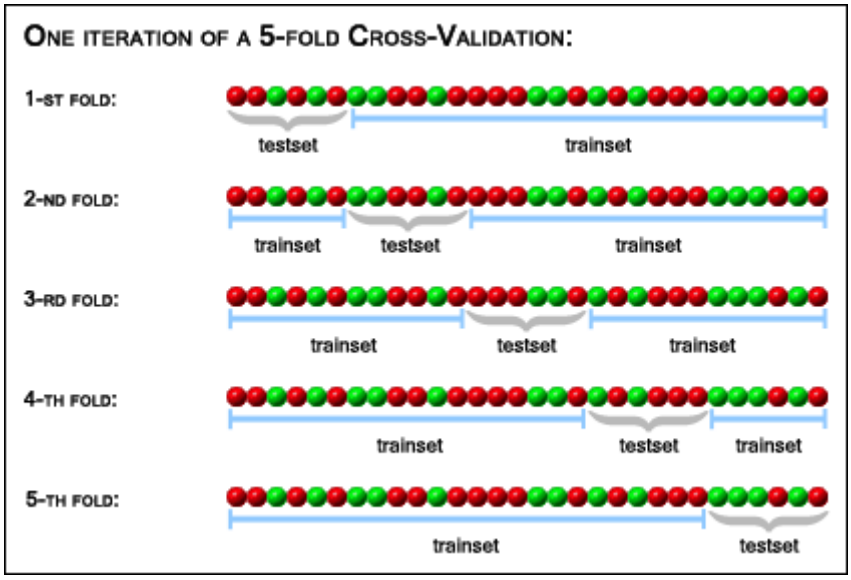

Figure 2.2: Principle of 5 folds cross-validation. For some situations, this process can be repeated for several rounds. Original figure from stats.stackexchange.com/questions/1826/cross-validation-in-plain-english.

2.4 Quality measures

To assess the prediction performance of a model, model quality measures need to be defined. Quality measures not only assess the prediction ability of the model but also allow us to compare other competitive models. In this study, the predicted tasks are binary classification problems. For assessing a binary problem, 4 values must be defined as a premise. They are ***TP***, ***TN***, ***FP*** and ***FN***. ***TP*** is the number of true positive samples, ***TN*** is the number of true negative samples, ***FP*** is the number of false positive samples and ***FN*** is the number of false negative samples. Positive and negative are two classification signs, respectively. Those four values

construct a 2×2 confusion matrix to evaluate machine learning prediction performance as shown on Figure 2.3.

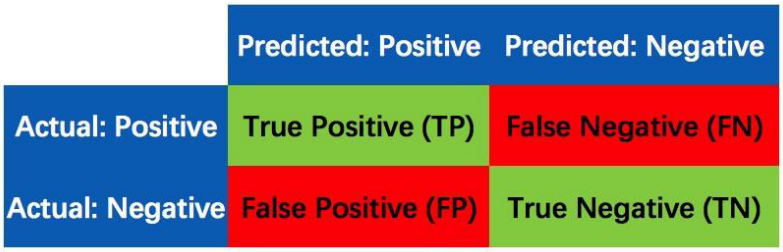

Figure 2.3: The confusion matrix used for measuring the prediction quality.

The most common equation for assessing models is ***accuracy***. It can be simply understood as the proportion of correct predictions. The equation for accuracy is shown as Equation (2.1).

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

However, the ***accuracy*** is not useful when the sizes of the two classes are very different. For example, if the correction rate of the positive set is much better than negative set and the positive set has a larger number of samples, the model also can give high ***accuracy***, however, it does not mean that the model truly gives a high performance. To fairly evaluate a model, Brain W. Mattews introduced a method[71] to assess prediction performance which takes into account the ratio of true and false positive samples and negative samples. When the sizes of two classes are very different, it can also be regarded as a balanced measure. The method was named the Matthews Correlation Coefficient (***MCC***).

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{2.2}$$

In addition, ***sensitivity*** and ***specificity***[72] are also employed to evaluate the prediction accuracy of a positive and a negative set, respectively. ***sensitivity*** measures the proportion of positive samples that are correctly predicted while ***specificity*** measures the proportion of negative samples that are correctly predicted. Both indications are mathematically expressed as:

$$sensitivity = \frac{TP}{TP + FN} \tag{2.3}$$

$$specificity = \frac{TN}{TN + FP} \tag{2.4}$$

The evaluation of model needs to consider a trade-off between these measures.

2.5 Normalization

In our research, the features of molecular data are described by different physicochemical properties, theoretical molecular properties and topological fingerprints. For solving the tasks of machine learning, molecular descriptors which are of different types need to be normalized. Thus, the range of different feature values possibly falls into a large scale so that in a classification task, the features with initially larger ranges outweigh features with initially smaller ranges. The function of normalization is to avoid this situation. There are various normalization methods. In this study, the Z-score was employed to normalize the features. The Z-score is also called as standard-score. The Z-score is expressed as Equation (2.5).

$$x_d^*(n) = \frac{x_d(n) - \langle x_d \rangle^{training}}{\sigma_d^{training}} \tag{2.5}$$

Where $\langle x_d \rangle^{training}$ represents the average value of d^{th} feature in all samples of training set. $\sigma_d^{training}$ denotes the standard deviation of d^{th} feature in all samples of training set. $x_d(n)$ is the original value of d^{th} feature in n^{th} sample.

$$\langle x_d \rangle^{training} = \frac{1}{N^{training}} \sum_{n=1}^{N^{training}} x_d(n) \tag{2.6}$$

$$\sigma_d^{training} = \left\langle (x_d - \langle x_d \rangle^{training})^2 \right\rangle^{1/2} \tag{2.7}$$

Here, the $N^{training}$ means the number of samples in the training set.

The absolute value of $x_d^*(n)$, Equation (2.5), represents the distance between the original value and the population mean in units of the standard deviation. When the original value, $x_d(n)$, is below (above) the population mean, the Z-score value is negative (positive).

In the Kaggle competition project, the dataset had been pre-processed before downloading. The normalization method for that is the min-max method which brings all original values into the range between 0 and 1. This method performs a linear transformation on the original data. Equation (2.8) is the min-max normalization.

$$x_d^*(n) = \frac{x_d(n) - x_d^{\min}}{x_d^{\max} - x_d^{\min}} \ , \tag{2.8}$$

Where $x_d^{\max}$ is the maximum and $x_d^{\min}$ the minimum value of d^{th} feature in the dataset.

2.6 Linear classifier

In my doctoral thesis, a linear classifier is the key to built prediction models for drug classification. The linear classifier is a statistical method in the machine learning field whose basic classification decision depends on the value of linear combinations of features which numerically characterize samples in a mathematical space. In this mathematical space, a given dataset involves ***N*** samples with ***D*** dimensions (features) for each sample. The samples of the dataset are linearly separable by a (***D*-1**)-dimensional hyperplane. In the frame of Bayes prediction scheme, the linear classifier is a suitable algorithm for operating classification tasks for which the samples belonging to different classes have equal probability distribution[73,74].

2.6.1 Linear discriminate function

A given dataset including ***N*** samples can be represented by Equation (2.9).

$$(\vec{x}_n, y_n),\ n=1,2,3,....N. \tag{2.9}$$

In our cases, a sample represents a compound. The $\boldsymbol{n^{th}}$ compound can be denoted as $\vec{x} \in \mathbb{R}^D$. Each compound is described by ***D*** features. $y_n \in \{+1, -1\}$ is the class label that represents for instance the biological activity of the compound (i.e. binding with a protein or not). For a binary classification problem, y_n can be +1 and -1 denoting positive and negative compounds, respectively. Based on the above definitions, a linear discriminate function is given by Equation (2.10). It is also called scoring function.

$$f(\vec{x}_n;\ \vec{w}, b) = \vec{w}^t \cdot \vec{x}_n + b \tag{2.10}$$

where $\vec{x} \in \mathbb{R}^D$ is the weight vector and ***b*** is the bias[74]. They involve the parameters to be optimized, using an objective function. The dot "•" in the above equation denotes the inner (scalar or dot) product between two vectors which is defined by Equation (2.11).

$$\vec{w}^t \cdot \vec{x}_n = \|\vec{w}\| \|\vec{x}_n\| \cos(\theta), \tag{2.11}$$

where $\|\vec{w}\|$ and $\|\vec{x}_n\|$ denote the Euclidean norm of $\vec{w}$ and $\vec{x}_n$, respectively and $\boldsymbol{\theta}$ denotes the angle between $\vec{w}$ and $\vec{x}_n$. If $\boldsymbol{\theta}$ is smaller (larger) than 90 degrees, the value of the dot product is positive (negative). If the dot product is 0, it means the $\boldsymbol{\theta}$ is 90 degrees, which represents $\vec{w}$ is perpendicular to $\vec{x}$.

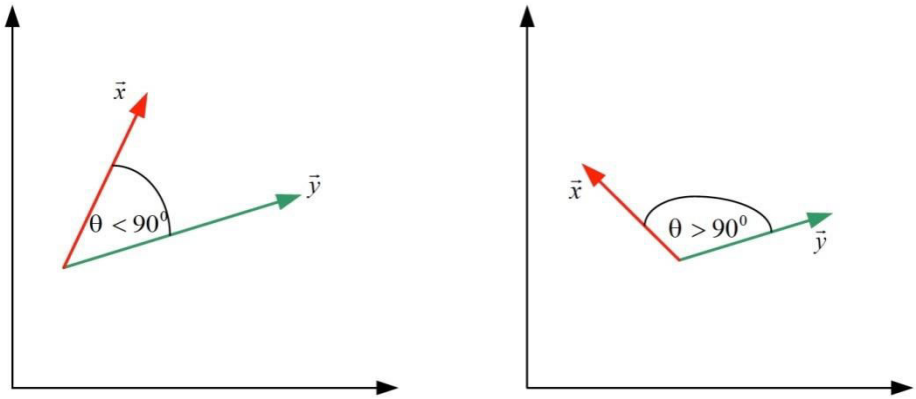

Figure 2.4: The inner product between $\vec{w}$ and $\vec{x}$ is defined by Eq. 2.10. Left side: If ***θ*** is less than 90 degrees, the value of the inner product is positive. As shown here, both the **red arrow** and the **green arrow** are in the same direction. Right side, if ***θ*** is less than 90 degrees, the value of the inner product is negative. Both arrows are in different direction. Figure source: Özgür Demir, Classification of Drug Molecules, Master thesis, Freie Universität Berlin. 2007.

The calculation of inner product is done by summing element-wise the products of the components of the two vectors as defined by Equation (2.12).

$$\vec{w}^t \cdot \vec{x}_n = \sum_{d=1}^{D} w(d) \times x_n(d) \ , \tag{2.12}$$

where ***n*** denotes the ***n^th^*** compound of the dataset, ***D*** denotes the number of features and ***d*** denotes the ***d^th^*** feature.

To separate objects in a multi-dimensional space, a hyperplane can be used, defined by $f(\vec{x}_n;\vec{w},b) = 0$ and $\vec{w}^t \cdot \vec{x}_n$ = **-*b***. The orientation of the hyperplane is determined by the vector $\vec{w}^t$ being the hyperplane normal vector. The hyperplane divides the multi-dimensional feature space, $\mathbb{R}^D$, into two sub-spaces, $\mathbb{R}_1^D$ and $\mathbb{R}_2^D$. The distance between a vector $\vec{x}_n$ and the hyperplane is given by Equation (2.13).

$$r_n = \frac{f(\vec{x}_n;\vec{w},b)}{\|\vec{w}\|} \tag{2.13}$$

Based on Equation (2.13), the distance between the origin of the coordinate system and the hyperplane is given by Equation (2.14).

$$r_{ori} = \frac{b}{\|\vec{w}\|} \tag{2.14}$$

For a binary classification problem, the compound vector $\vec{x}_n$ can be classified into the positive set if $f(\vec{x}_n;\vec{w},b) > 0$, while it is classified to belong to the negative set if $f(\vec{x}_i;\vec{w},b) < 0$. In the special case, where $f(\vec{x}_n;\vec{w},b)$ is exactly zero the compound is

assigned to both classes. In other words, the compound is exactly on the hyperplane. The decision rule can be expressed as Equation (2.15).

$$sign(f(\vec{x}_n;\vec{w},b)) \begin{cases} >0 & \text{if } \vec{x}_n \text{ belongs to the positive set} \\ <0 & \text{if } \vec{x}_n \text{ belongs to the negative set} \\ =0 & \text{if } \vec{x}_n \text{ belongs to both sets} \end{cases}. \tag{2.15}$$

For convenience, we simplify Equation (2.10) to include the bias parameter ***b*** into the vector $\vec{w}$. In order to achieve this purpose, the number of dimensions weight the vector $\vec{w}$ and the feature vector $\vec{x}$ are increased by one component according to Equation 2.16 and 2.17.

$$\vec{w} = \begin{pmatrix} w_1 \\ \vdots \\ w_d \end{pmatrix} \Rightarrow \underline{\vec{w}} = \begin{pmatrix} w_1 \\ \vdots \\ w_d \\ b \end{pmatrix} \tag{2.16}$$

$$\vec{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_d \end{pmatrix} \Rightarrow \underline{\vec{x}} = \begin{pmatrix} x_1 \\ \vdots \\ x_d \\ 1 \end{pmatrix} \tag{2.17}$$

Based on Equation (2.16) and (2.17), the Equation (2.10) can be simplified yielding Equation 2.18.

$$f(\vec{x}_n;\vec{w},b) = \vec{w}^t \cdot \vec{x}_n + b \Leftrightarrow f(\underline{\vec{x}}_n;\underline{\vec{w}}) = \underline{\vec{w}}^t \cdot \underline{\vec{x}}_n \tag{2.18}$$

Through training with the given training set, the both parameters can be determined and then unknown compounds can be predicted with Equation (2.10). However, to find these optimal parameters, we need to define an objective function. The next section will describe how to obtain the optimal parameters based on a given training set.

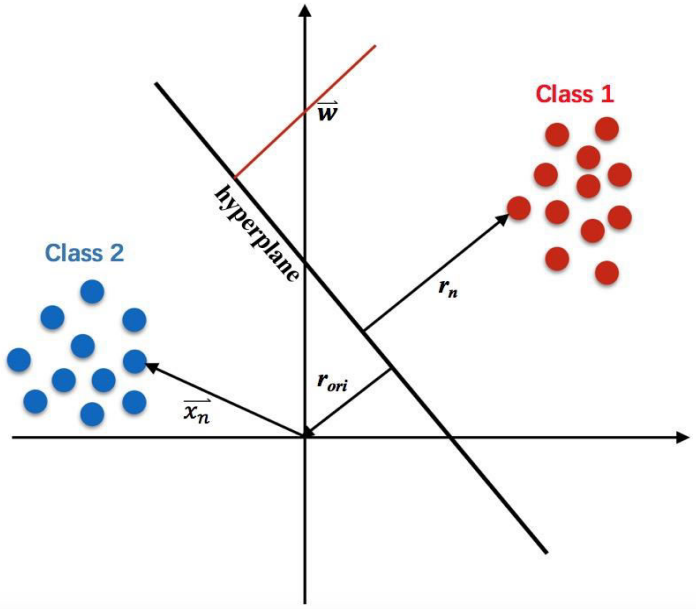

Figure 2.5: $f(\vec{x}_i;\vec{w},b) = 0$ describes a hyperplane with normal vector $\vec{w}$ and bias ***b***. In a binary classification, the hyperplane separates multi-dimensional space into 2 sub-spaces. Blue spots belong to class 2 and red spots belong to class 1.

2.6.2 Training

To determine the parameter vector $\vec{w}$, the model needs to be trained. For this purpose all compounds available in the training set (see Chapter 2.1) are considered. The linear classifier corresponds to supervised learning, since the compounds in the training set have known class labels (y_n = 1 or -1). Theoretically speaking, the basic assumption of the training process is that a new unknown compound should be from the same probability distribution as the training set so that the new compound can be classified correctly.

For a linear classifier, the aim of the training phase is based on the compounds with known activity to define a hyperplane, which can separate the training set into a positive and a negative sample sub-space. For a new compound the unknown activity is predicted according to which sub-space it falls into. Hence, compounds with unknown activity can be predicted based on the hyperplane by the following rules:

$$\begin{aligned} y_n &= 1 \quad \text{if} \quad f(\vec{x}_n;\vec{w}) > 0 \\ y_n &= -1 \text{ if} \quad f(\vec{x}_n;\vec{w}) < 0 \end{aligned} \tag{2.19}$$

or can be written as

$$y_n \times sign(f(\vec{x}_n;\vec{w})) > 0, \quad \text{for n} = 1...N\,. \tag{2.20}$$

To obtain the optimal parameter vector it is necessary to minimize the training error, which can be defined as Equation (2.21).

$$Error = \frac{1}{N}\sum_{n=1}^{N}\left| f(\vec{\underline{x}}_n;\vec{\underline{w}}) - \vec{y}_n \right| \tag{2.21}$$

However, this simple condition may not yield a unique solution. In other words, more than one hyperplane can meet this criterion. The possible solutions form a solution set. This problem may severely influence the prediction results on the test set because usually, the distribution of compounds in test set should be similar to the training set but not completely identical. A hyperplane randomly picked up from the solution set, which can correctly separate samples in the training set, possibly fails to properly classify compounds in a test set. Therefore, a proper hyperplane needs to be closer to the middle position of the solution set. An optimal hyperplane is expected to have a good generality so that it is capable of correctly classifying compounds with unknown activity. To achieve this purpose, more constraints need to be added to restrict the solution set.

A positive parameter, $\boldsymbol{m}$ is introduced to restrict the distance from a compound to the hyperplane in the space. This parameter requires the hyperplane not only to correctly classify the compounds in the training set but it also requests that the compound has a minimum distance to hyperplane, yielding a more rigorous training concept. In the following, Equation (2.22) and (2.23) contain this margin condition to restrict the number of possible hyperplanes, by redefining Equation (2.19) and (2.20).

$$\begin{aligned} \mathrm{y}_n &= 1 \quad \text{if} \quad f(\vec{\underline{x}}_n;\vec{\underline{w}}) > \quad m_n \\ \mathrm{y}_n &= -1 \text{ if} \quad f(\vec{\underline{x}}_n;\vec{\underline{w}}) < -m_n \end{aligned} \tag{2.22}$$

or it can be written as

$$y_n \times sign(f(\vec{\underline{x}}_n;\vec{\underline{w}})) > m_n, \quad \text{for n} = 1...N\,. \tag{2.23}$$

With this new condition, the distance from the hyperplane to compounds in the training set needs to be at least $m_n / \left\|\vec{\underline{w}}\right\|$. Since one can maximize $m_n / \left\|\vec{\underline{w}}\right\|$ with a minimum value for $\left\|\vec{\underline{w}}\right\|$, it means we only need to search an optimal $\left\|\vec{\underline{w}}\right\|$, i.e. to minimize $\left\|\vec{\underline{w}}\right\|$. Therefore, without loss of generality of the model, the parameter $\boldsymbol{m_n}$ can be set to 1 for all compounds.

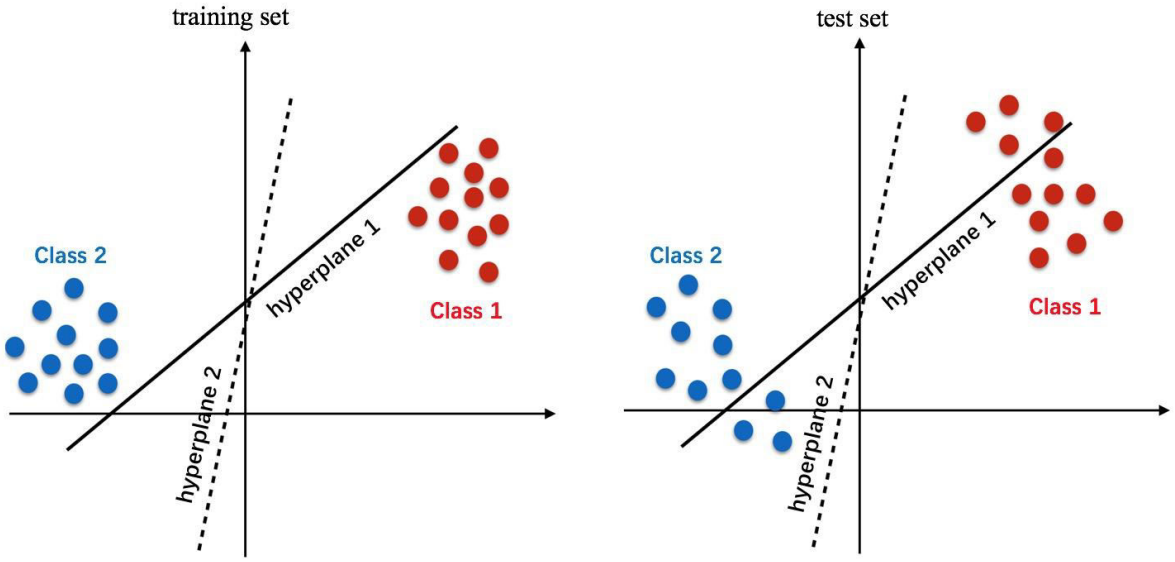

Figure 2.6: For the training set (right side), both hyperplanes 1 and 2 are in the solution set, which can properly separate class 1 and 2. However, on the test set (left side) the hyperplane 1 cannot perfectly separate class 1 and 2, but hyperplane 2 still can properly separate the test set. Evidently, hyperplane 1 is closer to the points of the two classes, leading to a worse generality than hyperplane 2.

2.6.3 Objective function

As mentioned above, when seeking an optimal parameter vector $\vec{w}$, an objective function need to be constructed. Minimizing the objective function one can gain a solution to the set of linear inequalities (Equation 2.23). Usually, a gradient descent algorithm is utilized to minimize the objective function. For my projects, rProp, a gradient descent algorithm is employed to solve the objective function (see Chapter 2.7 for details this algorithm). Here, it needs to be noted that the easiest choice for the objective function would consist of minimizing the number of misclassified compounds. However, this function is a piecewise constant where a gradient descent algorithm would fail to solve it. Therefore, before performing gradient descent, we reset Equation 2.23 to Equation 2.24, which makes the distance from a given compound to the hyperplane exactly equal to $m/\|\vec{w}\|$.

$$y_n \times sign(f(\vec{x}_n;\vec{w})) = m, \quad \text{for n} = 1...N \tag{2.24}$$

In most cases, a solution for Equation 2.24 is nonexistent. Hence, the optimal parameter vector $\vec{w}$, needs to be approximated by introducing a loss function.

2.6.3.1 Loss function

The loss function measures the difference between $y_n \times sign\left[f(\vec{x}_n;\vec{w})\right]$ and $\boldsymbol{m_n}$. Here, when $\boldsymbol{m_n = m}$ if $\boldsymbol{y_n} = 1$ and $\boldsymbol{m_n} = \boldsymbol{-m}$ if $\boldsymbol{y_n} = -1$. Minimizing loss function can approximate the

solution as good as possible. For the binary classification task, an easy and natural selection for a loss function is the 0-1 loss function which would return the value of 0 if a test compound is predicted correctly while returning 1 for a wrong classification. However, 0-1 loss function is not a continuous function and therefore not differentiable. Therefore, a continuous, convex loss function is used replacing the 0-1 loss function. In the following, several tractable and common loss functions are listed:

1. Mean-squared error (MSE)

$$g(f(\underline{\vec{x}}_n;\underline{\vec{w}}),m_n)=(f(\underline{\vec{x}}_n;\underline{\vec{w}})-m_n)^2 \tag{2.25}$$

2. Hinge loss

$$g(f(\underline{\vec{x}}_n;\underline{\vec{w}}),m_n)=\max(0,\ 1-m_n\times f(\underline{\vec{x}}_n;\underline{\vec{w}})) \tag{2.26}$$

3. One sided log Lorentzien (1slL)

$$g(f(\underline{\vec{x}}_n;\underline{\vec{w}}),m_n)=\begin{cases}\ln(f(\underline{\vec{x}}_n;\underline{\vec{w}})-m_i)^2+1 & ,\text{if } f(\underline{\vec{x}}_n)\square m_n<0\\ 0 & ,\text{else}\end{cases} \tag{2.27}$$

4. binomial negative log-likelihood (Bnll)

$$g(f(\underline{\vec{x}}_n;\underline{\vec{w}}),m_n)=\ln\left[1.0+\exp\left(\frac{1}{q(\vec{f}_n,\vec{w},b)\square m_n}\right)\right] \tag{2.28}$$

Those loss functions listed above are all continuous function which can be used for obtaining the optimal solution with gradient descent algorithm. During the training phase, the sum of loss function values for all training compounds constitute objective function. Take MSE as an example, the objective function would be as follows:

$$L(\underline{\vec{w}})=\sum_{n=0}^{N}(f(\underline{\vec{x}}_n;\underline{\vec{w}})-m_n)^2 \tag{2.29}$$

The minimization can be performed using gradient descent algorithm for which the gradient is given by differential form (see Equation (2.30)).

$$\frac{\partial L(\underline{\vec{w}})}{\partial \vec{w}}=\sum_{n=0}^{N}2(\underline{\vec{w}}\cdot\underline{\vec{x}}_n-m_n)\vec{x}_n \tag{2.30}$$

Then the resulting optimal solution $\underline{\vec{w}}$ can be used to predict the property of compounds with unknown activity by Equation (2.10).

2.7 Gradient descent algorithm

The gradient descent algorithm can be used to minimize the objective function yielding optimal model parameters $\underline{\vec{w}}$. The gradient descent algorithm works iteratively. Usually, a simple gradient starts with arbitrary initial weight values $\underline{\vec{w}}_1$. The gradient descent algorithm intrinsically has a predetermined step size. For j^{th} iteration, $\underline{\vec{w}}_j$ is obtained by moving a step in

direction of the negative gradient vector. The process of this iteration does not stop until the length of the step falls below a certain threshold. Alternatively, the stop condition of gradient descent can be defined as the condition that the absolute difference between two consecutive values of the objective function falls below a given threshold. This can be used as a convergence criterion. The selection of the convergence criterion is important, and needs to be made carefully. A larger value easily leads to a poor prediction whereas a small threshold value possibly causes the calculation not to be convergent. On the other hand, the gradient descent algorithm also needs to consider setting the initial step size. If the initial step size is chosen too small the convergence will take too much time. If the step size chosen is too large, the algorithm will oscillate around the minimum. Currently, there are several efforts to solve this problem in the basic gradient descent algorithm.

In my doctoral work, an excellent gradient descent algorithm, resilient propagation (Rprop)[75] was employed to operate gradient descent. Rprop It is a very fast method to run gradient descent. There are three main improvements over basic gradient descent algorithms. (1) During the optimization process, the step size is not fixed. Rather, it is updated, referring to the result of the last iteration. If the projection of gradient vectors of subsequent iterations is negative, it means that the minimum has been traversed in the last step. Thus, the step size is decreased by a given factor, $\boldsymbol{\eta^-}$. On the other hand, if the projection is positive, the step size is increased by another factor, $\boldsymbol{\eta^+}$. (2) The change of step size for each component is calculated independently. This enables the algorithm to find the minimum more efficiently.

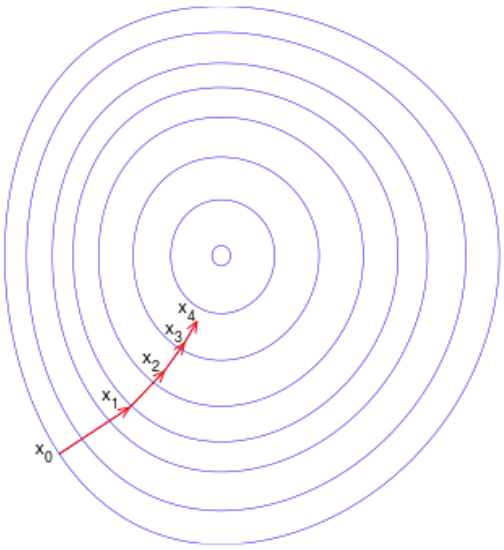

Figure 2.7: Illustration of gradient descent on a series of iteration steps. If $\boldsymbol{L(X_n)}$ is an objective function with a number of features, then $\boldsymbol{L(X_0) \geq L(X_1) \geq L(X_2) \geq L(X_3) \geq L(X_4)}$. Original figure from Wikipedia.

(3) Only the direction of the gradient is taken into account. This prevents the algorithm from slowing down as the step size gets smaller, since the gradient gets smaller when approaching the minimum. The iteration rules of the rProp algorithm can be formally described as follows:

$$w_d^{j+1} = w_d^j + \gamma_d^j \bullet sign(\Delta E_d^j), \text{ with } \gamma_d^1 = 0.001$$

$$\gamma_n^{j+1} = \begin{cases} \min(\gamma_d^j \bullet \eta^+, \gamma_{\max}) \text{ if } \Delta E_d^j \bullet \Delta E_d^{j-1} > 0 \\ \max(\gamma_d^j \bullet \eta^-, \gamma_{\min}) \text{ if } \Delta E_d^j \bullet \Delta E_d^{j-1} < 0 \\ \gamma_d^j \qquad \text{else} \end{cases} \tag{2.31}$$

where w_d^j denotes the d^{th} component (weight) of the rProp optimized gradient at the j^{th} step of the optimization process. γ_d^j denotes the step size for the d^{th} component (weight) at the j^{th} step of the optimization process. $\gamma_{\max}$ and $\gamma_{\min}$ represent the upper and lower step size limits, respectively. ΔE_d^j is the d^{th} component of the gradient of the objective function at the j^{th} step. The parameters used in rProp are given empirically: $\gamma_{\max} = 50$, $\gamma_{\min} = 10^{-6}$, $\eta^- = 0.5$ and $\eta^+ = 1.2$.

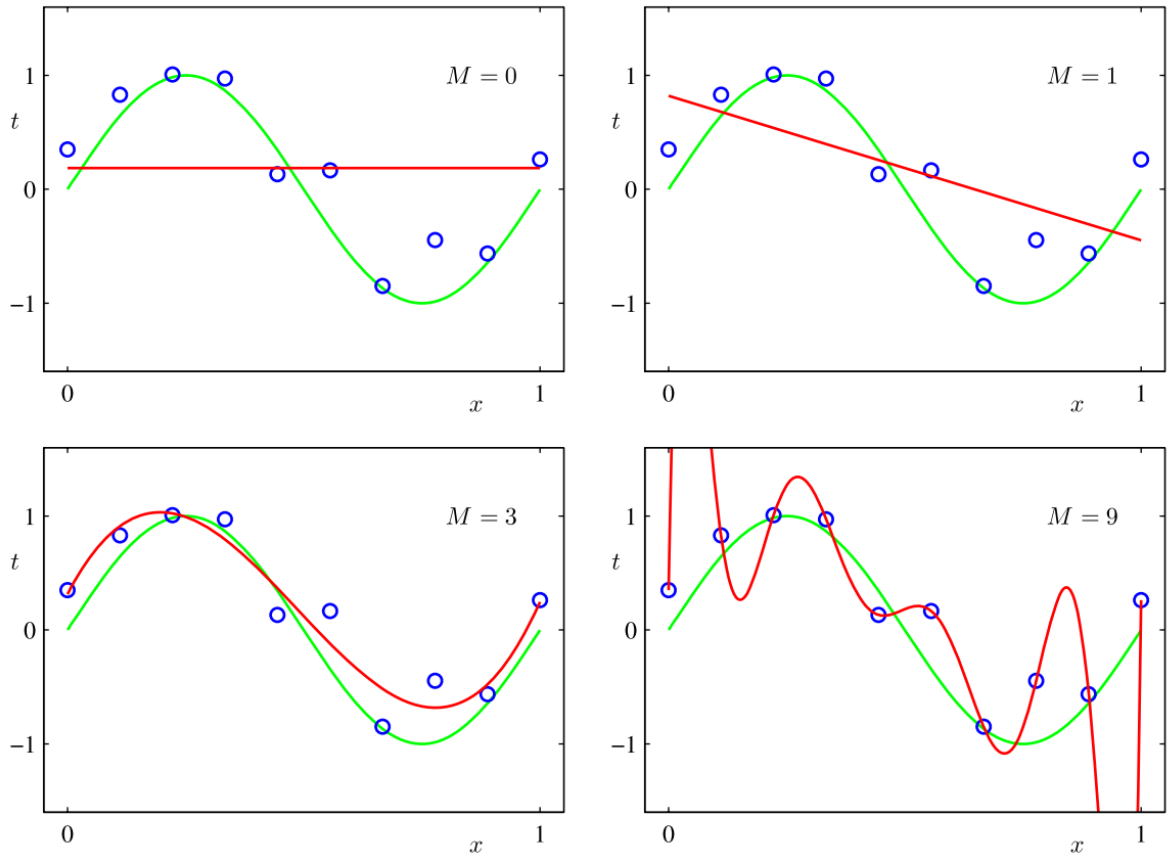

Figure 2.8: A polynomial is employed to fit the ten **blue points**. The **red curve** is the fitting model of the polynomial. M is the degree of the polynomial. For M = 9, the curve becomes very complex. The curve fits the given **blue points** perfectly, but yields a poor approximation

between the points. This is a typical overfitting phenomenon. When M = 0 & 1, the **red curve** does not fit the blue points appropriately. When M=3, the **red curve** may generalize better to more points drawn from the underlying unknown probability distribution. **Green curve** is the perfect fitting curve. Figure source: Christopher M. Bishop, Pattern Recognition and Machine Learning. Springer, 2006.

2.8 Regularization

In machine learning, the process of training is to fit a model to a given training set. This training set is expected to be representative enough for the probability distribution of the targets to be predicted so that the test set is covered by the sample space. However, if the model overreacts to minor fluctuations in the training set, overfitting could occur. Overfitting means the model specializes for the training set so perfectly that it looses the ability to generalize. As a consequence, the prediction performance on a new dataset, or say test set is poor because new unknown compounds can be similar but not identical with compounds in the training set. To improve the robustness and generalization of the prediction model, overfitting needs to be avoided. As demonstrated by Figure 2.8, a polynomial algorithm is used as an example to demonstrate overfitting.

Regularization is a method to reduce overfitting in order to improve the generalization of a prediction model. For this purpose an additional term is added to the objective function. During the training phase, the constraints (see Equation (2.22) & (2.23)) are relaxed allowing for more compounds in training set to be misclassified. Thus, regularization term penalizes unnecessary complexity of the model to improve the generality of the model. In the studies of my doctoral projects, two types of regularization were employed. These are Lasso regularization (L1)[57] and ridge regression regularization (L2)[78].

2.8.1 Lasso Regularization

Lasso is a commonly used regularization[57], also called L1 regularization. During modeling, Lasso regularization adds the sum of the absolute values of features to the objective function. Lasso has the inherent linear dependence on input features of the model so that it disables irrelevant features leading to sparser sets of features. Due to the linear form of Lasso regularization, shown as Equation (2.32), the weights of many features contributing weakly to prediction are rigorously set to zero. Thereby Lasso regularization can be an embedded method, efficiently operating feature selection. The regularization parameter λ_1 is a multiplier to determine the degree of regularization. The larger λ_1 value is, the more features whose

weights are set to zero. Usually, before modeling, several λ_1 would be given manually, using a validation dataset to determine the optimal value.

$$\lambda_1 \left\| \vec{w} \right\|_1 = \lambda_1 \sum_{d=1}^{N} \left| w_d \right| \ , \tag{2.32}$$

Where w_d represents the weight value of the d^{th} feature and N is the number of features.

Lasso regularization is not differentiable whenever weights of features are set to zero so that a special solver is required to solve the objective function embedded with the Lasso regularization term. Currently, several solvers[76-78] are available to minimize the objective function with Lasso regularization term. In my project, Orthant-Wise Limited-memory Quasi-Newton algorithm (OWL-QN), developed by Andrew *et al.*[77] was employed to solve L1 regularization, which has been implemented in DemPred software package[79]. The OWL-QN algorithm utilizes a condition that for a given orthant (half-space) of the function space, a differentiable objective function added with L1 term is again differentiable.

2.8.2 Ridge regression Regularization

Ridge regression regularization is closely related to Lasso regularization[78], which is also used to reduce overfitting during modeling. This regularization is also called L2 regularization. In contrast to Lasso regularization, Ridge regression is milder. L2 adds the sum of square of model parameters to the objective function (see Equation (2.33)).

$$\lambda_2 \left\| \vec{w} \right\|_2^2 = \lambda_2 (\sum_{d=1}^{N} \left| w_d \right|^2) \ , \tag{2.33}$$

where w_d represents the weight value of the d^{th} feature and N the number of features.

Since the quadratic form of Ridge regression regularization cannot set the weights of features rigorously to zero during optimization, ridge regression regularization is not capable of reducing the number of features directly. However, with increasing λ_2, the weights of irrelevant or redundant features are decreased. In other words, the influence of these features is diminished. λ_2 can be determined with a validation dataset. In my projects, the L1 and L2 regularization cannot be optimized simultaneously. The objective function combined with the L2 regularization term is minimized with the Rprop algorithm[75,76].

2.9 DemPred

Previously, our group developed a machine learning software package named DemPred[79]. The aim of this software is to help people understand underlying biochemical processes as well as speed up the detection of new active drug compounds for research targets.

The implementation of state-of-the-art prediction methods requires a great deal of expertise and time. Hence, a device with high performance yet easy to use is needed. DemPred was developed to satisfy this demand. In addition, DemPred can be extended to build own models for a particular prediction task.

Currently, DemPred had been successfully applied to various tasks such as prediction of peptides binding to the major histocompatibility complex II (MHC II)[80], prediction of human volume of distribution and clearance[81] and the prediction of protein decoys[82]. In most cases, the generated model yields results that were as good as or even better than those of state-of-the-art prediction techniques at the time of development employed by other groups. DemPred is an object-oriented software package. It can be used to deal with various biological problems. Depending on the problem that needs to be solved, a specific object function is predefined and implemented for handling.

The objective function of DemPred is given as Equation. (2.34).

$$\begin{aligned} L(\underline{\vec{w}}) = {} & W^{+}\frac{(1-\sum_p \lambda_p)}{N^{+}}\sum_{n=1}^{N+}\{g(f(\vec{x}_n,\underline{\vec{w}}),y_n)\}+ \\ & W^{-}\frac{(1-\sum_p \lambda_p)}{N^{-}}\sum_{n=1}^{N-}\{g(f(\vec{x}_n,\underline{\vec{w}}),y_n)\}+\sum_p \lambda_p \parallel \vec{w} \parallel_p, p=1,2 \end{aligned} \tag{2.34}$$

where $g(f(\vec{x}_n,\underline{\vec{w}}),y_n)$ represents the loss function and $\boldsymbol{y_n}$ is the biological response value. The additional term $\sum_p \lambda_p \parallel \vec{w} \parallel^p$ is a regularization term that is used to reduce overfitting. $\boldsymbol{p}$ = 1 or 2 denotes the L1 or L2 regularization, respectively. $\boldsymbol{N}$ is the total number of compounds in the training set.

2.10 DemFeature

DemFeature is also an *in silico* model, which is further developed based on DemPred. The core idea of DemFeature is to constitute a specific training subset for a specific test compound. Certainly, the source of the training subset is from the original training set. In the training phase, this method ignores compounds of the training set that are not sufficiently similar or dissimilar to the compound to be classified. Based on this strategy, ideally, the considered sample can be more accurately interpreted based on the feature values.

The workflow of DemFeature is illustrated in Figure 2.9. Step 1: A compound $\boldsymbol{k}$ to be predicted is randomly picked up from the test set; Step 2: This picked compound $\boldsymbol{k}$ needs to be measured for similarity with all compounds in training set; Step 3: The similarity value of compound $\boldsymbol{k}$ with compounds of the training subset is the criterion to compose a training

subset for compound ***k***; Step 4: The selected training subset is used to construct a specific model to predict molecule ***k***. Thus, a specific prediction model is designed to each molecule in the test set.

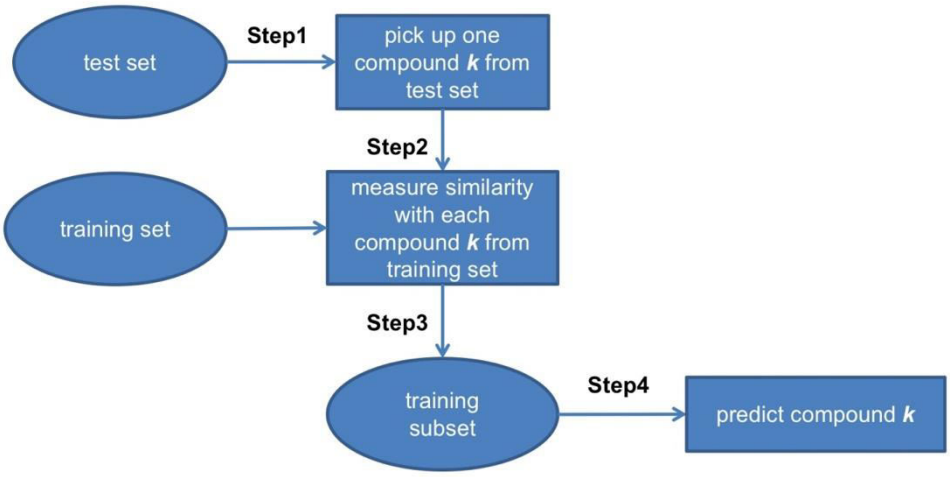

Figure 2.9: Basic workflow of DemFeature. DemFeature-1 and -2 have the same work model. The difference between them is the method for constituting a training subset.

For the definition of similarity, in DemFeature, a new parameter ***s***(***k***, ***n***) is introduced, which accounts for the similarity between the sample ***k*** to be predicted and a compound ***n*** in training set. Before calculating ***s***(***k***, ***n***), the participating feature vectors need to be normalized following Equation. 2.35.

$$\hat{\vec{x}}_n = \vec{x}_n / \sqrt{\vec{x}_n \cdot \vec{x}_n}\ , \tag{2.35}$$

where $\vec{x}_n$ is the feature vector of the compound ***n*** in training set. The compound ***k*** in a test set also needs to be normalized in this way. With the normalized feature vectors the similarity ***s***(***k***, ***n***) is defined by Equation. 2.36.

$$s(k,n) = \hat{\vec{x}}_k \cdot \hat{\vec{x}}_n \quad s(k,n) \in [-1.0, 1.0] \tag{2.36}$$

$s(k,n)$ measures the similarity between the compounds ***k*** and ***n***. The range of this value is between -1 and 1. The closer this value is to 1.0, the more similar both compounds are.

2.10.1 DemFeature-1

The definition of the similarity between two compounds can determine how to select a training subset to predict a compound ***k*** in test set. However, the $s(\boldsymbol{k}, \boldsymbol{n})$ itself cannot decide the size of the training subset. The aim of DemFeature is to select a training subset according to the compound ***k*** to be predicted. If the size of training subset is too large, relatively dissimilar compounds could influence the correctness of prediction and also increase CPU run

time. On the other hand, if the size of the training subset is too small, the diversity of training set may be so low that prediction performance goes down. Therefore, for DemFeature-1, the parameters ***cutoff*** ∈ [-1, +1] are given to control the size of the training subset. For example, if $s(\boldsymbol{k}, \boldsymbol{n}) > \boldsymbol{cutoff} =$ -0.2, then only the compounds ***n*** in the training set fulfilling $s(\boldsymbol{k}, \boldsymbol{n}) >=$ -0.2, would be selected to predict compound ***k***.

Based on the objective function of DemPred, a new objective function $L_k(\vec{w}_k)$ of DemFeature-1 is defined as below:

$$\begin{aligned} L(\vec{w}_k) = W^+ \frac{(1-\lambda_2)}{N^+} \sum_{n^+=1}^{N_{CUT}^+} \{ | s(k,n^+) | \times g(f(\vec{x}_{n^+}, \vec{w}_k), y_{n^+}) \} \\ + W^- \frac{(1-\lambda_2)}{N^-} \sum_{n^-=1}^{N_{CUT}^-} \{ | s(k,n^-) | \times g(f(\vec{x}_{n^-}, \vec{w}_k), y_{n^-}) \} + \lambda_2 \left\| \vec{w}_k \right\|_2^2 \end{aligned} , \quad (2.37)$$

where N_{CUT}^+ and N_{CUT}^- represent the size of the training positive subset and negative subset, respectively. $\boldsymbol{W}^+$ and $\boldsymbol{W}^-$ are weights for controlling the balance between the positive and negative set. $s(\boldsymbol{k}, \boldsymbol{n})$ is given by absolute value to multiply the loss function. $g(f(\vec{x}_n, \vec{w}), y_n)$ represents the loss function and $\boldsymbol{y_n}$ the biological response value. In this objective function $L(\vec{w}_k)$, the regularization term is the ridge regression regularization. The rProp algorithm is utilized to minimize the objective function to obtain the parameters, $\vec{w}_k$ for the compound ***k*** to be predicted.

2.10.2 DemFeature-2

Although DemFeature-1 introduces ***cutoff*** parameters to govern the size of the training subset of each compound to be predicted, it still cannot rigorously fix the size of the training subset. For example, if there is a training set containing 3000 compounds, $\boldsymbol{cutoff} =$ 0.0 could select 1500 compounds as the training subset for the compound ***k*** while compound ***k***+1 only has 300 compounds as a training subset. Thus, for the dataset having a huge number of compounds with a large number of features, it probably costs too much CPU run time. To solve this problem, DemFeature-2 was designed.

Based on the similarity between the compound to be tested and the compounds in the training set, the idea of DemFeature-2 is to select the 4 small sets from the original training set to constitute an even more specialized training subset for each test compound. These 4 small training sets are: (1) most similar positive compounds set, (2) most similar negative compounds set, (3) most dissimilar positive compounds set and (4) most dissimilar negative

compounds set. The number of compounds in those sets is given manually. It means the number of training subset is fixed. Thus, the size of training subsets can be restricted to control the CPU run time for modeling more precise. Moreover, this even simpler model can be easily interpreted to understand the relations between compounds and biological response.

As described above, the objective function of DemFeature-2 includes four parts, namely, the most similar positive set, the most similar negative set, the most dissimilar positive set and the most dissimilar negative set. The Equation (2.38) is the objective function of DemFeature-2 including an L2 regularization term.

$$\begin{aligned} L_k(\vec{w}_k) = & \frac{W_{sim}^{+}}{S_{sim}^{+}} \sum_{n_{sim}^{+}=1}^{N_{sim}^{+}} [\mid s(k, n_{sim}^{+}) \mid \times g(f(\vec{x}_{n_{sim}^{+}}, \vec{w}_k), y_{n_{sim}^{+}})] \\ & + \frac{W_{diss}^{+}}{S_{diss}^{+}} \sum_{n_{diss}^{+}=1}^{N_{diss}^{+}} [\mid s(k, n_{diss}^{+}) \mid \times g(f(\vec{x}_{n_{diss}^{+}}, \vec{w}_k), y_{n_{diss}^{+}})] \\ & + \frac{W_{sim}^{-}}{S_{sim}^{-}} \sum_{n_{sim}^{-}=1}^{N_{sim}^{-}} [\mid s(k, n_{sim}^{-}) \mid \times g(f(\vec{x}_{n_{sim}^{-}}, \vec{w}_k), y_{n_{sim}^{-}})] \\ & + \frac{W_{diss}^{-}}{S_{diss}^{-}} \sum_{n_{diss}^{-}=1}^{N_{diss}^{-}} [\mid s(k, n_{diss}^{-}) \mid \times g(f(\vec{x}_{n_{diss}^{-}}, \vec{w}_k), y_{n_{diss}^{-}})] + \lambda_2 \left\| \vec{w}_k \right\|_2^2 \end{aligned} \tag{2.38}$$

$$\begin{aligned} S_{similar}^{\pm} &= \sum_{n_{similar}^{\pm}=1}^{N_{similar}^{\pm}} \mid s(k, n_{similar}^{\pm}) \mid \text{ with } \; s(k, n_{similar}^{\pm}) > 0 \\ S_{dissimilar}^{\pm} &= \sum_{n_{dissimilar}^{\pm}=1}^{N_{dissimilar}^{\pm}} \mid s(k, n_{dissimilar}^{\pm}) \mid \; \text{ with } \; s(k, n_{dissimilar}^{\pm}) < 0 \end{aligned} \tag{2.39}$$

where for a training subset of the compound $\boldsymbol{k}$ to be predicted, N_{sim}^{+}, N_{diss}^{+}, N_{sim}^{-} and N_{diss}^{-} respectively represent the number of samples in the similar positive set, the similar negative set, the dissimilar positive set and the dissimilar negative set, respectively. W_{sim}^{+}, W_{sim}^{+}, W_{sim}^{+} and W_{sim}^{+} indicate the weights of each part for controlling the balance among the sets. $s(\boldsymbol{k}, \boldsymbol{n})$ is given by absolute value to multiply loss function. $g(f(\vec{x}_n, \vec{w}), y_n)$ represents the loss function and $\boldsymbol{y_n}$ the response value.

2.11 Quadratic features

The principle of quadratic features is to map the current model from a lower dimensional into a higher dimensional feature space by increasing the number of features. Quadratic features can be produced by using the original (linear) features to generate products of important

quadratic features and add them to the linear features. The important features set is actually a subset of linear features. The importance of features refers to the absolute values of the weights, $\underline{\vec{w}}$, which can be obtained by solving the objective function of DemPred (see Equation (2.34)). The larger the absolute values of the weights are, the more important are the corresponding features. Since the method of using the important features to generate quadratic features can produce a huge number of features, the number of the important features needs to be chosen carefully referring to the number of linear features. If the number of quadratic features is too large, it will take too much CPU time to train the prediction model. For the $\boldsymbol{n^{th}}$ compound in the dataset, the notation of the quadratic feature vector $\vec{f}^{\,q}$ is given by Equation (2.40).

$$\vec{f}^{\,q} = (f_1 f_1^i, f_1 f_2^i, \ldots\ldots f_1 f_I^i,\ f_2 f_1^i(2), f_2 f_2^i, \ldots\ldots f_2 f_I^i,\ \ldots. f_N f_1^i, f_N f_2^i, \ldots\ldots f_N f_I^i) \qquad (2.40)$$

where $\boldsymbol{I}$ denotes the number of important features and $\boldsymbol{N}$ is the total number of features in the dataset. Note that the values of the features depend on the molecule considered. In this thesis, quadratic features were combined with the linear features together to build the prediction model.

2.12 Comparison of the performance between two models

When several approaches are employed to solve a classification task, those approaches may show different results. Although statistical performance metrics such as MCC, accuracy, can be used to compare the differences among approaches, it may be impossible to reflect the statistical significance of those differences with those performance metrics. Therefore, to properly evaluate the difference between two approaches in a binary classification, the $\boldsymbol{p}$-value can be employed as a parameter to compare two prediction schemes to tell how much they differ[83].

Suppose predicted results generated by model **A** and **B** are based on the same dataset. Comparing both results to the correct response value, there are four situations shown in Figure 2.10.

model A correct model B correct a	**model A correct model B wrong b**
model A wrong model B correct c	**model A wrong model B wrong d**

Figure 2.10: Comparing prediction results of two models (A and B) with the correct value, there are four possible outcomes. **a**, **b**, **c** and **d** represent the count of four situations.

As shown above, if the probabilities that the models disagree (one model is correct the other wrong) are equal ($p_b = p_c$), both models (**A** and **B**) perform equally well and have the same accuracy. This is the null hypothesis (H_0: $p_b = p_c$), i.e. the two models are equivalent. We like to find out how much the prediction results of the two models need to differ that we can state that the two models are inequivalent. The McNemar's test[84] was employed to test the difference of the distributions in this research defined by Equation (2.41).

$$\chi^2 = \frac{(|b-c|-1)^2}{(b+c)} \tag{2.41}$$

Based on **b** and **c**, the McNemar's test can output a χ^2 value, which can be compared with the value from the χ^2 distribution that corresponds to a specific significance level. Alternatively, the binominal distribution[83] (see Equation (2.42)) can be used to calculate the ***p***-value based on **b** and **c**. However, this ***p***-value needs to be multiplied by 2 because a 2-sided test is used.

$$p = \sum_{b}^{b+c} \frac{(b+c)!}{b!(b+c-b)!}(0.5)^{(b+c)} \tag{2.42}$$

If the ***p***-value is very large, the two models are not significantly different. Conversely, the models are significantly different if the ***p***-value is very small. Usually, if more than 2 models need to be compared by ***p***-value, a matrix of ***p***-values can be prepared to compare them.

2.13 Confidence measure

Measuring the confidence of the prediction performance of a model is a way to assess the reliability. In this research, we used a method to calculate confidence corresponding to probability of correctness of outcome, which is employed to solve binary classification tasks. Our scoring function used in the study is the linear scoring function, $y = \vec{w} \cdot \vec{x}_n + b$, which is the scoring function of DemPred and is also used in our newly developed methods, DemFeature-1&-2. After the model parameters $\underline{\vec{w}}$ are determined, the scoring function can be used to predict unknown samples. In the training ***y*** adopts the idealized values +1 or -1 if the

considered compound belongs to the positive or negative set, respectively. Hence, if ***y*** is above 0.0, the compound is predicted to belong to the positive set, while for ***y*** below 0.0, the compound should belong to the negative set. The probability measures that a compound belongs to the positive or negative set is expressed as Equation (2.43).

$$p_{\pm} = \max\{0, \frac{1}{2}(1 \pm y)\} \tag{2.43}$$

However, the actual value of ***y*** differs from +1 or -1. As long as ***y*** is in the interval[-1,+1] the probabilities are properly normalized: $p_{-} + p_{+} = 1$. If the value ***y*** is outside of the interval, we normalize the probabilities, as follows:

$$\hat{p}_{\pm} = \frac{p_{\pm}}{p_{+} + p_{-}} \tag{2.44}$$

These normalized probabilities can be used to define a **confidence value**, Equation (2.45):

$$confidence_{\pm} = 2\hat{p}_{\pm} - 1 \ , \tag{2.45}$$

which provides the information of how probable a prediction is.

3 Applications

This chapter mainly describes the results obtained with our methods. It consists of two projects: This first predicts the datasets of the Kaggle™[53] competition and the second predicts the dataset of phospholipidosis[54]. The datasets of the Kaggle™ competition were used by many professional persons and groups so that it can be used to obtain realistic evaluations of the performances of our models. All datasets used in this competition can be downloaded from http://www.kaggle.com/c/bioresponse/data. Alternatively, phospholipidosis, a side effect caused by taking medicine, has elicited an increasing interest within the drug discovery community recently. This dataset will be used to verify whether our models can be applied in the early stage of drug development as a mean of reducing costs. For each project, both DemPred and DemFeature are used to predict the datasets. In addition, we used other models to compare the prediction performances with DemPred and DemFeature.

3.1 Project 1: Kaggle™ competition

In 2012, Boehringer Ingelheim Inc. launched a competition[53] on the Kaggle™, that is an online data mining platform on which people can submit datasets to launch a competition for obtaining the best model from participants all over the world. The aim of this competition is to investigate the utility of computational crowdsourcing in generating highly predictive models for use within the pharmaceutical industry. It is believed that the competitive dynamics of participants is more effective in driving optimized models than developing the powerful prediction models based on an objective measure of performance. The data provided for this contest were anonymized using not the clear names of features. Furthermore during the competition the specific property to be predicted was not revealed to participants until the end of the competition, in order to ensure a truly blind contest. In other words, these anonymous data were provided during the competition for the purpose of mitigating any possible influence from a pharmic expertise bias so as to guarantee a fair contest of the methods applied by the participants. Additionally, this competition also helps not only the organizer but also researchers in computer aided drug development to know the relations between machine learning skills and field expertise.

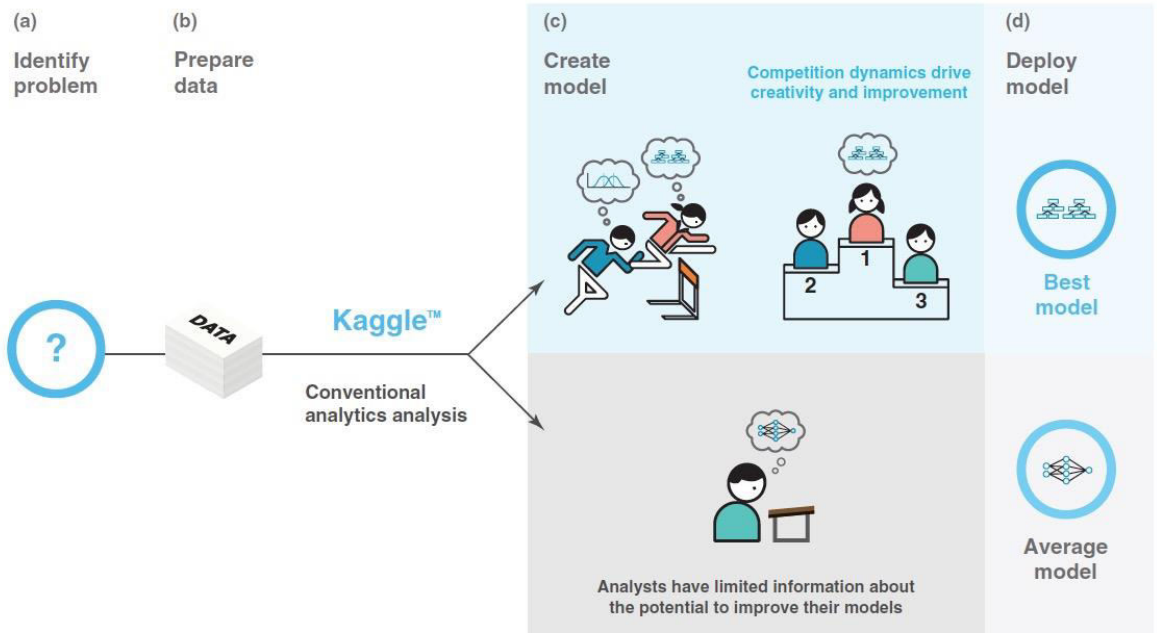

Figure 3.1: Two paths to construct a model to predict the target of interest. The lower path is the traditional way of building models by experts. The upper path is the crowd sourcing approach, which identifies the best model through competition. Original figure is taken from the article of Bentzien *et al*[53].

In this Kaggle™ competition, the organizers (Boehringer Ingelheim Inc., a pharmaceutical company) have chosen a biologically relevant target based on prior articles that describe the provenance of this biological data and many prior prediction models that were built and produced results. The reference to the prior biological prediction models is to enable a realistic expectation setting.

The purpose of using the dataset of the Kaggle™ competition to design the algorithm in this research is helpful for us to evaluate our models from a relatively objective perspective. However, it is inevitable that in our studies, the most successful model was selected through a comparison of the prediction performances of the different models while being fully aware of the results. In spite of the fact that we were trying to train our prediction model in a competition scenario, our prediction is not fully unbiased, since we know the final results while the participants of the Kaggle™ competition did not knew them before the competition terminated. Therefore, we do not claim that our prediction model is the truly better than the best models in the Kaggle™ competition.

3.1.1 Dataset

The dataset used in the Kaggle™ competition is the benchmark dataset for *in silico* prediction of Ames mutagenicity, as described by Hansen *et al.*[85]. The Ames test is the bacterial reverse mutation assay, which is used to detect mutagenicity *in vitro*. This method is an efficient and early alert mechanism of potential genotoxicity. Within drug discovery, genotoxicity is an important global property of high pharmaceutical relevance. On the other hand, this curated dataset is of high quality and is well known within the academic communities and many expertise investigations[86] have been done on this dataset, which is convenient for obtaining a direct comparison with expertise methods.

The original benchmark dataset of Ames comprises 6500 compounds. The dataset used in the Kaggle™ competition is an updated version consisting of 6512 compounds. The original compounds are represented as SMILES string. For calculating the descriptors of those compounds, they were converted into 3D molecules by the competition organizers, out of which 9 compounds failed to be converted due to the software limitations. A pipeline plot protocol[62] was employed to filter the remaining 6503 molecules. This protocol removed 251 compounds, including: ***1.*** compounds with non-zero formal charge. ***2.*** Molecules with more than 99 atoms. ***3.*** compounds with undesirable atoms of types D, B, P, Al, Ga, Si, Ge, Sn, As, Sb, Se, Te, At, He, Ne, Ar, Kr, Xe, Rn. Finally, after this screening, the remaining 6252 compounds comprised the dataset of the competition of which 3401 compounds were active (positive compounds) in Ames test while the rest of the 2851 compounds were inactive in the Ames test (negative compounds). The ratio of the positive set to the negative set is 1: 1.19, which is considered as a balanced dataset. In the competition, this dataset was randomly divided into three parts: training set, public test set and private test set (Table 3.1). During the run of the competition, all participants could only access the training and public test set.

Table 3.1: the datasets used in the Kaggle™ competition.

	# compd. whole dataset	**#compd. pos.**	**#compd. neg.**	**the ratio pos. & neg.**
training set	3751	2034	1717	1:1.18
public test set	625	329	296	1:1.11
private test set	1876	1038	838	1:1.24

Six software packages were used to prepare descriptors for those compounds. As mentioned above, the real names of these descriptors were hidden to participants during the competition. Table 3.2 shows the molecular descriptors used in this dataset. According to the article of the

organizers[53], they calculated a total of 5030 molecular descriptors. The absence of stereochemical information in the original dataset prevents that the 3-D molecular descriptors can be calculated. Moreover, the organizers also deleted 3253 molecular descriptors due to two reasons: ***1.*** 2537 descriptors with low variance (≤0.01), which was calculated using the standard deviation over all compounds in the dataset, and ***2.*** 716 descriptors showing high correlation (Caret high correlation filter >0.90)[53]. Finally all remaining 1776 features were normalized with the min-max method as defined by Equation (2.8) such that the feature values are in the interval [0, 1].

Table 3.2: the molecular descriptors used in the Kaggle™ competition.

descriptors class	# descriptors	# remaining descriptors
MOE2D[65]	186	76
MolConn-Z[57]	745	174
clogP[87]	1	0
CADDAP[88]	1920	696
pipeline Pilot[62]	130	5
daylight-FP[89]	2048	825
total	5030	1776

In addition to *in silico* molecular descriptors, the dataset also has a biological target to be predicted. For this target, +1 represent the positive property and -1 represent the negative property.

In this research, in addition to applying DemPred (see Chapter 2.9), we developed a new algorithm (DemFeature) to perform predictions for this dataset because it provides a realistic up-to-date prediction scenario for drug classification and the predictions submitted in this contest from different professional persons and groups may come close to the theoretical limits of what can be achieved for this prediction task. Hence, it is very useful to evaluate our models with respect to this dataset.

3.1.2 Results

This section depicts the prediction results of DemPred, DemFeature-1 and DemFeature-2 to predict datasets of the Kaggle™ competition. These models were built with the training set of the Kaggle™ competition. In addition to linear features, quadratic features (see Chapter 2.11) were also used to build the DemPred model. All datasets were automatically normalized by Z-score (Equation 2.5) in DemPred and DemFeature software packages. Moreover, in order to

evaluate the objectivity of these models, the results of ranked models of the Kaggle™ competition were also used as means of comparison with our methods. The results of these comparisons are also reported in this section.

3.1.2.1 Prediction results of DemPred

DemPred is an object-oriented prediction method developed by our group. For a detailed introduction of DemPred, refer to Chapter 2.9. In this study, The DemPred model was built with the training set of the Kaggle™ dataset, which includes 3651 compounds. Since an L1&L2 two-step[90] method was employed to construct the DemPred model to predict CoEPrA tasks[91] and achieved good prediction results, we also decide use this method to construct the DemPred model for the Kaggle™ dataset. It must be emphasized that the datasets in CoEPrA are typical biological datasets containing octo- and nona- peptides relevant to MHC class I binding, which play an important role in the immune response of mammals.

As the procedure mentioned in the publication of Demir *et al.*[90], DemPred combined with L1 feature selection was used in the first stage to predict the private and public test set of the Kaggle™ competition. As the L1 approach is able to precisely set the weights of the weak features to zero, it is able to control overfitting of model by feature selection. After feature selection through L1, the L2 regularization was employed to deal with the remaining features. The L2 approach is a quadratic term that cannot set the weight of features to zero, which means that L2 would not cause disappearance of features. The strength of L1 & L2 depends on the weight parameter λ in Equation(2.34). Therefore, choosing the appropriate λ value is important to govern the L1 & L2 approaches. In this study, several λ values were considered including $\lambda_1 \in$ {0.002, 0.003, ..., 0.02, 0.021, 0.022, 0.024, 0.026, 0.028, 0.03, 0.035, 0.04, 0.045 ,0.050}; $\lambda_2 \in$ {0.01,0.03,0.04, …, 0.2, 0.22, 0.25, 0.28, 0.3, …, 0.37, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8}. Usually, the appropriate value of this parameter is determined by applying it to a validation set (see Chapter 2.1). As mentioned above, the public test set could be accessed by all participants during course of the competition. Therefore, in order to have the same conditions as all participants, the public test set was also used to optimize parameters. The process of optimization is the most CPU time demanding step. The larger the λ value used, the more CPU time is needed.

In addition to linear features, quadratic features were also produced for the construction of the DemPred prediction model. In this study, the linear features are the original features of the Kaggle™ dataset. The quadratic features were produced based on the linear features. The

detailed descriptions of the quadratic features are presented in Chapter 2.11. The quadratic features are capable of mapping the dataset targeted onto a space of higher dimensions for classification.

3.1.2.1.1 Linear features prediction results

Choosing different loss functions (see Chapter 2.6.3.1) could have different effects on prediction results. In this research, we mainly used two loss functions in the construction of the DemPred model, namely, the Mean Squared Error (MSE) (Equation (2.25)) and one sided log-Lorentzian (1slL) (Equation (2.27)). Both loss functions were employed along with L1 or L2 terms to minimize the objective function. Table 3.3 shows ***MCC*** (Equation (2.2)) results of DemPred model with two different loss functions for the Kaggle™ dataset. To simulate the competition as closely as possible, the public test set was first used to optimize λ values. In addition, the λ values optimized by 10-cross validation using the training set are also used for comparison.

As shown in Table 3.3, at the stage of L1 feature selection, it can be clearly seen that very small λ_1 values have removed a large number of features. For the ability to delete features, the loss function 1slL makes L1 to display stronger effects than by using MSE. When the λ_1 value is just 0.004, the 1slL can remove near 2/3 of the features, causing a decrease from 1776 to 544 features. The prediction results of the private test set are better than of the public test set. Although the optimized λ values are based on the public test set, it does not lead to a measurable advantage for the prediction of the public test set to be better than that for the private test set. As clearly shown in Figure 3.2, the prediction performance of the private test set is always better than that of the public test set. This may be so, since the private test set was constructed to be easier predictable than the public test set[53].

Furthermore, at the second step, the L2 regularization was applied to the remaining features after L1 feature selection. However, this effectively does not improve the prediction performance. In fact, the prediction performance of DemPred built with MSE declines after regularization with the L2 approach. In addition, as evident in Table 3.3, for both loss functions, the obvious difference in prediction results was not influences very much by the number of features, which differed in all four considered cases. It can be deduced that the features in the dataset include many redundant or weak features so that decreasing the number of features from 517 and 544 to 280 and 251, respectively, does not influence the quality of the prediction.

Table 3.3: Prediction with DemPred using linear features with two different loss functions.

loss function	Mean Squared Error		one-sided log Lorentzian	
test set	private	public	private	public
optimization based on public test set				
L1 feature selection				
λ_1 / # features [a]	0.012 / 517		0.004 / 544	
prediction ***MCC***	0.617	0.577	0.600	0.586
L2 regularization				
λ_2	0.200		0.130	
prediction ***MCC***	0.590	0.557	0.610	0.577
optimization based on 10-fold cross validation with training set				
L1 feature selection				
λ_1 / # features [a]	0.024 / 280		0.011 / 251	
prediction ***MCC***	0.604	0.570	0.583	0.534
L2 regularization				
λ_2	0.350		0.150	
prediction ***MCC***	0.601	0.557	0.616	0.567

a. The number of features after the deletion of weak features by the L1 approach.

The L1&L2 two-step optimization procedure was applied for the construction of the DemPred prediction model. Step 1: only L1 approach is used. Step 2: The features whose weights are set to zero with L1 were removed. L2 regularization was applied to all remaining features.

Figure 3.2 shows that the prediction performance represented by plotting the ***MCC*** value (vertical axis) versus the λ_1 values for private, public test set and training set (recall). The latter actually means to recall what has been learned before. The difference in performance between prediction and recall can be used to evaluate the degree of overfitting. The right side of Figure 3.2 displays the prediction results obtained with the MSE loss function and the left side shows the prediction performance obtained with the 1slL loss function. It is clearly visible that for both loss functions, the prediction performances for the public and private test sets first increase before they decrease with λ_1. The latter happens, since with increasing λ_1 eventually to many features are effectively removed. On average, the prediction performance of the private test set is better than that of the public test set. The recalls of both loss functions keep decreasing as the value the of λ_1 increases. By increasing the λ_1 value, the generality of the DemPred model increases. This increase in generality, however, cannot offer a better prediction performance as it goes along with loss of specificity.

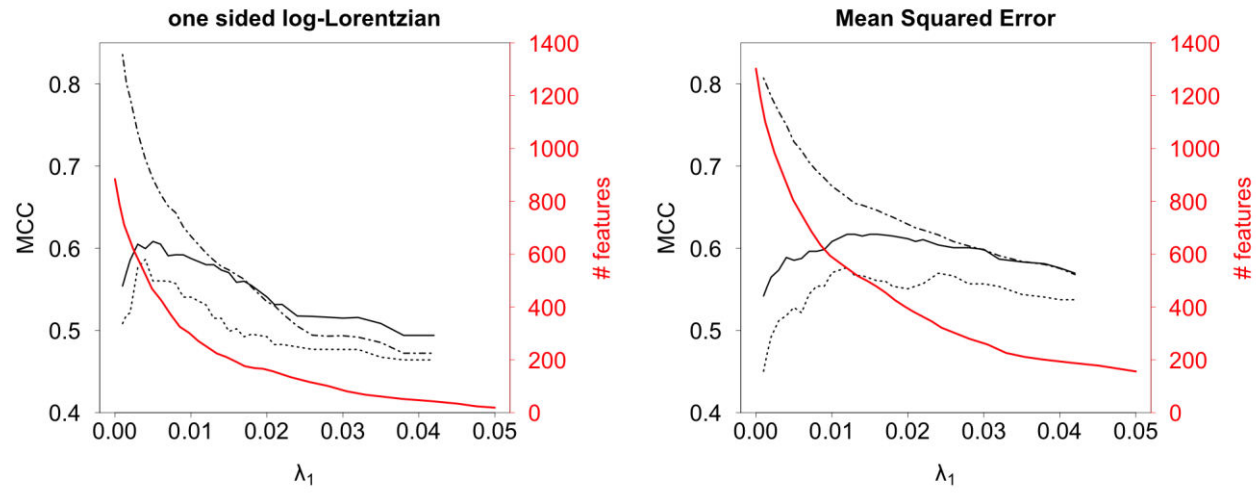

Figure 3.2: Right vertical axes and **black** lines: MCC values plotted versus λ_1 values. Private test set (solid line), public test set (dotted line), training set (i.e. recall) (dashed-dotted line). Left vertical axes, **red** line: number of features plotted versus λ_1 values. This figure was made by R v3.1.3.

When comparing the prediction performances of both loss functions, optimal prediction with 1slL is reached with a relatively small λ_1 value (around $\lambda_1 = 0.004$), while optimal prediction with MSE is reached for $\lambda_1 = 0.012$. In both cases the maximum MCC values is almost the same.

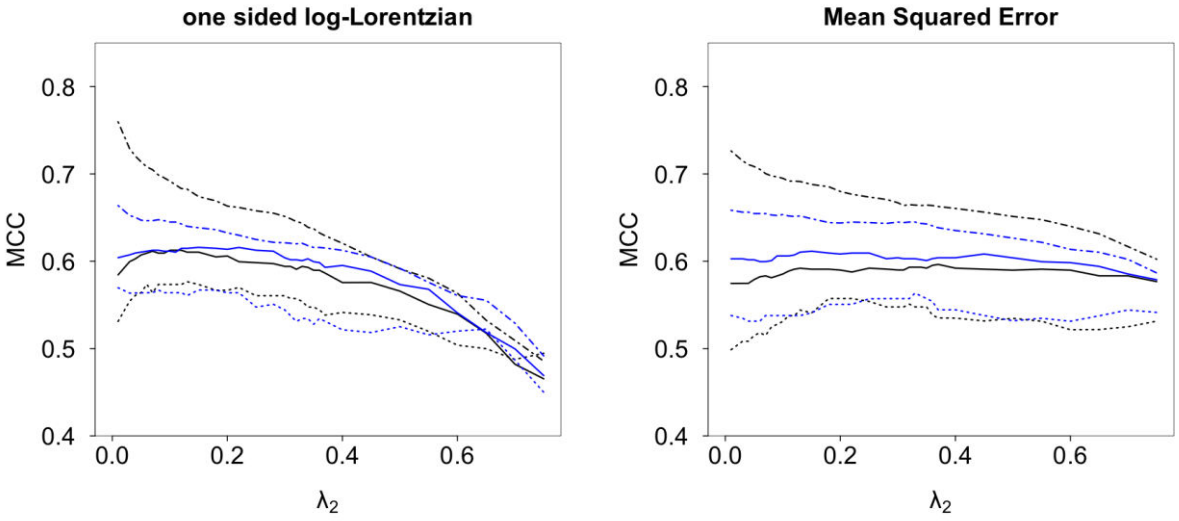

Figure 3.3: MCC values plotted versus the λ_2 value. Private test set (solid line), public test set (dotted line), training set (i.e. recall) (dashed-dotted line). For one sided log-Lorentzian (left), **black** lines: 544 features. **blue** lines: 251 features. For Mean Squared Error (right): **black** lines: 517 features. **blue** lines: 280 features. This figure was made by R v3.1.3.

Figure 3.3 shows the prediction performance represented by the ***MCC*** value (vertical axis) as a function of the λ_2 values for private test set, public test set and training set (recall). The L2

regularizations were applied for the reduced feature sets after L1 feature selection. Although relatively large λ_2 values were used, the influence of λ_2 values is not as large as λ_1 (see Figure 3.2).

In the Kaggle™ competition, the successful teams identified D27 as the most important feature, which strongly correlates with the biological target[53]. In this study, we also used the DemPred model to investigate the importance of features. The absolute values of the weights calculated with the objective function reflect the importance of the corresponding features. Since the feature values vary in the interval [0, 1] contributions of features to the negative biological target value (-1) require negative values of the corresponding weights, while for the positive biological target value (+1) the weights of corresponding features need to be positive. Figure 3.4 demonstrates the most important 20 features measured by the absolute values of weights. Obviously, the absolute weight of D27 feature is far stronger than other features, indicating that it is a very important feature. This result basically agrees with the results of the winning teams of the the Kaggle™ competition.

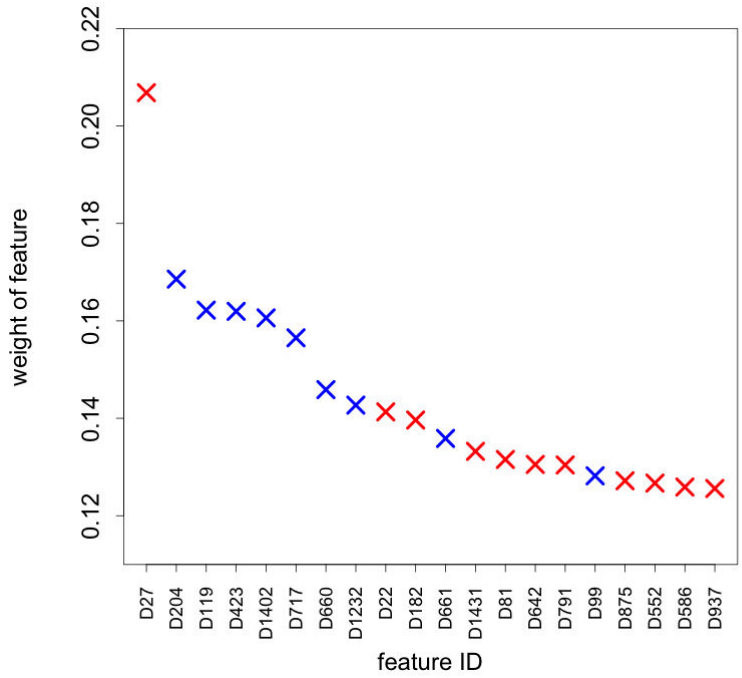

Figure 3.4: Weights of the most important 20 features. The horizontal axis is the features ID and vertical axis reflects the absolute weight values of corresponding features. Red crosses correspond to positive weights and blue crosses to negative weights. The weight values were generated by DemPred model with L2 regularization with λ_2 = 0.008 and MSE loss function. This figure was made by R v3.1.3.

To further investigate the importance of D27, two DemPred models were built. The first DemPred model was built with only the D27 feature while the remaining 19 features were

used to build the second DemPred model. The comparison of prediction performance is shown in Table 3.4. By comparison, it can be seen that D27 exhibits a stronger prediction power than the other 19 best features. Using only D27, the prediction results, ***MCC,*** of the private and public test sets can reach 0.494 and 0.462, respectively. However, the ***MCC*** of the model built with remaining 19 features only manages to reach 0.144 and 0.156 for the private and public test sets. On the other hand, although D27 exhibits a powerful prediction ability, it cannot achieve a good prediction result by itself alone.

Table 3.4: The comparison of prediction results (***MCC***) for the DemPred model.

features	private test set	public test set
only D27[a]	0.494	0.462
remaining 19 features[b]	0.144	0.156

a. ***MCC*** prediction result built only with the D27 feature.
b. ***MCC*** prediction results built with best 19 features except D27.
Both DemPred models were built with the MSE loss function. λ_1 and λ_2 are set to 0.0.

3.1.2.1.2 Prediction results with quadratic features

Applying DemPred for the original set of linear features, the weight of each feature can be calculated. As mentioned before, the weight characterizes the importance of a feature. Based on the absolute values of the weights, the most important 50 features were selected and then multiplied in a pairwise manner with all the linear (original) features. The products formed by the total set of features and the selected features constitute the quadratic features. Chapter 2.11 provides details how the quadratic features are generated.

For the construction of prediction models we now merging the set of quadratic features with the set of original linear features. The resulting larger dimension of the feature space may possibly yield better prediction results. A similar approach was also used for the SVM using the technique of kernel functions to map datasets into a higher-dimensional feature space[31-33]. Brown *et al.*[92]used such an approach to classify genes into different functional categories, obtaining good results. However, sometimes additional information produced by the higher dimensions may hamper the model building so that the prediction performance decreases [82].

The datasets of the Kaggle™ competition use 1776 features, excluding the biological target feature (-1 and +1). In addition to using all 1776 linear features to generate quadratic features, there are 4 other reduced linear features sets prepared by the DemPred L1 approach, which eliminate less important features. By using λ_1 values of 0.002, 0.004, 0.006 and 0.008, the

number of features is reduced from 1776 to 1108, 804, 633, 426, respectively. These are produced using the MSE loss function. Thus, 5 linear feature sets were prepared to produce quadratic features. For each features set, the 50 features with the highest weight values (absolute values) were used to multiply pairwise with all the linear features of the corresponding set. In this way, 5 quadratic feature sets are generated comprising of a large number of new features. The numbers of them are 88000, 55400, 40300, 31650, and 21300, respectively. The detailed information of these quadratic features is listed in Table 3.5.

Table 3.5: The number of quadratic features.

linear features	quadratic features	deleted features [a]	remaining features
1776(all features)	88000	5199	82801
1108	55400	3898	54292
806	40300	2214	36678
633	31650	1671	31017
426	21300	1246	20874

a. features whose values do not vary.

The quadratic features of the five sets are merged with the corresponding linear features. Thus, for each compound, there are 5 feature sets used for generating a prediction model. These five feature sets involve 84577, 55098, 38892, 31650 and 21300 features, respectively.

For these combined feature sets, the L1&L2 two-step method was again applied to build the DemPred model for the Kaggle™ datasets. The 1slL loss function can deal with outliers more appropriately. But, since both loss functions, MSE and 1slL, yield almost the same performance with the linear feature set. There seem to be no strong outliers in the datasets. However, to avoid possible weak outliers, the following studies used 1slL to construct models that the value of loss function increase smoothly with increasing prediction errors and lslL is a 0-1 indicator function, which is more suitable for classification. Using the public test set to optimize results, the L1 feature selection was applied in the construction of DemPred models, out of which the 1108 set produces the best prediction results. The results are shown in Table 3.6. The results of only using linear features were also added to the comparison with quadratic features. The specific feature ID indexes of 1108 features are shown in Appendix 1.

As shown in Table 3.6, in addition to ***MCC***, ***accuracy***, ***sensitivity*** (positive prediction accuracy) and ***specificity*** (negative prediction accuracy) were also employed to evaluate the model quality (see Chapter 2.4). Although a huge number of quadratic features greatly increases the CPU calculation time, the addition of quadratic features has significantly

improved the prediction performance of ***MCC*** and ***accuracy***, as compared to the prediction results of only using linear features dataset. For the public test set, the value of ***MCC*** improves from 0.586 to 0.624 while ***accuracy*** improves from 79.4% to 81.3%. Conversely, for the private test set, the value of ***MCC*** improves from 0.610 to 0.618 while ***accuracy*** improves from 80.8% to 81.8%. This enhancement is mainly due to the ***specificity***. For the public test set, the number of negative compounds correctly predicted increases from 218 to 233 and for the private test set, increases from 630 to 642. Since ***specificity*** reflects the prediction accuracy of negatives, seemingly, it can be deduced that the additional quadratic features are helpful in detecting negative compounds.

Table 3.6: Prediction results with DemPred including quadratic features. The ***accuracy*** is presented as percentage of correct predictions. The number of compounds classified correctly/total number of compounds is given in parentheses. The ***sensitivity*** and ***specificity*** are defined as decimal value that is the number of positive or negative compounds classified correctly/total number of positive or negative compounds, shown in parentheses. All metrics equations can be refer to Chapter 2.4.

linear features only	**MCC**	**accuracy**	**sensitivity**	**specificity**
L1 feature selection with $\lambda_1 = 0.004$; number of features [a] 517				
public test set	0.586	79.4% (496/625)	0.845 (278/329)	0.736 (218/296)
private test set	0.600	80.3% (1506/1876)	0.844 (876/1038)	0.752 (630/838)
L2 regularization $\lambda_2 = 0.130$; number of features [a] 517				
public test set	0.577	78.9% (493/625)	0.839 (276/329)	0.733 (217/296)
private test set	0.610	80.8% (1516/1876)	0.855 (887/1038)	0.751 (629/838)
with quadratic features [b]	**MCC**	**accuracy**	**sensitivity**	**specificity**
L1 feature selection with $\lambda_1 = 0.017$; number of features [a] 1038				
public test set	0.624	81.3% (508/625)	0.836 (275/329)	0.787 (233/296)
private test set	0.618	81.2% (1523/1876)	0.849 (881/1038)	0.766 (642/838)
L2 regularization $\lambda_2 = 0.550$; number of features [a] 1038				
public test set	0.621	81.1% (507/625)	0.845 (278/329)	0.774 (229/296)
private test set	0.614	81.2% (1519/1876)	0.847 (879/1038)	0.764 (640/838)

a. Features selected after L1 approach
b. The addition of quadratic features generated with the 1108 feature set.

As shown in Table 3.6, after using L1 feature selection, the L2 regularization does not improve the prediction performance and even makes it a little worse. It seems that the

quadratic features have been selected rigorously by L1 feature selection. In addition, it is worth noting that many quadratic features are not really useful so that a small λ_1 drastically removed a large number of features, causing a decrease from 54292 to only 1038. The L1&L2 two-step optimization procedure was applied to construct the DemPred model. Step 1: only L1 feature selection is used. The features whose weights were set to zero were removed. Step 2: only L2 regularization was applied to all remaining features after L1 feature selection.

As for the difference between the private and public test sets (by ***MCC***), after the addition of quadratic features, the prediction result of the private test set is better than that of the public test set. However, when the prediction quality of the test sets are judged by ***accuracy***, the difference between the private and public test sets is very small. For the dataset with additional quadratic features, the difference in the ***accuracy*** between the private and public test sets is only 0.1%. Overall, the tests seem to perform by better when judged by ***sensitivity*** than by ***specificity***. However, since datasets are asymmetric containing more molecules with positive than with negative target value the ***MCC*** result is a more reliable measure of prediction quality than ***accuracy***.

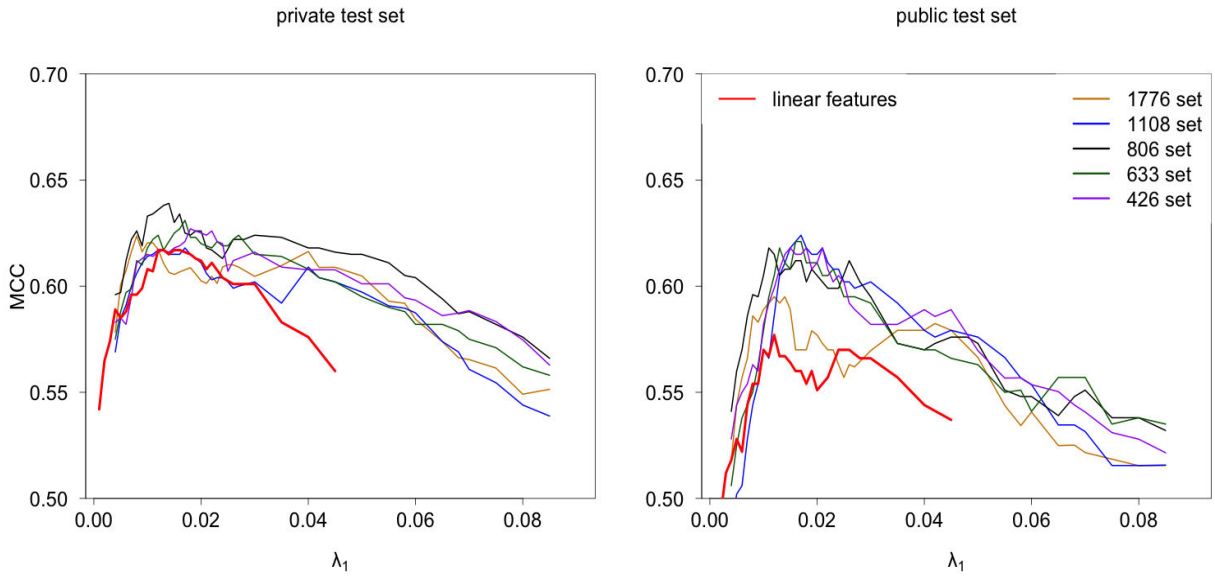

Figure 3.5: The prediction performances of the DemPred models in *MCC* versus the λ_1 value after the addition of quadratic features, and using only linear features. This figure was made by R v3.1.3.

Figure 3.5 illustrates the ***MCC*** performances of DemPred models after the addition of quadratic features and using only linear features. All curves come close to reaching the maximum ***MCC*** value at almost the same λ_1 value of about 0.01 to 0.02. When comparing the overall results using only linear features and after adding quadratic features, it can be found

that the enhancement of prediction performance obtained by the addition of quadratic features is more pronounced for the public test than for the private test set. From the five different quadratic features sets, the prediction performance is the lowest if all 1776 linear features are used to generate the quadratic features. This result seems to reflect that a very large number of features may lead more likely to over-fitting.

3.1.2.2 Prediction results of DemFeature

DemFeature is a new algorithm based on DemPred that has been developed for my doctoral thesis. In DemFeature, each compound in the test set is considered individually, i.e. a specific training subset is designed for each compound to be predicted, based on molecular similarity, which is a computed value varying from -1 to +1. Chapter 2.10 introduces the DemFeature algorithm in detail. DemFeature considers for a compound to be predicted only a limited number of the most similar compounds for the training set. Therefore, the 1slL loss function was used because it is not sensitive to outliers. The objective function of DemFeature includes a L2 regularization term for which a λ_2 value needs to be optimized. Since DemFeature requires more CPU time to calculate results, only the public test set was used to optimize the λ_2 parameter and no cross validation with the training set was used as was done with DemPred.

In this section, two versions of DemFeature: DemFeature-1 and DemFeature-2 were utilized to constitute models to predict the Kaggle™ datasets. DemFeature-1 uses a parameter ***cutoff***, which excludes compounds of the training set that are not sufficiently similar to the compound to be predicted. By contrast, for a compound to be predicted in the training phase, DemFeature-2 only considers the most similar and the most dissimilar compounds of the training set, which can reduce the required CPU time is considerably.

3.1.2.2.1 Prediction results of DemFeature-1

As introduced in Chapter 2.10.1, the parameter $S(k,n)$ represents the similarity between a compound n from the training set and the compound k from the test set. $S(k,n)$ can vary from -1.0 to +1.0. For $S(k,n)$ close to +1.0, the two compounds k and n are very similar, for $S(k,n)$ close to -1.0 the two compounds are opposite in character while for $S(k,n)$ close to 0.0 the two compounds are unrelated (dissimilar). Based on the measurement of similarity, a parameter, ***cutoff*** $\in [-1, +1]$, is used to decided on the number of compounds in the training subset for the compound to be predicted in the test set. Thus, compounds that are not sufficiently related to the compound that is to be classified are ignored in the training phase. ***cutoff*** is also an

empirical parameter. In this study, the optimal ***cutoff*** value out of -0.4, -0.2, 0.0, 0.2, 0.4 was determined using the public test set. Moreover, the five different linear feature sets (1776, 1108, 806, 633 and 426 feature sets as produced in Chapter 3.1.2.1.2) were prepared for the construction of the DemFeature-1 model. Firstly, the ***cutoff***=0.0 was used to find the best feature set. Among five feature sets, the 1108 features dataset gives the best prediction performance (see Figure 3.6). Then the optimized ***cutoff*** value (-0.2) is obtained for the 1108 feature set. Table 3.7 shows the prediction performance of DemFeature-1.

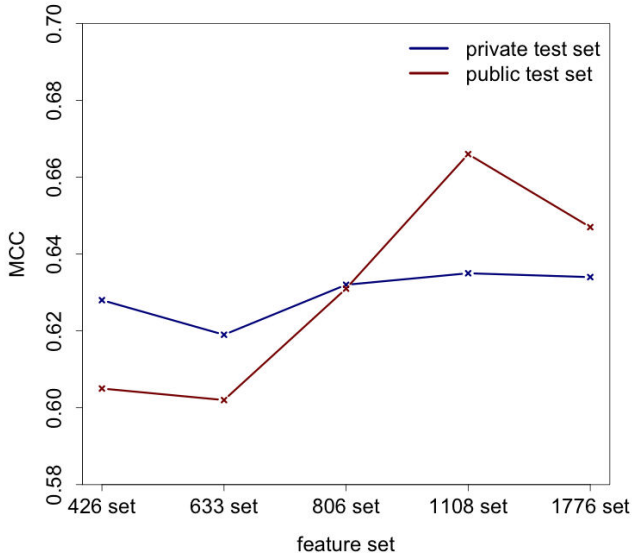

Figure 3.6: 5 *MCC* for the feature sets 1776, 1108, 806, 633 and 426. The 1108 feature set produces the best prediction results in public and private test sets. Here, ***cutoff*** = 0.0. This figure was made by R v3.1.3.

Table 3.7: Prediction performance of DemFeature-1 for the Kaggle™ datasets using the 1108 feature set with ***cutoff*** = -0.2 and λ_2 = 0.06.

	MCC	**accuracy**	**sensitivity**	**specificity**
public test set	0.676	83.8% (534/625)	0.857 (283/329)	0.807 (241/296)
private test set	0.641	82.3% (1543/1876)	0.842 (875/1038)	0.797 (668/838)

For DemFeature-1, the prediction quality is for the public test set better than for the private test set, albeit the differences are smaller for ***accuracy*** than for the ***MCC*** values. This result is astonishing, since the public test set should by construction contain compounds, which are on the average more difficult to predict[53] than the private test set. However, the parameters ***cutoff*** and λ_2 have been optimized by using the public test set, which might have introduced some preference for the prediction of the public test set. The value of the ***sensitivity*** is larger than of

the ***specificity***, since the number of positive compounds is larger than the number of negative compounds.

Since for each compound considered for prediction a specific DemFeature-1 model must be built, applications with DemFeature-1 are more CPU time intensive than with DemPred. However, as shown by the results in Figure 3.7, DemFeature-1 greatly improves the quality of prediction. Interestingly, using quadratic features for the DemPred model improves the prediction performance compared to using only linear features, but it is not better than the DemFeature-1 model and needs more CPU time.

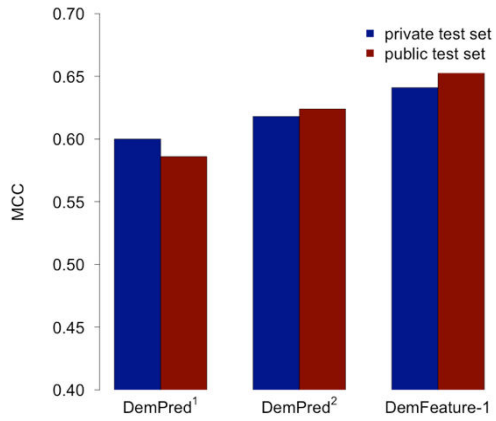

Figure 3.7: Comparison of prediction performances for three different models. DemPred[1] model built with linear features only. DemPred[2]: built with linear and quadratic features. DemFeature-1 built with 1108 features, ***cutoff*** = -0.2 and λ_2 = 0.06. This figure was made by R v3.1.3.

How the prediction performance depends on the parameter ***cutoff*** is shown in Figure 3.8 for the DemFeature-1 model. It illustrates that the distribution of compounds in the training set is based on a similarity with compounds in the private and public test sets. The similarities of a compound to be predicted with DemFeature-1 using the compounds in the training set are different for the two test sets. Based on the similarity value ***S*(*k*,*n*)**, a different number of compounds in the training set would be distributed across different intervals of similarity (vertical axis of left side of Figure 3.8). The horizontal axis of left side of Figure 3.8 represents the mean of the overall compounds in the test sets at each similarity interval.

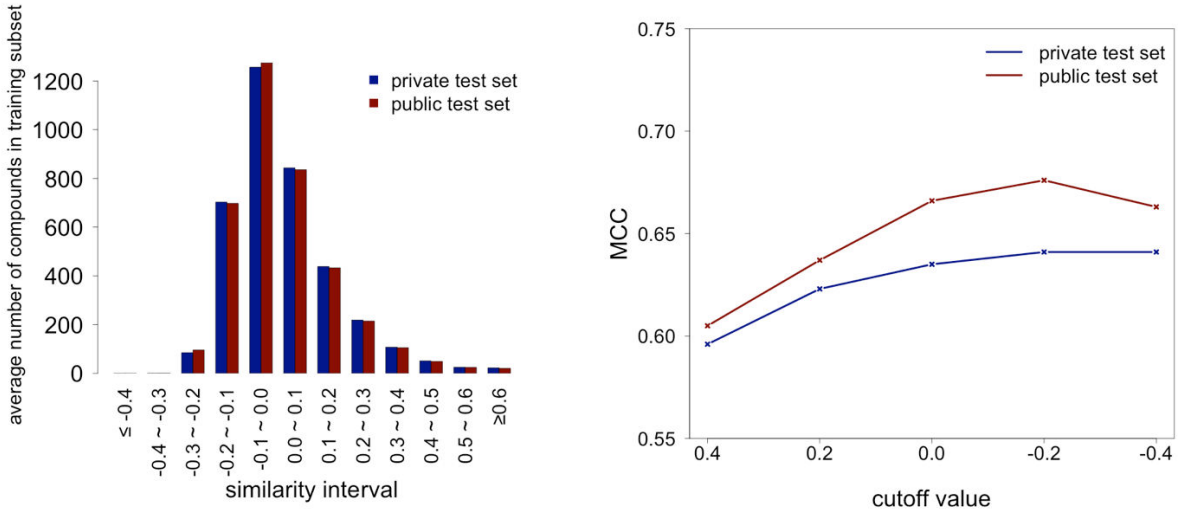

Figure 3.8: Left: Histogram of similarities. The number of training compounds is plotted as function of the similarity averaged over the compounds of the public (red) and private (blue) test sets. Right: prediction performance (***MCC***) as function of the ***cutoff*** value. This figure was made by R v3.1.3.

Basically, the similarities of the compounds in the training set averaged relative to the test sets are approximately normal distributed (left side of Figure 3.8). The similarity distributions of the compounds in the training set relative to the private and public test sets are very similar. The similarity is concentrated between -0.2 to 0.2. The most similar compounds (similarity more than 0.2) and the most dissimilar compounds (similarity less than -0.2) occupy less than 15% of all compounds in the training set. The parameter ***cutoff*** decides on the number of compounds in the training subset through a measurement of similarity. For example, if the ***cutoff*** value is equal to 0.2, in order for a compound to be predicted, the compounds in the training set with similarities above 0.2 would be selected to construct a specific training subset.

The left side of Figure 3.8 provides MCC values for ***cutoff*** value as large as 0.4, where only the few most similar compounds are used to construct the training subset and thus cannot produce the best prediction result. However, if the ***cutoff*** value is very small, for example -0.4, too many unrelated compounds will be brought into the training subset, which causes the prediction quality to decline. Combining this with the right part of Figure 3.8, it is clear that to predict a compound, the optimized ***cutoff*** value of -0.2 removes only a few irrelevant compounds (less than 0.03% on average) from the training subset but retains the majority of compounds. For DemFeature-1, it seems that in the dataset, only the few very dissimilar compounds lead to a decrease in prediction performance.

3.1.2.2.2 Prediction result of DemFeature-2

DemFeature-2 employs a different method to compile a compound specific training subset. While DemFeature-1 utilizes the parameter ***cutoff*** to determine the number of compounds in the training subset for a compound to be classified, DemFeature-2 directly selects a fixed number of most similar and dissimilar compounds from a training set to constitute a training subset for a compound to be predicted. For Equation (2.38), the parameters N_{sim}^{+}, N_{dis}^{+}, N_{sim}^{-}, N_{dis}^{-} represent four respective sets: the number of most similar positive compounds, most dissimilar positive compounds, most similar negative compounds and most dissimilar negative compounds in the training set for a considered test compound. For instance, when N_{sim}^{+} =100, it means that for the considered test compound, the 100 most similar positive training compounds will be selected to be a part of training subset. In the following text, the names of these sets are abbreviated as ***PosSim***, ***PosDis***, ***NegSim*** and ***NegDis***. Since these numbers are empirical parameters, the prediction results of the public test set were used to determine the number of these parameters as well as used for optimizing other parameters. In addition, the objective function of DemFeature-2 includes the L2 regularization term for controlling overfitting with possible values of $\lambda_2 \in \{1.2, 1.5, 1.8, 2.0, 2.5, \ldots, 40.5\}$. Several parameter combinations were chosen and the prediction results of the private and public test sets were listed in Table 3.9 and Figure 3.9.

As shown in Table 3.9 and Figure 3.9, the number of compounds in ***PosSim*** was increased from 100 to 400 and while the number of compounds in the other three sets (***NegSim***, ***PosDis*** and ***NegDis***) remained at 100. Then, the same operation was repeated in ***NegSim***. Results show that the increment of the number of compounds in ***PosSim*** and ***NegSim*** does not reflect an obvious influence on the prediction performance and does not result in better prediction performance than the combination of 100 compounds in all four sets (***PosSim***, ***PosDis***, ***NegSim*** and ***NegDis***). However, as compared with using 100 compounds in all four sets, results indicate that by increasing the number of compounds on ***PosSim*** improves ***sensitivity*** while the same performance in ***NegSim*** improves ***specificity***.

Increasing the number of compounds in ***PosDis*** improves the ***specificity*** while increasing the number of compounds in ***NegDis*** enhances ***sensitivity***. However, increasing the number of compounds in ***PosDis*** or ***NegDis*** produces a side effect: when the number of compounds in ***PosDis*** is increased, the ***specificity*** increases while the ***sensitivity*** gradually decreases. Increasing the number of compounds in ***NegDis*** is on contrary to the same increment in

PosDis: as the number of compounds in ***NegDis*** increases, ***sensitivity*** gradually ascends and ***specificity*** gradually descends.

Table 3.9: Prediction results with DemFeature-2 using different training subsets I.

PosSim	PosDis	NegSim	NegDis	λ_2	public test set				private test set			
					MCC	accuracy	sensitivity	specificity	MCC	accuracy	sensitivity	specificity
100	100	100	100	7.5	0.634	0.818	0.851	0.780	0.607	0.807	0.847	0.757
150	100	100	100	6.5	0.634	0.818	0.848	0.784	0.607	0.807	0.847	0.757
100	**150**	100	100	6.5	**0.663**	0.830	0.805	0.858	**0.622**	0.810	0.790	0.835
100	100	**150**	100	13.0	0.638	0.819	0.869	0.764	0.614	0.810	0.838	0.764
100	100	100	**150**	11.0	0.600	0.795	0.912	0.315	0.603	0.803	0.901	0.681
200	100	100	100	6.5	0.634	0.818	0.854	0.777	0.607	0.807	0.851	0.752
100	**200**	100	100	1.5	0.606	0.798	0.736	0.868	0.614	0.801	0.739	0.877
100	100	**200**	100	1.8	0.634	0.818	0.851	0.780	0.608	0.820	0.841	0.757
100	100	100	**200**	7.5	0.557	0.770	0.924	0.598	0.572	0.783	0.932	0.598
300	100	100	100	4.5	0.632	0.816	0.854	0.774	0.607	0.807	0.850	0.753
100	**300**	100	100	2.0	0.535	0.747	0.593	0.912	0.565	0.761	0.632	0.921
100	100	**300**	100	15.0	0.637	0.819	0.830	0.807	0.62	0.812	0.828	0.792
100	100	100	**300**	2.0	0.499	0.733	0.942	0.500	0.528	0.753	0.958	0.500
400	100	100	100	4.0	0.625	0.813	0.854	0.767	0.605	0.807	0.849	0.752
100	**400**	100	100	2.0	0.492	0.715	0.517	0.936	0.538	0.733	0.551	0.956
100	100	**400**	100	2.0	0.631	0.816	0.854	0.774	0.604	0.805	0.845	0.755
100	100	100	**400**	2.0	0.466	0.707	0.964	0.422	0.498	0.732	0.972	0.436

1. ***PosSim***: number of the most similar positive compounds; ***PosDis***: number of the most dissimilar positive compounds; ***NegSim***: number of the most similar negative compounds; ***NegDis***: number of the most dissimilar negative compounds.
2. The parameters highlighted indicate the numbers varied in a combination.
3. λ_2 was determined by the prediction results of public test set

Based on the results described above, in order to obtain a good prediction performance, a proper balance between ***sensitivity*** and ***specificity*** is necessary. Although a larger number of compounds in ***PosDis*** or in ***NegDis*** can produce a higher ***specificity*** or ***sensitivity***, the improvement on ***sensitivity*** or ***specificity*** significantly weakens the other, thus leading to a poor prediction performance. The combination of 150 compounds in ***PosDis*** and 100 compounds in the other three sets produces the best prediction performance of both public test set and private test set. Results show that using 150 compounds in ***PosDis*** can improve ***specificity*** while it does not significantly impair ***sensitivity***. It is worth noting that based on the public test set, this is also the optimized combination. Furthermore, the prediction results of increasing the number of compounds on two sets concurrently can be observed in Table 3.10 and Figure 3.10.

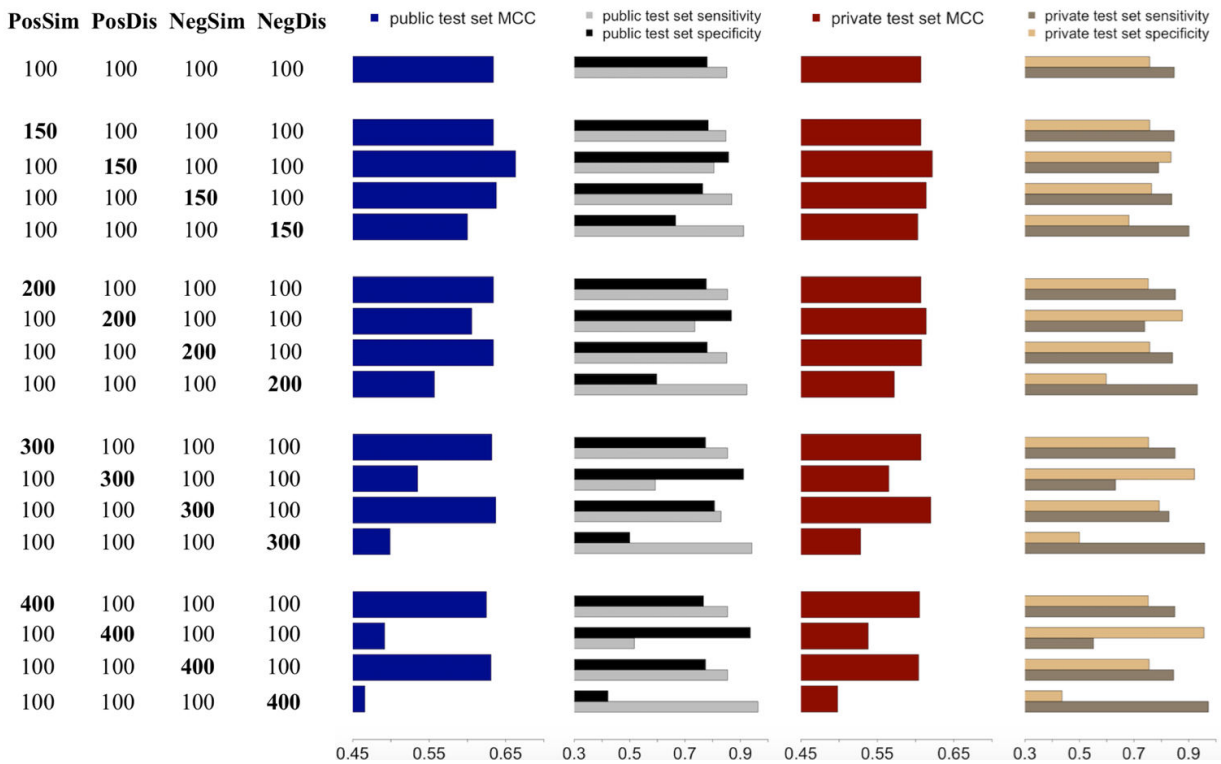

Figure 3.9: Prediction results with DemFeature-2 using simultaneously four different training subsets (***PosSim***, ***PosDis***, ***NegSim***, ***NegDis***). To investigate the influence of the size of these four sets, the number of compounds in one of the four sets is increased while that of other sets remain unchanged. This figure was made by R v3.1.3.

Table 3.10: Prediction results with DemFeature-2 using four different training subsets.

PosSim	PosDis	NegSim	NegDis	λ_2	public test set				private test set			
					MCC	accuracy	sensitivity	specificity	MCC	accuracy	sensitivity	specificity
150	100	**150**	100	2.0	0.640	0.821	0.851	0.787	0.595	0.803	0.842	0.749
100	**150**	100	**150**	6.0	0.631	0.816	0.854	0.774	0.624	0.814	0.856	0.763
200	100	**200**	100	5.5	0.634	0.818	0.845	0.787	0.608	0.807	0.845	0.760
100	**200**	100	**200**	6.5	0.654	0.827	0.866	0.784	0.624	0.814	0.857	0.761
300	100	**300**	100	33.0	0.634	0.818	0.446	0.784	0.618	0.811	0.836	0.779
100	**300**	100	**300**	6.5	0.651	0.826	0.872	0.774	0.619	0.812	0.855	0.760
400	100	**400**	100	29.0	0.634	0.818	0.842	0.791	0.626	0.816	0.840	0.785
100	**400**	100	**400**	6.5	0.644	0.822	0.869	0.770	0.620	0.813	0.853	0.764
300	300	300	300	3.0	0.660	0.830	0.872	0.784	0.627	0.816	0.854	0.770
400	400	400	400	9.5	0.644	0.822	0.869	0.770	0.621	0.813	0.846	0.772

1. ***PosSim***: number of the most similar positive compounds; ***PosDis***: number of the most dissimilar positive compounds; ***NegSim***: number of the most similar negative compounds; ***NegDis***: number of the most dissimilar negative compounds.
2. The parameters highlighted indicate the numbers varied in a combination.
3. λ_2 was determined by the prediction results of public test set.

Comparing the results listed in Table 3.10 with results of the combination (***PosSim***:100; ***PosDis***: 100; ***NegSim***:100, ***NegSim***:100), it is found that the increment of the number of compounds in ***PosSim*** and ***NegSim*** does not effectively improve prediction performance. On the other hand, the concurrent increment of the number of compounds in both ***PosDis*** and ***NegDis*** are capable of properly balancing ***sensitivity*** and ***specificity***. However, its prediction performance still is not better than for the combination (***PosSim***:100; ***PosDis***: 150; ***NegSim***:100, ***NegSim***:100).

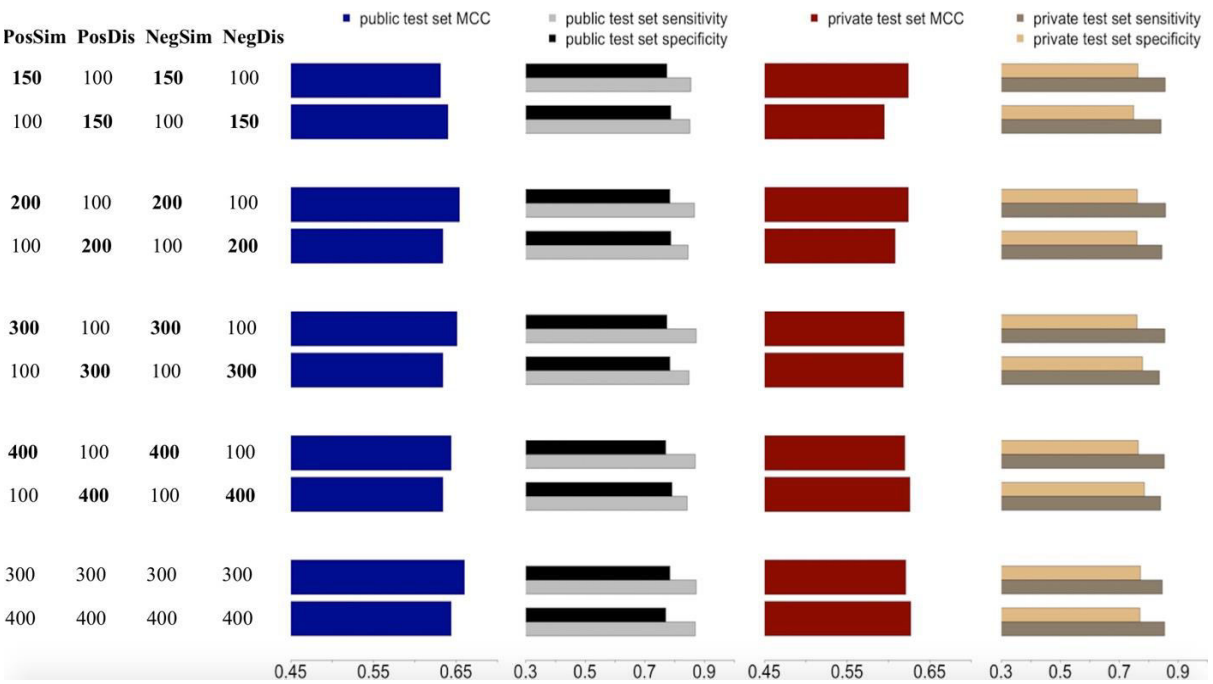

Figure 3.10: Prediction results with DemFeature-2 using simultaneously four different training subsets (***PosSim***, ***PosDis***, ***NegSim***, ***NegDis***). In this investigation, the number of compounds in ***PosSim*** and ***NegSim*** were increased at the same time while those in ***PosDis*** and ***NegDis*** remained at 100 compounds. Then this was repeated by increasing the number of compounds in ***PosDis*** and ***NegDis*** and keeping the number of compounds in ***PosSim*** and ***NegSim*** at 100. This figure was made by R v3.1.3.

When the number of compounds in all four sets is increased to 300, a fairly good prediction performance (MCC: 0.660 on public test set; MCC: 0.627) can be achieved. However, the number of compounds in a training subset for each compound to be predicted would reach 1200, which takes as much CPU time as when using DemFeature-1, but it does not produce a prediction performance which is as good as obtained with DemFeature-1. In Summary, when DemFeature-2 considers only the most similar and dissimilar training compounds for a test compound, it produces a better prediction performance than DemPred (see Figure 3.11). Although the prediction results are not as good as the one of DemFeature-1, it takes considerably less CPU time than DemFeture-1. Additionally, its results are easier to be

explained when further investigations are conduced with a smaller number of compounds in the training subset.

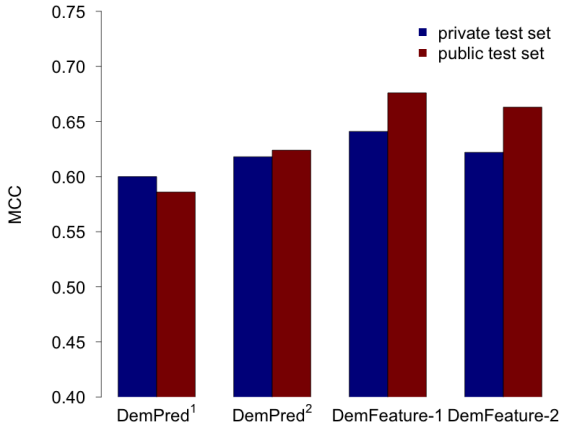

Figure 3.11: Comparison of prediction performances among four different models. DemPred[1] built with linear features only. DemPred[2] built with quadratic features plus linear features. DemFeature-1 built with 1108 features with ***cutoff*****:** -0.2. DemFeature-2 built with the combination (***PosSim***:100; ***PosDis***: 150; ***NegSim***:100, ***NegSim***:100). This figure was made by R v3.1.3.

Table 3.11: Comparison of prediction performance of different models.

method	MCC	accuracy	Sensitivity	specificity
public test set				
1. on public test set; 3. on private test set[a]	0.637	81.9% (516/625)	0.818 (268/329)	0.794 (248/296)
11. on public test set; 1. on private test set	0.652	82.7% (512/625)	0.815 (273/329)	0.838 (239/296)
DemFeature-1	0.676	83.8% (534/625)	0.857 (283/329)	0.807 (241/296)
random forest benchmark on Kaggle™ competition	0.586	79.4% (496/625)	0.818 (269/329)	0.767 (227/296)
SVM benchmark on Kaggle™ competition	0.529	76.5% (478/625)	0.833 (274/329)	0.689 (204/296)
private test set				
1. on public test set; 3. on private test set	0.668	83.6% (1569/1876)	0.860 (893/1038)	0.807 (676/838)
11. on public test set; 1. on private test set	0.660	83.2% (1560/1876)	0.841 (873/1038)	0.820 (687/838)
DemFeature-1	0.641	82.3% (1543/1876)	0.842 (875/1038)	0.797 (668/838)
random forest benchmark on Kaggle™ competition	0.659	83.2% (1560/1876)	0.855 (873/1038)	0.802 (687/838)
SVM benchmark on Kaggle™ competition	0.535	77.0% (1445/1876)	0.792 (888/1038)	0.743 (672/838)

[a] For the competition, this model is ranked 1st in the public test set and ranked 3rd in the private test set.

3.1.2.3 Comparison with results from the Kaggle™ competition

Since the DemFeature-1 model gives the best result, the achieved results by the top-ranking participants of the Kaggle™ competition were used to compare with the result of the DemFeature-1 model. The Kaggle™ competition ranks the contest in an order following the LogLoss (see Equation (3.1)),

$$Logloss = -\frac{1}{N}\sum_{n=1}^{N} m_n \log(\hat{m}_n) + (1-m_n)\log(1-\hat{m}_n) \tag{3.1}$$

where $\hat{m}_n$ is the posterior probability that the $\boldsymbol{n}^{\text{th}}$ sample elicited a response and m_n is the experimental observation value. Regarding the property of the DemFeature-1 model, we did not use the LogLoss to evaluate the prediction performance. According to the information provided by Boehringer Ingelheim Inc., we calculated several quality measures (see Chapter 2.4) of top models for comparison. The results are shown in Table 3.11.

For the Kaggle™ competition, the best results of the public and private test sets were not achieved by the same participant. Therefore, we provide the models ranking 1st on the private and public test sets, respectively. In addition, the results produced by the random forest benchmark and the SVM benchmark of the Kaggle™ competition are also provided for comparison. In the Kaggle™ competition, the random forest and SVM models are provided by the Scikit-learn package[93]. The random forest model was trained with default parameters. SVM used radial basis function as its kernel function and was also trained with default parameters. By contrast, the DemFeature-1 model obviously gives the best results in public test set by ***MCC*** (public test set: 0.676, private test set: 0.641) and ***accuracy*** (public test set: 83.8%, private test set: 82.3%). With respect to the ***sensitivity*** of the public test set, DemFeature-1 is also the best. The model that ranked 1st in the private test set gives a higher ***specificity*** than the DemFeature-1 model. Although the DemFeature-1 model does not give the best result in the private test set, it can still considered to be reasonably good. On the other hand, the ***sensitivity*** is higher than the ***specificity*** for both private and public test sets, which can be explained by the number of positive compounds being larger than the number of negative compounds (public test set: 329 positives and 296 negatives; private test set: 1038 positives and 838 negatives), which in turn means that the positive compounds are easier identified by *in silico* prediction. It must be noted that high accuracy of positive compounds is more important in drug development.

Another point worth taking note is that the SVM benchmark performs very poorly in the Kaggle™ competition. However, it cannot be denied that the SVM model is an excellent

algorithm[31-33]. SVM has been extensively applied in different fields and has performed excellent for different datasets. Basically, DemFeature-1 algorithm works similarly as SVM; both are linear classifier algorithms. However, DemFeature-1 has been significantly improved for the prediction of the Kaggle™ dataset. In other words, the DemFeaure-1 model could be an alternative solution for the problems that cannot be solved sufficiently precise by SVM. In summary, all of the top-ranking models have excellent prediction performance in both private and public test sets (see Table 3.11). In contrast to the results from the Kaggle™ contest DemFeature-1 performs slightly better for the public test set than for the private test set although the public test set should by construction be more difficult to predict[53]. This may be the case, since DemFeature-1 uses more information from the public test set than the participants of the Kaggle™ contest.

3.1.2.4 *P*-values to compare different models

In order to compare the difference in performance, the *P*-value (see Chapter 2.12) is employed to evaluate the difference in the prediction results generated by two different models. A *P*-value can be calculated using a binomial test (see Equation (2.41)) or a McNemar's test (see Equation (2.42)), and usually varies between 0.0 and 1.0. The higher a *P*-value is, the more similar the results of the two models are. In this research, a *P*-value matrix (see Table 3.12) was constructed to pairwise compare the differences among the top-ranking models of the competition: the random forest benchmark, the SVM benchmark and the DemFeature-1 model. The *P*-values of the models are compared separately for the private and public test sets.

As shown in Table 3.12, in the private test set, the top-ranking models of the Kaggle™ competition, as well as the random forest and DemFeature-1 models are significantly different from the SVM model, since the corresponding *P*-values are all smaller than 0.0001. The *P*-value between the top-ranking models and random forest are all significantly larger than zero for the private test. Thus, the top-ranking models are not significantly different from the random forest benchmark. This small difference proves furthermore that for private test set the prediction results among the top-ranking models are very similar. Based on the *P*-values for the private test set of DemFeature-1 with random forest and the top-ranking models, it is clear that the DemFeature-1 model is not significantly different from these models.

Table 3.12: Comparison of *P*-values among different models.

public test set					
methods	**SVM**	1st on private 11th on public	2nd on private 26th on public	3rd on private 1st on public	**DemFeature-1**
random forest	0.102	0.005	0.007	0.038	**0.001**
SVM		0.001	<0.0001	0.001	**<0.0001**
1st on private test set 11th on public test set			0.860	0.556	**0.322**
2nd on private test set 26th on public test set				0.522	**0.201**
3rd on private test set 1st on public test set					**0.097**
private test set					
methods	**SVM**	1st on private 11th on public	2nd on private 26th on public	3rd on private 1st on public	**DemFeature-1**
random forest	<0.0001	0.921	1.000	0.452	**0.229**
SVM		<0.0001	<0.0001	<0.0001	**<0.0001**
1st on private test set 11th on public test set			1.000	0.342	**0.193**
2nd on private test set 26th on public test set				0.322	**0.162**
3rd on private test set 1st on public test set					**0.047**

For the public test set, the *P*-values between SVM and random forest prediction model are larger than for the private test set (in private test set: <0.0001; in the public test set: 0.102). In addition, the *P*-values between SVM and the two top-ranking models (the model of 1st in the private test set and the model of 3rd in the private test set) also increase from <0.0001 to 0.001. Moreover, the *P*-values between the DemFeature-1 model and the top-ranking models are slightly higher in the public test set than in the private test set. On average, the *P*-values among top-ranking models in the public test set are also higher than in the private test set. Hence, overall, except for the random forest model, the prediction results of all models are more consistent in the public test set than in the private test set. This may be due to the fact that all prediction models except the random forest used at least implicitly information on the prediction performance for the public test set.

The DemFeature-1 model is based on an algorithm that is very different from the random forest. On the other hand, all top-ranking models of the Kaggle™ contest were developed from random forest[53]. However, it is worth noting that except when compared with SVM (*P*-value is smaller than 0.0001), the DemFeature-1 model does not reflect any significant difference for the private and public test sets when compared with the high ranked models of the Kaggle™ contest.

3.1.2.5 Measure confidence of prediction

Correlating the confidence (see Equation(2.45)) for each compound in the dataset with prediction performance is helpful to estimate the validity of the prediction results. The introduction of confidence measurement is detailed in Chapter 2.13. In this study of the Kaggle™ dataset prediction, a polynomial function of third order is employed for fitting the prediction performance to the confidence. Since the number of compounds in the datasets is not large, for properly generating a fitting curve, all compounds of private test set and public test set were used together. Hence, the total number of the considered compounds is 2501. The predictive values of DemFeature-1 (see Table 3.7) model were used to measure the confidence of prediction as it produces the best prediction results among all of our models. Since the prediction performance cannot be calculated for a single compound, the confidence (ranged between 0.0 and 1.0) was equally divided into 20 intervals. For each interval, 30 compounds would be randomly picked up for the calculation of ***accuracy*** and ***MCC***. This process would be repeated 20 times to calculate the mean value of ***accuracy*** and ***MCC***. In addition, the standard deviations of 20 times prediction results of those intervals are also calculated for generating error bars.

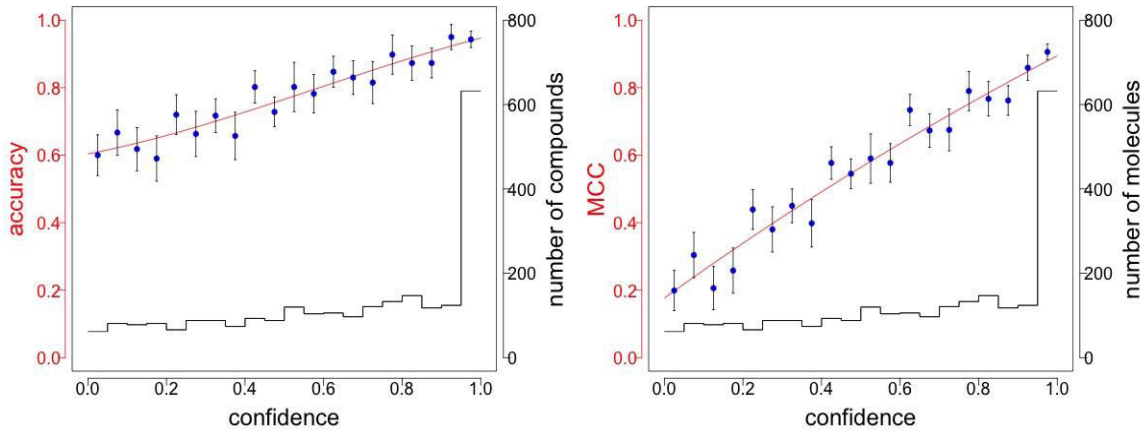

Figure 3.12: Correlation between confidence (see Equation (2.45)) and the prediction performance (***accuracy*** and ***MCC***) for both private test set and public test set. The predictive values are produced by DemFeature-1 (Equation (2.37)). Histogram (**black**) indicates the distribution of the number of compounds at the 20 confidence intervals (left to right on the horizontal axis: low confidence to high confidence). The filled **blue** circles represent ***accuracy*** and ***MCC***. The fitting curves (**red**) are generated by third order polynomials:

$$accuracy = 0.01761conf^3 - 0.13912conf^2 + 0.83949conf^1 + 0.17666,$$

$$MCC = -0.1597conf^3 + 0.2787conf^2 + 0.2245conf^1 + 0.6035;$$

This figure was made by R v3.1.3.

The mean values and standard deviations were calculated by randomly picking up 30 compounds from each interval and repeated 20 times.

Figure 3.12 shows that the prediction performances (***accuracy*** and ***MCC***) strongly correlate with the confidence of the compounds. Basically, the ***accuracy*** (left side) has a good agreement with the ***MCC*** (right side), whereas, for the fitting curves of the prediction performances, the ***MCC*** has a steeper slope than the ***accuracy***. Generally, the number of compounds in the intervals gradually increases from a low confidence level to a high confidence level, although the slight declines occurred at several interval bins. It drastically increases on the interval between 0.95 to 1.00, which accounts for 25.3% compounds of dataset and its ***accuracy*** can reach 0.943 and ***MCC*** is 0.906. This clearly shows that the compounds with a higher confidence are more likely to be classified correctly demonstrating the usefulness of the confidence value.

3.1.3 Discussion

Analysis of outliers

In this study, two loss functions combined with L1&L2 two-step regularization were applied to build DemPred models. The MSE loss function (Equation (2.25)) is capable of recalling each compound in the training set as accurate as possible so that outliers of the training set may have a strong influence on the prediction result. The second loss function used in this research is 1slL (Equation (2.27)), which is not sensitive to the outliers in the training set. Technically, 1slL is more suitable to deal with a classification task because it is not necessary to precisely locate all compounds in the feature space. Its value increases smoothly with increasing prediction error so that it can weaken the influence of potential errors in the dataset. However, as shown in Figure 3.2, the maximum prediction results of the DemPred models with the two loss functions are almost the same. Consequently, it can be deduced that no significant outlier exist in the Kaggle™ datasets.

DemPred built with L1 & L2 two step method

The L1 approach is capable of setting the weights of features to zero so that features can be removed. Already with a small value of $\boldsymbol{\lambda_1}$ many features were removed and at the same time the prediction performance was improved. From this we can conclude that the deleted features have no predictive power or may even disturb prediction results. On the other hand, among the remaining useful features many are nearly equivalent. This is demonstrated by comparing

the prediction performance using 544 features with the one using only 251 features (see Table 3.3), which is nearly the same. It is interesting to observe that after L1 feature selection, the L2 regularization for the remaining features did not improve prediction performance any more. This phenomenon was also described by Demir *et al.*[90]. This may be explained by the rigorous feature selection done by the L1 approach.

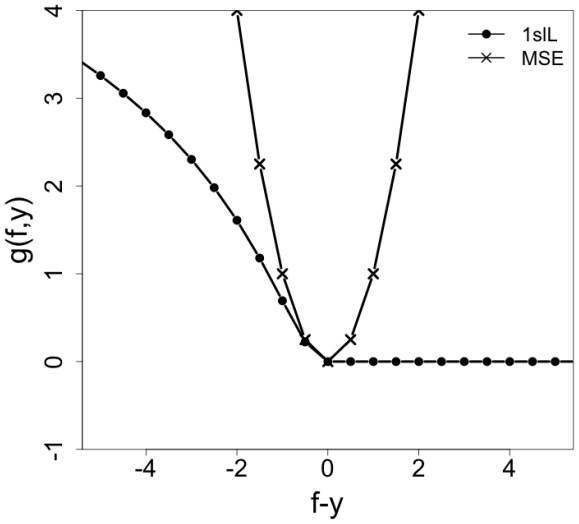

Figure 3.13: The two loss functions ***g(f,y)*** used in the computations: 1slL (Equation (2.27)) and MSE (Equation (2.28). Horizontal axis represents the ***f-y***, which is the difference between the estimating value of scoring function ***f*** and the corresponding true property value ***y***. Crosses represent MSE and solid points represent 1slL. This figure was made by R v3.1.3.

Analysis of feature D27

Our results demonstrated that feature D27 is obviously more important than all other features used for the Kaggle™ competition. Another direct investigation was carried out to observe the numerical value of D27. D27 is a binary feature with values of 1 and 0. Assigning the value 1 of for feature D27 to the biological target value +1 and 0 to the biological target value -1, 74% of all compounds involving both test sets and the training set do match. For 4628 compounds of the 6252 compounds in the training set, private test set and public test set, the values 1 and 0 of the D27 feature of compounds exactly match with the corresponding biological target values, +1 and -1, respectively. This result explains why D27 plays such an important role in this dataset.

DemFeature-1

DemFeature-1 utilizes a parameter ***cutoff*** to govern the number of compounds in an individual training subset. With the value of the parameter ***cutoff*** the more similar compounds in the training set are selected to constitute a training subset for a specific test compound to be considered. Therefore, choosing an appropriate ***cutoff*** value is crucial. Since the public test set was available during the Kaggle™ competition, it was used here to optimize the parameters ***cutoff*** and λ_2. The prediction performance improved if the very dissimilar compounds were not contained in the training subset (see Figure 3.8), which was achieved by using a value of ***cutoff*** that is not too negative. This shows that very dissimilar compounds may hamper the resulting prediction model. If on the other hand the value of ***cutoff*** is too positive, the number of compounds in the training subset becomes too small to generate a successful prediction model. Since the individual feature values vary with different test compounds, the number of compounds in the training subset obtained with the same value of ***cutoff*** is different for each test compound.

DemFeature-2

For DemFeature-2, the number of compounds of training subset is fixed. Four parameters including ***PosSim***, ***PosDis***, ***NegSim***, ***NegDis*** are utilized instead of ***cutoff***. For a test compound, these parameters represent the number of most similar positive compounds, the number of most dissimilar positive compounds, the number of most similar negative compounds and the number of most dissimilar negative compounds, respectively. These parameters are optimized using the public test set. If ***PosSim***, ***NegSim*** and ***NegDis*** remain unchanged and only the number of compounds in ***PosDis*** is increased, the ***specificity*** would improve whereas the ***sensitivity*** would deteriorate. If ***PosSim***, ***PosDis*** and ***NegSim*** remain unchanged and the number of compounds in ***NegDis*** increases, this would lead to an increase of ***sensitivity*** while ***specificity*** would decrease. Therefore, a proper balance between ***sensitivity*** and ***specificity*** is necessary for obtaining a good prediction performance. As the results in Chapter 3.1.2.2.2 showed, this balance requires that the number of compounds in ***PosDis*** or in ***NegDis*** should not be too large. The best combination found is ***PosSim***:100; ***PosDis***: 150; ***NegSim***:100, ***NegDis***:100. For this combination, when the number of compounds is slightly increased to 150 in ***PosDis***, the ***specificity*** reaches 0.858. This increment does not impair ***sensitivity*** too much as ***sensitivity*** remains at 0.805. By contrast, if the number of compounds in ***NegDis*** is increased to 150 while the other three sets remain unchanged, the ***sensitivity*** reaches 0.912 but the ***specificity*** goes down to 0.315. Hence, it

seems that the majority of the compounds in ***NegDis*** are useful for the prediction performance of the positive set, while only a small portion of the compounds in ***PosDis*** are useful for the prediction performance of the negative set. Another reason causing this phenomenon could be that the number of compounds in the positive set is larger than the number of compounds in the negative set (see Table 3.1) so that ***specificity*** changes more easily by the number of compounds predicted correctly in the negative set. Hence, for obtaining a robust DemFeature-2 model, it is very important to balance between ***sensitivity*** and ***specificity*** through a proper setting of the number of compounds in the four sets.

Why DemFeature works better for the public than for the private test set

In this work, the DemFeature models produce better prediction results in the public than in the private test set, which is in contrast to the results of DemPred and of the participants in the Kaggle™ competition. Although optimization based on the public test set is advantageous in that better prediction results can be obtained, we have discovered through an internal observation on all of the prediction results of both public and private test sets (with all parameter candidates prepared to be optimized) that the DemFeature algorithm indeed predicts data from the public test set more easily than from the private test sets. To understand this different behavior, we consider for each compound ***k*** of the test set the most similar compound in the training set, whose similarity value s_k is given by Equation (3.2)

$$\max_n(\hat{\vec{x}}_k \cdot \hat{\vec{x}}_n) = s_k \tag{3.2}$$

$\hat{\vec{x}}_k$ is the feature vector of compound ***k*** in the test set and $\hat{\vec{x}}_n$ is the feature vector of compound ***n*** in the training set. The features are normalized according to Equation (2.35). The average of the s_k values over all compounds of the public or private test set is given by Equation (3.3).

$$\langle S \rangle = \frac{1}{N_{test}} \sum_{k=1}^{N_{test}} s_k \ , \tag{3.3}$$

Where N_{test} is the number of compounds in the considered test set.

For the public test set $\langle S_{public} \rangle$ = 0.8191 while for the private test set $\langle S_{private} \rangle$ = 0.8174. Hence, $\langle S_{public} \rangle$ is slightly larger than $\langle S_{private} \rangle$ such that the DemFeature prediction model, which focuses on the more similar compounds of the training set has an advantage in predicting the compounds of the public test set. However, another factor that fosters better prediction results for the public than for the private test set is due to the fact that the global parameters ***cutoff*** and λ_2 are optimized by using the public test set.

3.1.4 Conclusion

In this study, several approaches were applied to the classification task of the 2012 Kaggle™ competition launched by Boehringer Ingelheim Inc. for evaluating the prediction potential of our models within the pharmaceutical industry. The aim of the Kaggle™ competition was to explore the capacity of crowd computing for use within drug discovery and development. Hence, this competition used a curated experimental genotoxicity dataset, which is an important property in the detection of drug safety at the early stages of drug development. To avoid any possible expert bias, during the competition, all biological and chemical information was withheld from all participants. The reasons why we used this dataset to examine our models are: (i) it provides a realistic up-to-date prediction scenario for drug classification; (ii) the predictions submitted by different professional persons and groups certainly came close to the theoretical limits of what could be achieved for this prediction task. Here, it needs to be emphasized that the aim of this study is not to predict a biological problem, rather, it is to develop *in silico* methods used for better and more efficient evaluation of properties of unknown compounds for the process of drug development.

We used DemPred with linear and quadratic features and DemFeature-1 & -2 with linear features to build models with the training set to predict the target values of compounds from the private and public test sets, respectively. The DemFeature-1 model produces the best prediction results. It outperforms all results achieved for the public test set in the Kaggle™ competition. Based on the information provided by Boehringer Ingelheim Inc., all of the top-ranking models are built with random forest or its variants. Also the random forest benchmark added to the Kaggle™ contest data produces good prediction results, especially for the private test set. By contrast, the SVM benchmark produces relatively poor results. It seems that these Kaggle™ competition datasets are hard to be predicted by a simple linear classifier. However, the obvious improvement of prediction results using DemFeature-1 compared with results of SVM benchmark and DemPred, demonstrates that a well-designed training subset can improve the prediction quality of a linear classifier considerably. The DemFeature-2 algorithm is a variant of the DemFeature-1 algorithm. Although DemFeature-2 cannot produce better prediction results than DemFeature-1, it can significantly reduce the CPU time required and the fixed numbers of similar and dissimilar compounds in the training subset are perhaps helpful in explaining the connection between biological response and compounds. Overall, the DemFeature algorithms could be an alternative, especially in cases where problems are hard to be resolved by traditional linear classifiers.

3.2 Project 2: Drug-induced phospholipidosis prediction

Phospholipidosis (PLD) can be induced by administration of medicine for a long period, which is called drug-induced phospholipidosis. Actually, PLD is a lipid storage disorder in which complexes containing the drug and phospholipid accumulate within lysosomes in the living cell as lamellar inclusion bodies[94] which is the morphological hallmark of PLD. This complex has been found in a variety of tissue types[95, 96] such as lung, liver, brain, kidney, ocular tissue, heart, adrenal glands, hematopoietic tissue and circulating lymphocytes[97]. In preclinical studies, the human organs that most often showed characteristic lysosomes related with PLD are lungs and liver[98]. Apart from human beings, Drug-induced PLD also occurs in other species, such as rodents[99]. PLD can cause histological changes in tissues such as foamy macrophages, which can be observed with light microscopy[98] (Figure 3.14). These histological changes are sometimes also considered as markers that indicate whether a drug may cause PLD. Nevertheless, the gold standard method to determine whether a drug is capable of inducing PLD is the transmission electron microscopy (TEM), which is used to confirm the presence of multilamellar bodies within lysosomes[97] (Figure 3.15 & 3.16)

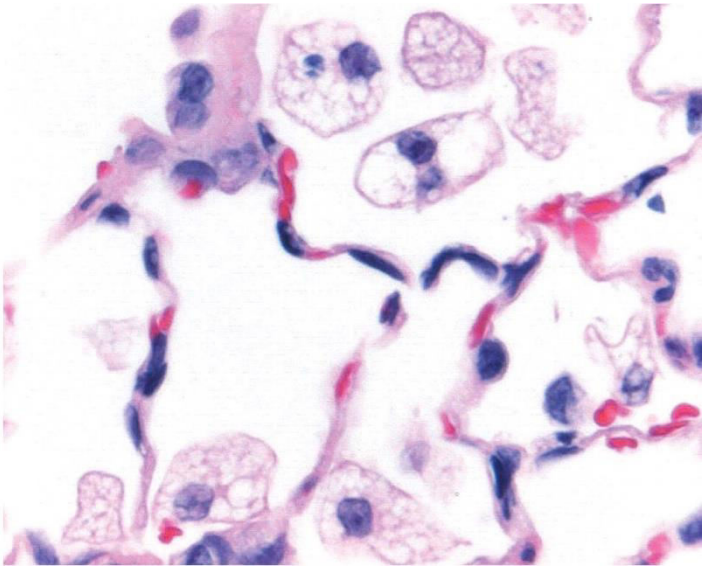

Figure 3.14: Pulmonary phospholipidosis: foamy macrophages in the cell of lung observed with light microscopy. Original figure by Chatman *et al.* [98].

Currently, a number of marketed drugs have been verified as causing PLD *in vitro* or *in vivo*[101,102]. The majority of PLD-inducers possess a cationic amphiphilic structure. Those drugs are also called Cationic Amphiphilic Drugs (CADs). The first clinical case of CADs-induced PLD was reported in 1971 in which multilamellar bodies were confirmed by TEM in several tissues of Japanese patients who were treated with the antianginal medication 4,4-

diethylaminoethoxyhexestrol (DH)[99]. CADs possess two typical structural features: a rigid hydrophobic moiety and a hydrophilic amine group that is charged as cation under physiological conditions[103]. The hydrophobic moiety of CADs usually is an aromatic ring. Slavov *et al.*[104] proposed a toxicophore model summarized by the structural features of CADs. In these compounds, the distances between the centroid of one hydrophobic ring and the hydrophilic group (amino group) are between 0.35 nm and 0.75nm. In addition, a second hydrophobic ring, usually an aromatic ring is present, at a distance of 0.55nm to 0.70nm from the amino group. The third feature is that the distance between the centroids of two ring structures is 0.40nm-0.50nm.

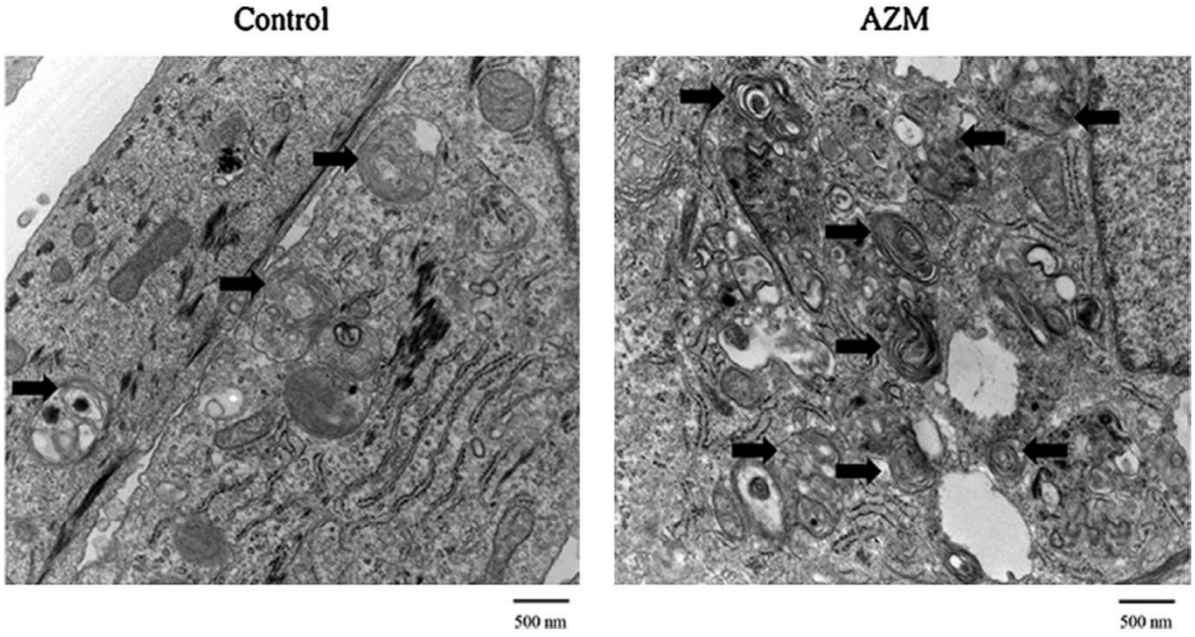

Figure 3.15: TEM images of phospholipidosis induced by azithromycin (an antibiotic) in human meibomian gland epithelial cells. The left image is the control test. On the right image, the dark onion-shaped parts are lamellar bodies indicated by arrows. Original figure by Liu *et al* .[100].

Figure 3.16: The TEM image of phospholipidosis in soft tissue. The multi-lamellar structure is clearly visible. Original figure by Chatman *et al.*[98].

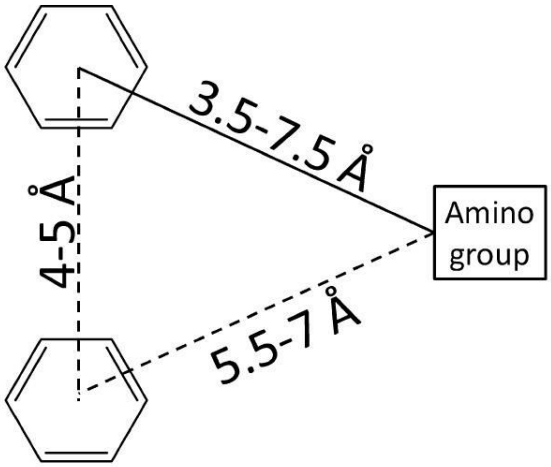

Figure 3.17: The toxicophore associated with PLD-inducers proposed by Slavov *et al.*[104].

Although CADs play an important role in inducing PLD, not all CADs can induce PLD, according to the FDA PLD database[102]. In the report by Choi *et al.*[105], they also stated not all PLD inducers are CADs. There are several non-CAD chemical species which are capable of causing PLD such as aminoglycoside, aminocyclitol and macrolide antibiotics.

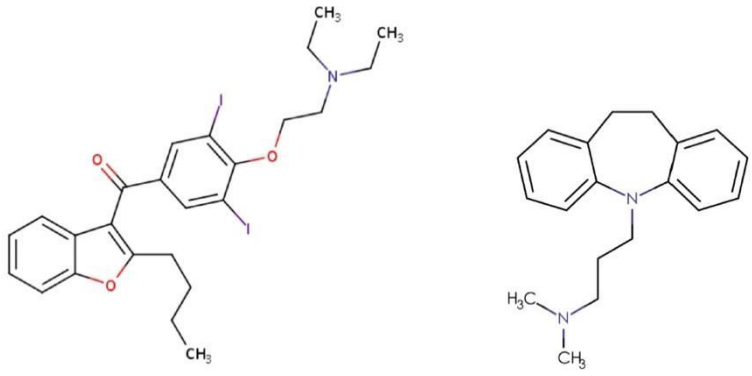

Figure 3.18: Both amiodarone and imipramine are CAD PLD-inducers, which have been tested *in vivo*[106]. Original figures were retrieved from DrugBank[107].

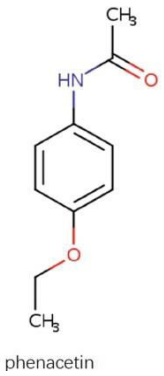

Figure 3.19: phenacetin is a non-CAD PLD-inducer. Original figures were retrieved from DrugBank[107].

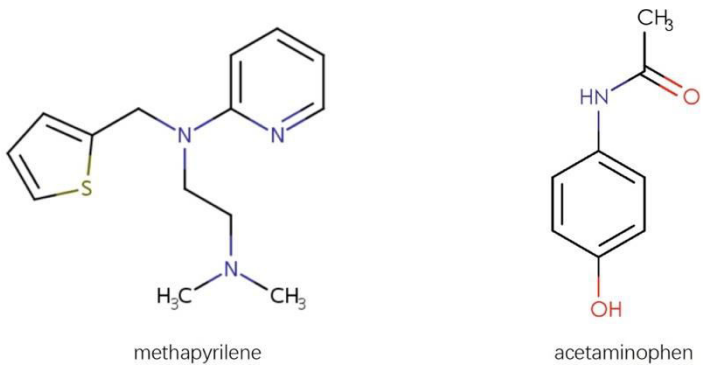

Figure 3.20: both compounds are PLD non-inducers, of which metapyrilene is a CAD compound and acetaminophen is a non-CAD compound. Original figures were retrieved from DrugBank[107].

Generally, PLD is considered as an adaptive response. This syndrome is reversible once the patient stops taking the related drugs[103]. In 2004, the US FDA established a phospholipidosis working group to investigate drug-induced PLD related toxicity[54]. So far, there is no direct

evidence indicating that PLD harms human health. Liu *et al.*[100] even found drug-induced PLD to have a beneficial effect, which can be useful in the treatment of meibomian gland dysfunction. However, PLD itself is related to the phospholipid disorder and is likely to have concomitant toxicological phenomena in affected organs[106]. For example, researchers proved that PLD is associated with several genetic conditions such as Niemann-Pick and Tay-Sachs diseases[108,109]. Additionally, two reports pointed out several PLD-inducing compounds also cause concurrent inflammatory and / or degenerative changes in tissues[110,111]. Therefore, for drug development, especially pharmaceutical companies, this uncertain harm cannot simply be ignored. Considering the reason of drug safety, there is a demand to identify PLD-inducers at an early stage of drug development to avoid huge efforts in both time and money.

The precise mechanism of drug-induced PLD remains unclear. There are several possible mechanisms for drug-induced PLD proposed by Sawada *et al.*[109]: 1. The activity of phospholipase is inhibited by drugs; 2. the lysosomal enzyme transport is inhibited by drugs; 3. The capacity of phospholipid biosynthesis in tissues is enhanced; 4. The capacity for synthesizing cholesterol is strengthened. Fischer *et al.*[96] also mentioned in their article that the formation of PLD is possibly induced by several mechanisms. Apart from the unclear mechanism of induction, it has been found that a PLD-inducer may perform differently under different conditions such as species, tissue type, age, sex etc.[112]. In addition, it has been found that some compounds that induce PLD *in vivo* do not induce PLD in *in vitro* tests, while the compounds demonstrated to induce PLD *in vitro* may not cause PLD *in vivo*[103,106]. According to the report written by Morelli *et al.*[112], gentamicin is capable of inducing PLD in both human and animal but in cell-based assays, the PLD induction effect is not demonstrated.

In summary, it can be said that is not easy to identify and predict PLD reliably. Currently, TEM is still the most reliable method to confirm PLD. However, it is a labor intensive method, which cannot be used to screen a large number of compounds in a short time. For drug development, TEM is difficult to apply in the early stage to investigate a large number of unknown compounds. Therefore, in past decades, scientific communities have made huge efforts to develop a variety of *in vitro* and *in silico* methods for predicting PLD.

To date, there are a number of *in vitro* PLD prediction assays based on cells, which are combined with different methods such as electron microscopy or fluorescent probes such as Nile red[97]. However, these cell based methods depend on the concentration and the cell line used. In addition to *in vitro* cell based methods for predicting PLD, researchers have also developed *in vitro* non-cell based methods for assessing the potential for inducing PLD.

Vitovic *et al.*[113] used the critical micelle concentration of short-term acidic phospholipids, using the surface tension activity to assess the PLD potential of drugs. Jiang *et al.*[114] employed a chromatography approach to screen PLD-inducers, which utilized immobilized artificial membrane chromatography and electrokinetic chromatography with surfactant unilamellar vesicles as the pseudostationary phase. Compared with *in vivo* methods, *in vitro* methods are relatively high-throughput methods. Nevertheless, a crucial point needs to be considered, as we mentioned above, the drugs that induce PLD *in vitro* may not demonstrate the same effect *in vivo* and vice versa.

in silico methods for predicting PLD have also attracted the attention of pharmaceutical scientists. Several *in silico* tools had been developed to assess drug's potential for inducing PLD. Ploemen *et al.*[115] only used *C*log*P* and pK_a to build up a simple equation to assess the PLD-inducing potential. Based on the works of Ploemen *et al.*, Tomizawa *et al.*[116] used net charge (NC) to replace pK_a and they found this modification can improve performance. Furthermore, Pelletier *et al.*[117] still used *C*log*P* and pK_a as prediction parameters but modified his rules slightly to upgrade the Ploemen model obtaining better prediction performance. In addition, Hanumegowda *et al.*[106] found that a pharmacokinetic parameter, the volume of distribution, can strengthen the Ploemen model to predict occurrence of PLD. In contrast to the Ploemen model and other related models, Przybylak *et al.*[97,118] utilized characteristic structural alerts to construct a prediction model. Ivanciuc[119] employed several excellent machine learning algorithms to built statistical models to predict a drug's PLD inducing potential.

3.2.1 *in silico* methods assessing the potential of drug inducing PDL

in vitro and *in silico* methods have their pros and cons. Chatman *et al.*[98] suggested that the two kinds of methods need not be mutually exclusive. They proposed three tiers in their publication to reduce the risk of PLD issues. In this three-tier program, they suggested the first tier should utilize *in silico* tools to screen lead compounds and then be supplemented with *in vitro* assays to verify minimal PLD liability. In fact, *in silico* methods are a low-cost tool and their biggest advantage is that compounds can be verified before chemical synthesis, which is very useful when verifying artificially designed compounds. In this section, a detailed description is given to introduce several popular *in silico* methods.

3.2.1.1 Ploemen model for predicting PLD

In 2004, Ploemen *et al.*[115] proposed an equation to rapidly discriminate possible PLD inducers among candidate compounds. This equation was concluded by an investigation of PLD inducers and they found those had high *C*log*P* and pK_a values. Thus, they derived a condition shown as Equation (3.4) below. Using this equation, if the sum of the squared *C*log*P* value and the squared pK_a value is greater than 90 and both *C*log*P* > 1 and pK_a > 8, then this compound is identified as a PLD-inducer. Here, *C*log*P* reflects hydrophobic characteristics and pK_a reflects the degree of ionization. In case the compound possesses two or more than two titratable groups the larger the pK_a is, the more basic is considered. On the other hand, if the physicochemical properties of a compound are not fulfilled by these criteria, then it is identified as PLD negative. This simple model is explicit and corresponds to the features of CADs. However, this model does not provide a deep understanding of the prediction ability. In particular, it is hard to produce a high prediction performance for non-CAD PLD inducers.

$$\text{If } (pK_a)^2 + (C\log P)^2 \geq 90 \text{ , and } pK_a \geq 8 \,\&\, C\log P \geq 1 \quad \text{(3.4)}$$
$$\text{the compound is predicted as PLD inducer}$$

3.2.1.2 Pelletier model: modified Ploemen model

Pelletier *et al.* [117]optimized the model proposed by Ploemen *et al*. In their work, they plotted a Spotfire figure, which reflects the relationship of *C*log*P* and pK_a for each compound. By using this figure, they made efforts to manually adjust parameters for selecting optimized values to separate positive and negative compounds. Finally, they derived a new rule (see Equation (3.5)) to predict drug-induced PLD by slightly modifying the Ploemen model. For a given test compound, if the sum of the squared values of both *C*log*P* and pK_a exceeds the threshold value of 50 and *C*log*P* ≥ 2 and pK_a ≥ 2, then it can be judged as a PLD-inducer. Conversely, if this sum is smaller than the threshold, then it can be considered a PLD negative compound. In their report, they prepared a dataset consisting of 201 compounds (85 positive compounds and 116 negative compounds) to compare their model with Ploemen model. The prediction result of the Ploemen model is: *Specificity* = 0.77, *Sensitivity* = 0.74 and *accuracy* = 0.75; the prediction performance of modified Ploemen model is: *Specificity* = 0.85, *Sensitivity* = 0.79 and *accuracy* = 0.82. Obviously, the modified model improved the prediction performance compared with the Ploemen model.

$$\text{If } (pK_a)^2 + (C\log P)^2 \geq 50 \text{ , and } pK_a \geq 6 \,\&\, C\log P \geq 2 \quad \text{(3.5)}$$
$$\text{then the compound is predicted as PLD inducer}$$

3.2.1.3 Tomizawa model

The Tomizawa model is another variant of the Ploemen model. In 2006, Tomizawa *et al.*[116] proposed to use the Net Charge (*NC*) to substitute pK_a value of the original model. They suggested that for the case of zwitterions, the positive charge from the high pK_a basic functional group can be counteracted by the low pK_a acidic functional group. Therefore, the *NC* is a better choice to reflect ionization of compounds in organelles. In their test, the *NC* of compounds was calculated at *pH* 4.0 because this value is close to the *pH* in lysosomes. Finally, based on *NC* value and *C*log*P* value, they provided a rule to judge PLD induction potential. Using the following prediction model: if *C*log*P* > 1 and $1 \leq NC \leq 2$ for a given compound, it is a PLD inducer. Compounds with *NC* > 2 were not included in their dataset. Furthermore, they provided a criterion for PLD risk ratings of compounds shown in Equation (3.6). In their publication, a total of 63 compounds were used to test the proposed method. The prediction *accuracy* reaches 98.4% (62/63). Of these 63 compounds, 33 compounds constitute the initial test set, which has been verified by TEM. The comparison with the Ploemen model based on these 33 compounds illustrated that the Tomizawa model improved prediction performance.

$$\begin{aligned}
&\text{If } NC < 1, \text{ then PLD negative} \\
&\text{If } NC = 1 \text{ and } C\log P < 1.61, \text{ then PLD positive, low risk} \\
&\text{If } NC = 1 \text{ and } C\log P \geq 1.61 \text{ and } <2.75, \text{ then PLD positive, medium risk} \\
&\text{If } NC = 1 \text{ and } C\log P \geq 2.75 \text{ or } NC \in (1,2] \text{ then PLD positive, high risk}
\end{aligned} \tag{3.6}$$

3.2.1.4 Hanumegowda model

Since PLD occurrence may result from the residence of compounds in tissues, Hanumegowda *et al.*[106] added a pharmacokinetic parameter, the volume of distribution (V_d), which is an important factor reflecting the presence of residual compounds in tissues. They deduced that this parameter can be used to screen PLD-inducers from PLD non-inducers. They proposed a new criterion to predict the PLD induction potential, shown as Equation (3.7) below. This equation combines V_d with *C*log*P* and pK_a values together. In their publication of Hanumegowda *et al.* applied this proposed method to predict 101 compounds (51 positive compounds and 50 negative compounds). They also used the Ploemen model to predict the same dataset for comparison. The *accuracy* of the Ploemen model for these 101 compounds was 77%, while the Hanumegowda model gave an *accuracy* of 88%. Moreover, the Tomizawa model was also used for comparison but they removed 3 compounds whose *NC* values were above 2.0. Therefore, the *accuracy* of the Tomizawa model is 82% based on 98 compounds. Based on those results, they said that their model can predict the PLD induction

potential better than the other two models. However, this model is still a relatively empirical model. More importantly, it is quite difficult and expensive to measure V_d values as these need to be tested *in vivo* in animals. Hanumegowda also mentioned that calculated V_d values would render their model less reliable.

$$\text{If } pK_a \times C\log P \times V_d \geq 180 \text{ , and } C\log P \geq 2 \quad \text{then the compound is predicted as PLD inducer} \tag{3.7}$$

3.2.1.5 SMARTS models

Przybylak *et al.*[97] proposed an *in silico* to model predict the PLD induction potential of drugs, which is based on several characteristic structural fragments of PLD-inducers to identify inducers and non-inducers of PLD. The workflow of this model is a like a decision tree, including three steps to screen compounds as shown in Figure 3.21. The first version of the SMARTS model was derived from the database made by Kruhlak *et al.*[95]. Based on this database, 32 structural patterns (called SMARTS patterns in their publications because in their model, all fragments are described in SMARTS strings) were developed to screen possible PLD inducers. In addition to those 32 structural patterns, patterns characterizing ring systems, carboxylic acids and nitro groups were also included in SMARTS model. In 2014, Przybylak *et al.* updated the SMARTS model by adding 7 new structural patterns concluded from the US FDA PLD database published in 2012 consisting of 743 compounds[118]. The new proposed SMARTS model has been shown to be better than the first version. A detailed list of SMARTS patterns is shown in Appendix 2.

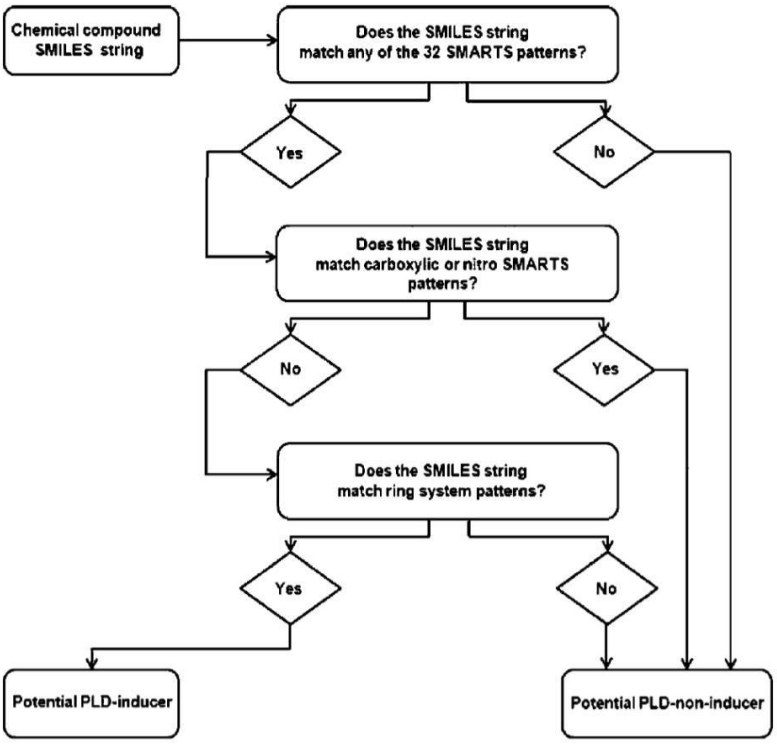

Figure 3.21: Workflow of the original SMARTS model. In this figure, the first phase only includes 32 structural patterns and it has been updated to 39 patterns in the second version. Original figure from Przbylak *et al.* [117].

In the SMARTS model, the first step is to input a given compound as a SMILES string and to check whether there are matches with the 39 structural patterns of the model. If a given compound possesses matches with one or more of those 39 structural patterns, the compound can be judged to be a possible PLD inducer. If not, it is directly classified as a PLD non-inducer. The possible PLD inducer goes into the second phase, if it matches carboxylic or nitro SMARTS patterns, then this compound is classified as the PLD non-inducer. If not, this compound is still a possible PLD inducer and goes into the final phase for judgment. In the final judgment phase, the possible PLD inducer needs to match with ring system patterns. If it matches with them, then it is predicted to be a PLD inducer; if not, it is considered to be a PLD non-inducer. It is worth noting that ring patterns are important features describing the hydrophobic moiety of PLD inducers, but they are not used in the first phase because many PLD non-inducers also possess ring systems. However, these patterns are used in the later phase for avoiding false positives. When investigating the prediction performance, Przybylak *et al.*[97], compared the SMARTS model with the Ploemen model, Pelletier model and Hanumegowda model. They found that the prediction performance of the SMARTS model is better than the other three models.

3.2.2 Phospholipidosis database

For *in silico* prediction, the quality of the database is a problem that must be considered because the prediction performances of the models depend critically on the datasets used for training. Currently, many PLD databases have been published by researchers. However, an *in silico* PLD prediction model cannot be better than the experimental dataset used to train it. In addition to technical limitations, there are several other problems in those datasets that prevent the construction of excellent models.

Published PLD databases come from different species because the number of compounds proved to induce PLD in human is limited. However, as mentioned above, some compounds induce PLD in animals but not in humans. Another point of concern is that the metabolism plays a key role in inducing PLD[54]. For some drugs, the drug itself cannot induce PLD but the metabolites of the drugs generated *in vivo* induce PLD. Unfortunately, few databases have considered the influence of such biotransformation. This is a major reason why *in silico* models do not provide correct predictions for some compounds. For example, the ketoconazole itself cannot induce PLD but its major metabolite, de-N-acetyl-ketoconazole(DAKC), can induce PLD.

Moreover, in many PLD databases, numerous so called PLD non-inducers were actually not tested for PLD induction. The reasons why they were defined as PLD negative compounds is simply that no reports exist indicating those compounds are related to PLD induction[54,120]. Many PLD positive compounds listed in the databases are only related to the presence of foamy macrophages, cytoplasmic vacuolations, cytoplasmic granules, lipidosis, dyslipidosis, histiocytosis and so on[54]. Those physiological phenomena are not reliable indications for confirming the occurrence of PLD. Detecting lamellar bodies in the lysosomes by TEM is still the most reliable method for judging whether a compound can induce PLD. Considering these reasons, in this study, we decided to use a curated database to build our models. Such a database was provided by Goracci *et al.*[54] Furthermore, we also used an independent test set provided by Boehringer Ingelheim to validate our models. However, it needs to be noted that the independent test set was only verified by an *in vitro* method and it was not confirmed using the TEM method.

3.2.2.1 Goracci phospholipidosis database

To provide a high quality PLD database, Goracci *et al.*[54] carefully analyzed seven popular databases listed in Table 3.13. To obtain a more reliable database, they corrected errors and

discrepancies appearing in those databases and deleted doubtful compounds because the PLD assignments of many compounds are inconsistent in different databases. In addition, they also considered metabolism and other factors influencing PLD induction properties. This rigorous analysis leads to a curated PLD database consisting of 331 compounds, which only contains disclosed compounds belonging to those seven databases. To date, this 331 compounds database is the most reliable PLD database. In our research, this database was used to examine the prediction performance of our *in silico* models.

Table 3.13: The seven PLD databases investigated by Goracci *et al.* [54]

author, year	database abbreviation	PLD+ compounds	PLD- compounds	unique compounds [a]
Orogo *et al.*, 2012[102]	O2012	215	232	262
Tomizawa *et al.*, 2006[116]	T2006	35	17	2
Pelletier *et al.*, 2007[117]	P2007	56	61	0
Vitovic *et al.*, 2008[113]	V2008	34	18	0
Lowe *et al.*, 2010[120]	L2010	99	82	2
Hanumegowda, *et al.* 2010[106]	H2010	38/33[b]	42/9[b]	3
Fischer *et al.*, 2012[96]	F2012	27/23[b]	5/9[b]	2

a. Compounds that are reported in that database only.
b. numbers refer to *in vitro* / *in vivo* data, respectively. Although some databases do not contain new compounds, they provide useful information.

The compounds used in this research are all disclosed compounds.

Confirming the presence of lamellar bodies in lysosomes with TEM is the gold standard for examining the PLD induction potential of drugs. Therefore, we further investigated those 331 compounds in the Goracci database[54]. We divided those compounds into two categories: TEM-confirmed definite compounds and TEM-confirmed unclear compounds. A detailed discussion of analyzed results will be reported in a later section.

Of the seven databases used to compile the Goracci database, only the four databases, namely, Orogo database, Tomizawa database, Pelletier database and Lowe database, clearly claim information related to TEM confirmation as shown on Figure 3.22. Lowe *et al.*[120] collected 185 compounds containing 102 PLD inducers and 83 PLD non-inducers. From the 102 PLD inducers, 68 compounds are TEM-confirmed and the remaining 34 compounds were reported to be PLD inducers because foamy macrophages or vacuolatins were observed in histopathological tests. Unfortunately, Lowe *et al.* did not provide a name list of TEM-confirmed compounds. Therefore, for 102 PLD inducers, it is impossible to know the specific names of TEM-confirmed PLD inducers. However, in the Lowe database, the 83 PLD non-inducers have all been confirmed by TEM.

In 2012, Orogo *et al.*[102] constructed an updated version of PLD working group database of US FDA, which was based on the old version of the PLD database[95] and added compounds from literature and according to information from pharmaceutical companies. The updated version of the database contains 447 disclosed compounds, which is the largest dataset of those seven databases considered by Goracci *et al*.[54]. The Orogo database classified all compounds into two categories: *high-confidence* and *medium-confidence*. The *high-confidence* PLD inducers are TEM-confirmed compounds while the *medium-confidence* PLD inducers were determined histopathological by presence of foamy macrophages, cytoplasmic vacuolations, cytoplasmic granules, lipidosis, dyslipidosis or histiocytosis. For PLD non-inducers, they investigated FDA documents. If PLD keywords were absent for a compound in the New Drug Application (NDA) documents, it was classified as *high-confidence* compound. For the *medium-confidence* compound, PLD keywords were not found in the Investigational New Drug (IND) documents.

The Pelleter database provides 117 public compounds including 56 PLD inducers and 61 PLD non-inducers. All 56 PLD inducers have been confirmed by TEM. In addition, in the report of Tomizawa *et al*., they also verified 23 compounds using the TEM method. However, it must be noted that the compounds collected in the four databases (Orogo database, Tomizawa database, Pelletier database and Lowe database) include duplicate compounds.

Based on the information in those four databases, it can be concluded that in the Goracci database[54], 178 compounds are TEM-confirmed. However, it cannot be claimed that the remaining 153 compounds are not TEM-confirmed (i. e. they are TEM-confirmed unclear), since we only took the information from the seven datasets cited by the Goracci database and did not perform an independent large-scale literature search. Hence, we consider only those 178 compounds to be TEM-confirmed, while it is unclear for the remaining 153 compounds.

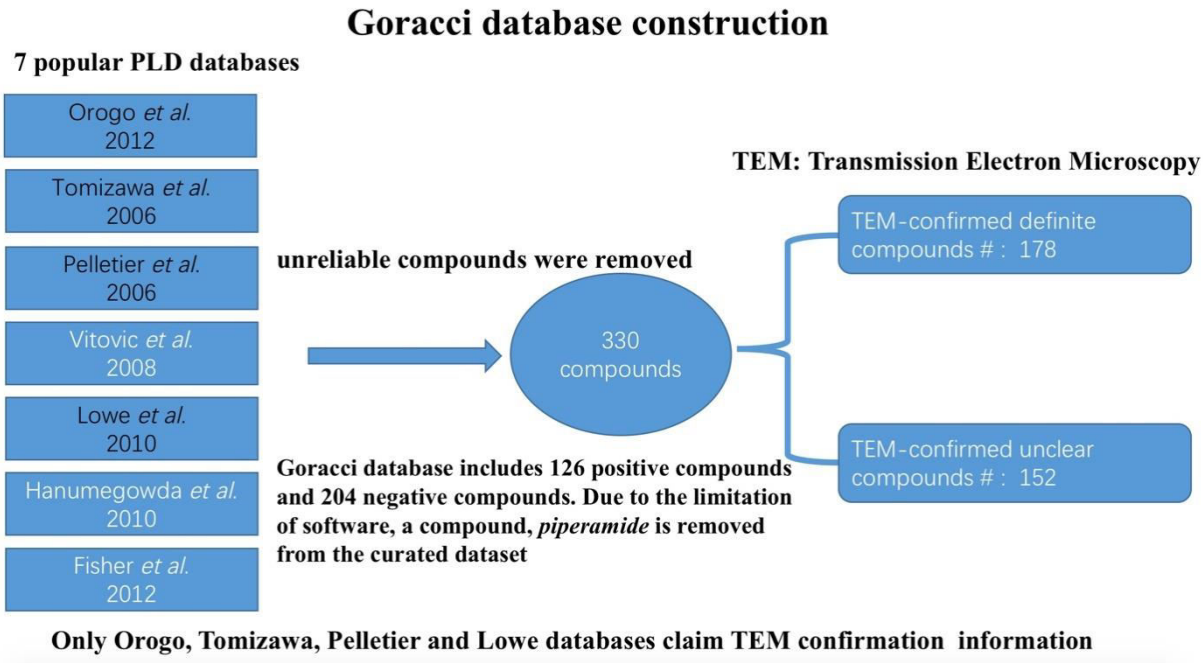

Figure 3.22: Construction of the Goracci database[54]. This database includes 331 compounds. However, one compound was removed because the molecular descriptors could not be generated with the software packages. From the remaining 330 compounds 178 are TEM-confirmed, while for 152 compounds it is unclear.

Due to software limitations one compound (*piperamide*) in the Goracci database[54] could not be characterized by molecular descriptors. Therefore, only 330 compounds were used for the present study. Details on the molecular descriptors are given in chapter 3.2.3. The information on the TEM-confirmed and TEM-confirmed unclear compounds are listed in Appendix 3.

3.2.2.2 Independent phospholipidosis test dataset

For this study, an independent test set was prepared to evaluate the performance of our models to predict the PLD induction potential. This independent test set contains 133 compounds, of which 72 are PLD positive and 61 PLD negative. The PLD measurement was based on *in vitro* cell assays. The dataset was provided by courtesy of Boehringer Ingelheim. Therefore, structure information of the compounds cannot be disclosed here. An *in vitro* fluorescent phospholipid-based assay was used to verify the compounds in the independent test set at Boehringer Ingelheim. This method was introduced first by Nioi *et al.*[108].

The assay setup can be briefly described as follows. HepG2 cells, which are human liver tissue cells, were seeded into 96 well plates at a density of 5,000 cells per well in 100 µl media and allowed to attach overnight. To each well was added 50 µl of the HCS

LipidTOXTM red reagent (Life Technologies; diluted 1:500 in normal growth medium) and 50 µl of the test article or control (diluted in normal growth medium). After 48 h, cells were fixed in 4% formaldehyde in PBS containing Hoechst 33341 for 20 min at 37 °C. Following washing, the fluorescence was measured using an Arrayscan VTi high content analysis reader (Thermo Scientific). For each assay, amiodarone (Sigma) was used as a positive control and aspirin was used as a negative control. Compounds and controls were typically assayed at 9 concentrations ranging from 0.78 to 200 µM. Compounds showing a dose-responsive increase of LipidTOX fluorescence intensity that is equal to or greater than 2.5-fold of the concurrent vehicle control at non-cytotoxic concentrations (>75% viability of concurrent vehicle control), is considered a positive response in this assay.

Table 3.14: Overview of molecular descriptors used for PLD.

feature groups	number of features
MoKa_pKa_logD[122]	5
ECFP4[123]	1024
Estat[124]	114
Fragment[130-132]	1162
Ghose Grippen[125]	120
MOE Descriptors[67]	185
PipelinePilot[126]	68
MDL2DKeys[127]	166
Sterimol[128]	12
Talete[129]	5571
total	**8427**
features used[a]	**3849**

a. After deleting duplicate features, the number of features used is 3849.
The molecular descriptors were provided by courtesy of Dr. Jörg Bentzien at Boehringer-Ingelheim.

3.2.3 Molecular descriptors for PLD

So far, it is not clear yet which descriptors of compounds are most important to identify their ability to induce PLD. Nevertheless, many researchers priories molecular descriptors relating to CADs because the majority of drugs that are verified as inducing PLD are CADs. But, not all PLD inducers are CADs. Therefore, in this study, we employed 10 different software packages to generate molecular descriptors and topological fingerprints for covering chemical diversity as much as possible. Thus, a total of 8427 molecular descriptors were calculated for each compound using the software MoKa[122], ECFP4[123], E-states[124], Fragment Descriptors[130-132], Ghose Grippen[125], MOE descriptors[67], Pipeline Pilot[126], MDL2Dkeys[127], Sterimol[128], Talete[129]. Table 3.14 lists information on the molecular descriptors.

For the compound *piperamide* in the Goracci database[54], features could be calculated due to limitations of the software. Therefore, this compound was excluded from the database. In addition, for *zidovudine* and *azaserine* three features, DRV, DSA and DSB, failed to be calculated. In this case for both compounds, feature values that are averages over the other compounds was used as substitute. To save CPU time and facilitate the identification of important features, duplicated features were removed from the database. As a result, 4573 molecular features were deleted, leaving a final number of 3849 features. To create a proper predictive environment for the independent test set, the same 3849 features were used.

3.2.4 Results

In this section, we depict the prediction results of several approaches applied to PLD datasets, including the Goracci database[54] and the independent test set. Both DemPred (see Chapter 2.9) and DemFeature-1(see Chapter 2.10.1) were utilized to predict the Goracci database (Chapter 3.2.2.1). In addition, in order to investigate whether the SAMRTS features are suitable for our models to predict PLD, the SMARTS features were also used to constitute the DemFeature-1 model. Moreover, for further validating our methods, both DemPred and DemFeature-1 models were also used to predict the independent test dataset (3.2.2.2). Before constituting DemPred and DemFeature-1 models, the features of all datasets have been normalized by Z-score (Equation (2.5)). In addition to our models, several popular models, namely, the Ploemen model(see Chapter 3.2.1.1), the Pelletier model (see Chapter 3.2.1.2), and the original and updated SMARTS models (see Chapter 3.2.1.5), were used to compare with our prediction models in this section.

Table 3.15: Number of positive and negative PLD compounds in the five subsets used for cross validation.

index	test set		training set		NumFea[a]
	positive	negative	positive	Negative	
1	17	49	109	155	3780
2	28	38	98	166	3790
3	24	42	102	162	3781
4	30	36	96	168	3790
5	27	39	99	165	3785

a. The original number of features is 3849. However, different training subsets have different numbers of features that do not vary. Therefore, the number of features is different for each subset.

3.2.4.1 Prediction of Goracci database

As mentioned in the Chapter 3.2.3, the modified Goracci database[54] used in our research involves 330 compounds characterized by 3849 features. To fully utilize each compound in the dataset to optimize the prediction performance of our models, a 5-cross validation technique (see Chapter 2.3) was applied to the dataset. During cross validation, each left out subset was not used for modeling so that the test set information was not used at all. Thus, each subset selected randomly from the whole dataset by cross validation represents an external dataset so the whole dataset can be truly predicted to efficiently assess the prediction performance of the models. Table 3.15 details the subsets separated from the whole dataset. For training, the ratio of positive and negative compounds was fixed at 1:1.5. Features whose values do not vary over all considered compounds cannot contribute to prediction and are therefore removed.

3.2.4.1.1 The prediction results of DemPred

The five training subsets were utilized to construct the DemPred models (see Chapter 2.9) to predict the corresponding test subsets. The DemPred models were built using the L1 & L2 two-step method (see Chapter 2.8). As applied to the datasets of the Kaggle™ competition (see Chapter 3.1), first the L1 approach was used for feature selection to reduce the complexity of the prediction problem. The L1 approach can identify the features that contribute only insignificantly to the prediction performance. After the L1 feature selection, the remaining features were further processed by L2 regularization. The L2 approach slightly adjusts the weight values of the features rather than completely deleting them. The efficiency of L1 & L2 approach depends on a proper selection of the λ values. Therefore, the choice of these values is important when using the L1 & L2 methods. The strategy to find the optimized λ values is different from the application for the Kaggle™ datasets. Here, for each combination of four subsets (listed in Table 3.15) that are used for training a ten fold-cross validation was performed to find the optimal λ values (described in Chapter 2.3) from a given set of values. These are $\lambda_1 \in$ {0.003, 0.004, 0.005, 0.006,….., 0.150, 0.152, 0.153, 0.155, 0.160, 0.162, 0165, 0.170} and $\lambda_2 \in$ {0.03, 0.04,… 0.10, 0.12, 0.15, 0.20, 0.25, 0.30,…, 0.37, 0.40, 0.45, 0.50, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.90}.

Since the project is a classification task, the one sided log Lorentzian (Equation (2.27)) was employed as loss function to construct the objective function because it can match the 0-1 indicator function. With this loss function the deviation between the values of the scoring

function (Equation (2.10)) and corresponding biological response values (-1 and +1) are smaller and the prediction results are less sensitive to outliers in the datasets. In this research, as shown in Table 3.15, 5 training subsets from the Goracci database[54] were used to build the DemPred models with the two-step L1 & L2 approach to predict the test subsets. The test subsets are external datasets - not used for model building - to test the prediction performance of the model.

As mentioned in Chapter 3.2.4, a total of 3849 features was used to constitute the prediction models. The prediction results of the DemPred model built with the L1 approach for 5 subsets of the Goracci database are shown in Table 3.16. The comparable results of 5 subsets demonstrated that the subset 4 has the worst prediction result (only 0. 391 by ***MCC*** and 69.7% by ***accuracy***). The main reason is that its ***sensitivity*** is significantly worse than other ones. The subset 5 gives the best results 0.654 by ***MCC*** and 83.8% by ***accuracy***. As a whole, the ***MCC*** is 0.532 and ***accuracy*** is 77.9%.

Table 3.16: Prediction results for the five subsets (Table 3.15) by L1 feature selection (λ_2=**0.0**).

	λ_1[a]	# features[b]	MCC	accuracy	sensitivity	specificity
subset 1	0.028	155	0.515	78.8%(52/66)	0.765(13/17)	0.796(39/49)
subset 2	0.045	143	0.578	78.8%(52/66)	0.821(23/28)	0.763(29/38)
subset 3	0.055	106	0.551	78.8%(52/66)	0.750(18/24)	0.810(34/42)
subset 4	0.075	56	0.391	69.7%(46/66)	0.500(15/30)	0.861(31/36)
subset 5	0.02	194	0.654	83.8%(55/66)	0.778(21/27)	0.872(34/39)
average			**0.532**[c]	**77.9%(257/330)**	**0.714(90/126)**	**0.819(167/204)**

a. λ_1 is determined by 10-fold cross-validation on the training subsets.
b. The number of features after feature selection.
c. The **MCC** for the whole dataset is calculated by summing up ***TP***, ***TN***, ***FP*** and ***FN*** of all subsets.

For the L1 feature selection, it is worth noting that already with a small λ_1 value a large number of features can be deleted. For each subset, less than 5% of all features were retained (see Table 3.16). Interestingly, the prediction performance is not deteriorated by strongly decreasing the number of features and the model with a smaller number of features can provide even better prediction performance than one with a larger number of features.

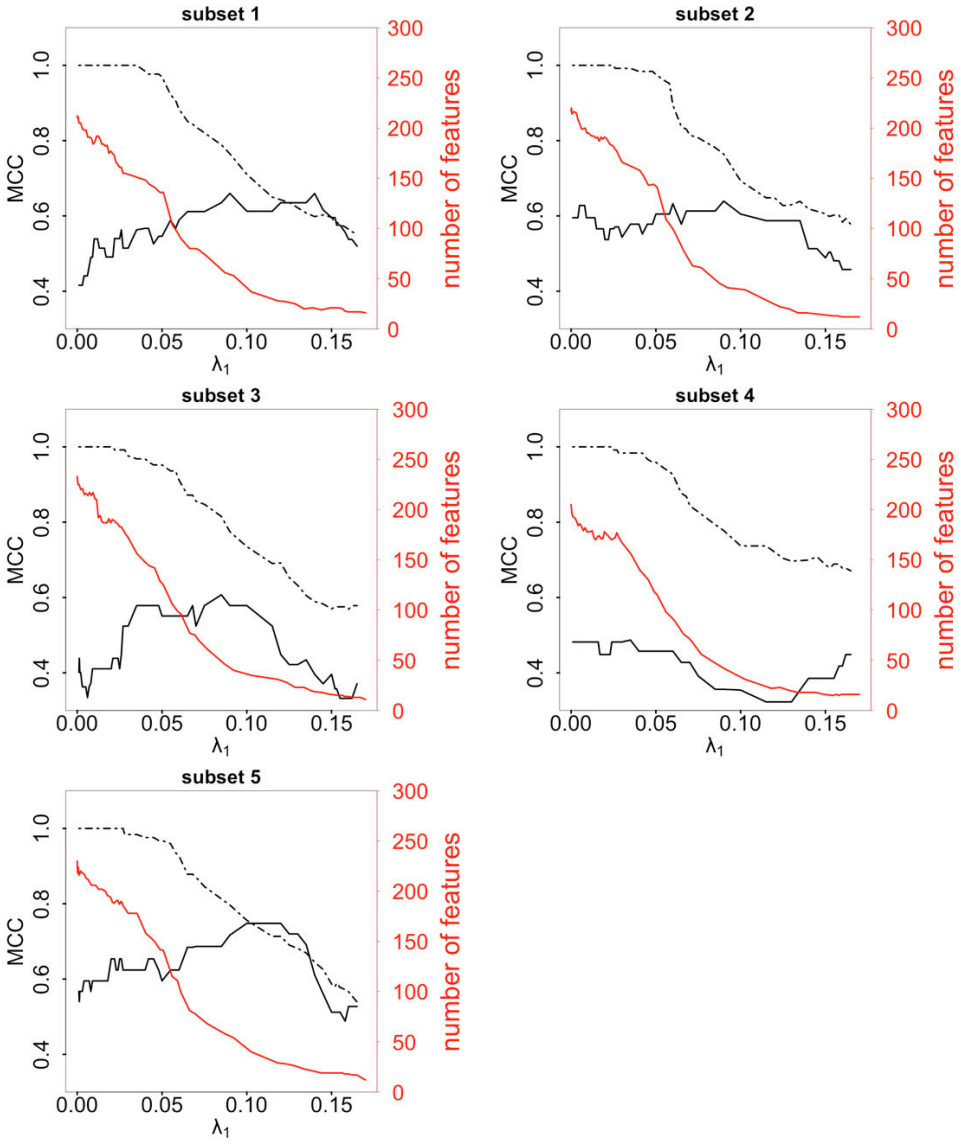

Figure 3.23: Prediction performances of the five subsets (Table 3.15) with L1 feature selection. Dashed lines represent training set (recall) and solid lines represent prediction performances of test subsets. The **red** line demonstrates the changing of number of features with increasing of λ_1 value. The left vertical axis (**black**) represents ***MCC*** performance and right vertical axis (**red**) represents the number of features retained. The horizontal axis represents the selected λ_1 values. The smallest λ_1 is 0.001. This figure was made by R v3.1.3.

Figure 3.23 shows the dependence of the prediction performance of DemPred and number of selected features as a function of the λ_1 value. This figure illustrates the relations between the λ_1 value and the number of selected features, which clearly reflects that the L1 approach has a strong effect for the sparsity of features. When the λ_1 is larger than 0.15, the number of features will be less than 20 for all five subsets. The highest prediction performance is given by a relatively small number of features. Except for subset 4 the prediction performance first increases with increasing λ_1 value before it finally decreases. For the subset 4, the L1 approach cannot efficiently improve the prediction performance.

After L1 feature selection, L2 regularization was employed based on the remaining features to build the DemPred model for the Goracci database. Table 3.17 shows the prediction results of the DemPred model. The prediction performances (***MCC*** and ***accuracy***) on subset 1, subset 3 and subset 4 are better than the results using only the L1 approach. For subset 5 the prediction results are the same for subset 2 they are slightly worse. The average prediction results are slightly better with additional L2 regularization yielding for ***MCC*** and ***accuracy*** 0.536 and 78.2%, respectively, as compared to 0.532 and 77.9% if only L1 feature selection is applied. The improvement is due to an increase in ***specificity*** from 0.819 to 0.828 while the ***sensitivity*** declines from 0.714 to 0.706. However, after L2 regularization, the prediction results of DemPred are better than the results of the Pelletier model (***MCC***: 0.534, ***accuracy*** 77.9%). In addition, it is worth noting that in this stage the prediction performances are not significantly influenced by λ_2 (see Figure 3.24) as the influence of λ_1 in the stage of L1 feature selection (see Figure 3.19).

Table 3.17: DemPred prediction results for the five subsets with L2 regularization (λ_1**=0.0**).

	λ^a	**# features**[b]	**MCC**	**accuracy**	**sensitivity**	**specificity**
subset 1	0.200	155	0.588	80.3%(53/66)	0.882(15/17)	0.775(38/49)
subset 2	0.300	143	0.571	78.8%(52/66)	0.785(22/28)	0.785(30/38)
subset 3	0.300	106	0.542	78.8%(52/66)	0.708(17/24)	0.833(35/42)
subset 4	0.350	56	0.398	69.7%(46/66)	0.466(14/30)	0.889(32/36)
subset 5	0.150	194	0.654	83.8%(55/66)	0.778(21/27)	0.872(34/39)
average			**0.536**[c]	**78.2%(258/330)**	**0.706(89/126)**	**0.828(169/204)**

a. λ_2 is determined by 10-fold cross-validation on training subsets as described in main text.
b. Number of features retained after L1 feature selection (see Table 3.16).
c. The **MCC** for the whole dataset is calculated by summing up ***TP***, ***TN***, ***FP*** and ***FN*** of subsets.

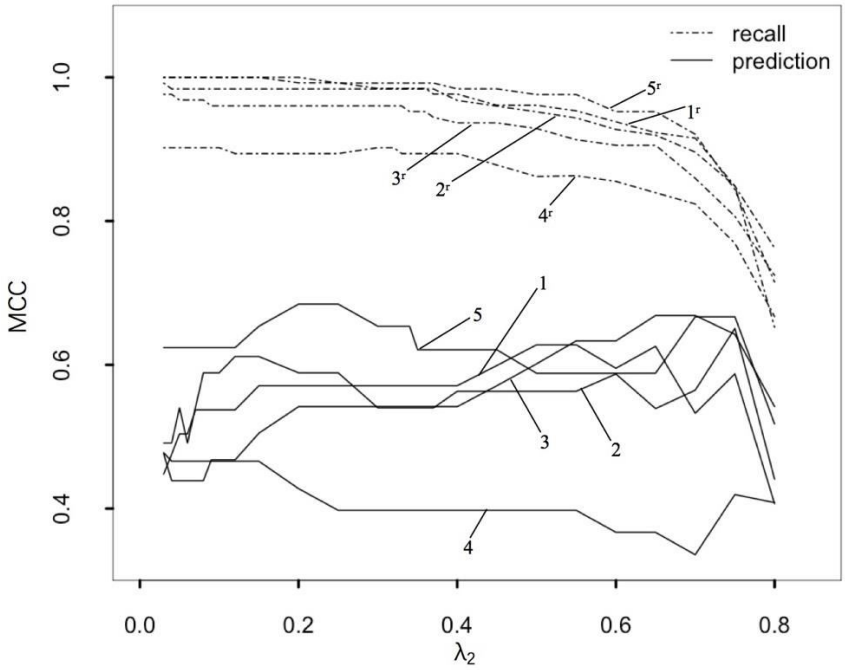

Figure 3.24: Prediction performances of the five subsets (Table 3.15) with L2 regularization after L1 feature selection. The number of features is given in Table 3.17. Dashed lines represent training set (recall) and solid lines represent prediction performances of test subsets. 1^r to 5^r indicate the recall curves of 1^{st} subset to 5^{th} subset and 1 to 5 indicate the prediction curves of 1^{st} subset to 5^{th} subset. This figure was made by R v3.1.3.

3.2.4.1.2 The prediction results of DemFeature-1

DemFeature-1 was also used to predict the Goracci database. DemFeature-1 was developed from DemPred. Unlike DemPred, DemFeature-1 specifically and carefully considers each compound in the test set. Based on the similarity rule (Equation (2.36)), an individual training subset is built for each compound of the test set. The method has been described in Chapter 2.10.1. DemFeature-1 uses a parameter (***cutoff***) to control the number of compounds in each training subset. In the application for PLD, the ***cutoff*** value 0.0 was used to construct DemFeature-1 models. The prediction power of this method has been verified through the datasets of the Kaggle™ competition as mentioned in Chapter 3.1. For the datasets of the Kaggle™ competition, albeit DemFeature-1 is CPU time expensive, it efficiently improves prediction performance. Due to the excellent prediction performance of DemFeature-1, it was employed also to predict the PLD datasets. The objective function of DemFeature-1(Equation (2.37)) includes an L2 regularization term used to mitigate the risk of overfitting. The parameter λ_2 needs to be optimized to control the strength of L2 regularization. Since DemFeature-1 is very CPU time expensive, one third of the compounds are randomly selected from the training subset as validation set to optimize λ_2. The candidates for λ_2 in this study include $\lambda_2 \in$ {0.15, 0.18, 0.20, 0.23, 0.25, 0.28, 0.30, 0.33, 0.35, 0.36, 0.37, 0.38, 0.4, … …,

0.45, 0.48, 0.50, 0.53, 0.55,… … , 0.7, 0.72, 0.75, 0.78, 0.79, 0.8, 0.82, 0.85, 0.88, 0.9, 0.92, 0.95}.

Table 3.18: Prediction results of the DemFeature-1 model with the reduced feature set obtained by L1 feature selection.

	λ_2[a]	# features[b]	MCC	accuracy	sensitivity	specificity
subset 1	0.55	155	0.567	78.8%(52/66)	0.941(15/17)	0.755(37/49)
subset 2	0.55	143	0.578	78.8%(52/66)	0.821(23/28)	0.763(29/38)
subset 3	0.65	106	0.562	78.8%(52/66)	0.708(19/24)	0.786(33/42)
subset 4	0.35	56	0.509	75.8%(50/66)	0.700(21/30)	0.806(29/36)
subset 5	0.3	194	0.561	83.3%(52/66)	0.741(20/27)	0.821(32/39)
average			**0.552**[c]	**78.2%(258/330)**	**0.778(98/126)**	**0.784(160/204)**

a. λ_2 is determined by randomly selecting one third of the compounds from the five training subsets.
b. The number of features was obtained using DemPred with L1 feature selection.
c. The ***MCC*** for the whole dataset is calculated by summing up ***TP***, ***TN***, ***FP*** and ***FN*** of five subsets.

Table 3.19: Prediction results of DemFeature-1 with all features (3849 features).

	λ_2[a]	# features[b]	MCC	accuracy	sensitivity	specificity
subset 1	0.250	3780	0.683	81.8%(56/66)	0.882(16/17)	0.816(40/49)
subset 2	0.350	3790	0.661	83.3%(54/66)	0.821(23/28)	0.816(31/38)
subset 3	0.150	3781	0.598	80.3%(54/66)	0.625(15/24)	0.929(39/42)
subset 4	0.650	3790	0.572	78.8%(52/66)	0.700(21/30)	0.861(31/36)
subset 5	0.550	3785	0.748	87.9%(58/66)	0.778(22/27)	0.949(36/39)
average			**0.645**[c]	**83.3%(275/330)**	**0.770(97/126)**	**0.873(178/204)**

a. λ_2 is optimized using one third of randomly selected compounds from the training subset.
b. The initial number of all features is 3849. For each subset, features, which did not vary, were deleted leading to feature numbers that can differ for each subset.
c. The ***MCC*** for the whole dataset is calculated by summing up ***TP***, ***TN***, ***FP*** and ***FN*** of five subsets.

DemFeature-1 was applied to the five subsets of the Goracci database[54] used by DemPred (see Table 3.16). Table 3.18 shows the prediction results of DemFeature-1 with L1 feature selection (Chapter 3.2.4.1.1). Compared with the prediction results of DemPred (shown in Table 3.17), the ***sensitivity*** improves from 0.706 to 0.762, especially on subset 4 whose ***sensitivity*** improved from 0.466 to 0.700 leading to a prediction performance of 0.509 by ***MCC*** and 75.8% by ***accuracy***. This substantial increase in correct identification of PLD inducers, which is important in the field of drug discovery, demonstrates the advantage of

DemFeature-1. On the other hand, the ***specificity*** of the DemFeature-1 model is only 0.784 compared to 0.828 obtained with DemPred. The overall ***accuracy*** of DemFeature-1 is not as good as of DemPred, However, the large enhancement in ***sensitivity*** yields a slightly larger ***MCC*** for DemFeature-1 for DemPred (0.536 versus 0.552).

In addition to using a reduced feature set to construct the DemFeature-1 model, the training set was also employed with all features (3849 features) to construct the DemFeature-1 model (results in Table 3.19). The DemFeature-1 model built with all features is significantly better than the DemFeature-1 model built with the reduced number features (see Figure 3.25). The ***MCC*** increases from 0.552 to 0.645 and the ***accuracy*** rises from 78.2% to 83.3%. Although the prediction performance for subset 4 is still the worst of all five subsets, the prediction result has improved considerably in terms of ***MCC*** and ***accuracy*** compared with the previous models (see Table 3.17 & Table 3.18). Comparatively, the ***specificity*** shows greater improvement than ***sensitivity***. The ***specificity*** of the DemFeature-1 model built with all features increases from 0.784 to 0.873, while the ***sensitivity*** stays practically on the same level.

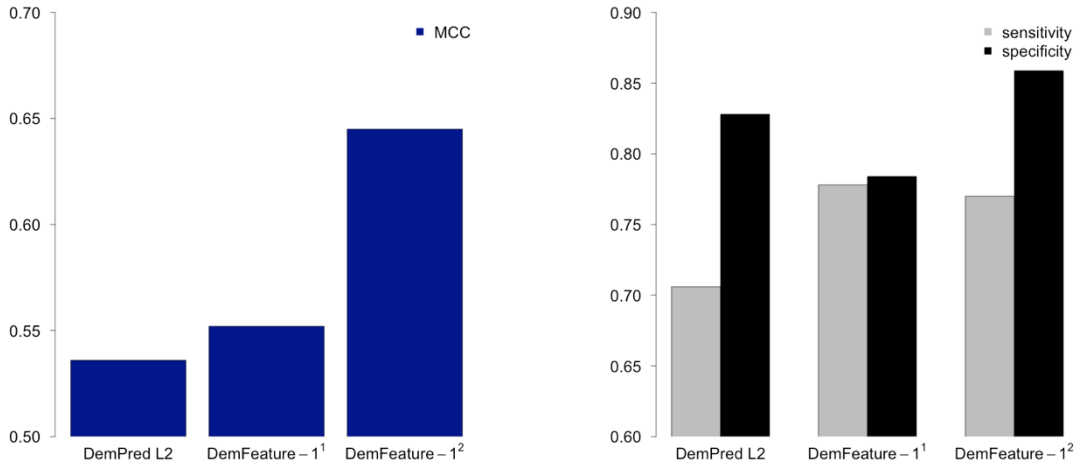

Figure 3.25: DemFeature-1^1 denotes the DemFeature-1 model trained with the reduced number of features. DemFeature-1^2 denotes the DemFeature-1 model trained with all features. The DemFeature-1 models are significantly better than the DemPred model, with a clear increase in ***sensitivity***. Compared to the DemFeature-1 model trained with reduced features, adding more features efficiently increases the ***specificity*** of the DemFeature-1 model. This figure was made by R v3.1.3.

Additionally, since the SMARTS features (Chapter 3.2.1.6) created by Przybylak *et al.*[97,117] offer an excellent prediction performance through a workflow, in this study the features of the updated SMARTS model were also employed to construct the DemFeature-1 model. The updated SMARTS model has 44 features including 39 structural patterns and 5 other

judgment conditions. The prediction results of the DemFeature-1 model with SMARTS features are shown in Table 3.20. The prediction performance of the DemFeature-1 model built with SMARTS features is much worse than the SMARTS model. Nevertheless, it is worth noting that the ***specificity*** of this model is 0.907, which is significantly better than the SMARTS model including original and updated versions whose ***specificities*** are 0.848 and 0.819, respectively.

Table 3.20: Prediction results of DemFeature-1 using the 39 SMARTS features[97, 118] (***cutoff*** = 0.0).

	λ_2[a]	# features[b]	MCC	accuracy	sensitivity	specificity
subset 1	0.300	39	0.621	86.4%(57/66)	0.529(9/17)	0.980(48/49)
subset 2	0.500	39	0.444	72.7%(48/66)	0.464(13/28)	0.921(35/38)
subset 3	0.300	36	0.381	72.7%(48/66)	0.458(11/24)	0.881(37/42)
subset 4	0.300	39	0.391	69.7%(46/66)	0.500(15/30)	0.861(31/36)
subset 5	0.300	39	0.390	71.2%(47/66)	0.481(13/27)	0.872(34/39)
average			**0.443**[c]	**74.6%(246/330)**	**0.484(61/126)**	**0.907(185/204)**

a. λ_2 is determined by a third validation sets randomly selected from training subsets.
b. The number of SMARTS features is 44. For each subset, the features (STD=0) were deleted leading to the number of features being different in each subset.
c. The **MCC** for the whole dataset is calculated by summing up ***TP***, ***TN***, ***FP*** and ***FN*** of 5 subsets.

To further investigate the influence of SMARTS features, they were added to the 3849 features used in our study to construct a DemFeature-1 model. Thus, a total of 3993 features were employed to build the DemFeature-1 model. The results of prediction performance of this model are shown in Table 3.21. The addition of SMARTS features is not useful to improve the prediction performance of the DemFeature-1 model. The prediction results of the models with additional SMARTS features and without SMARTS features almost stay on the same level. The ***sensitivity*** of this model is the same as the ***sensitivity*** given by the DemFeature-1 model only with the original 3849 features while the ***specificity*** of the model with additional SMARTS features decreases slightly from 0.907 to 0.858.

Table 3.21: Prediction results of DemFeature-1 with 39 added SMARTS features[97,118] (***cutoff*** = 0.0).

	λ_2[a]	# features [b]	MCC	accuracy	sensitivity	specificity
subset 1	0.3	3824	0.683	84.8%(56/66)	0.941(16/17)	0.816(40/49)
subset 2	0.35	3824	0.661	83.3%(55/66)	0.821(23/28)	0.842(32/38)
subset3	0.1	3822	0.563	80.3%(53/66)	0.625(15/24)	0.905(38/42)
subset4	0.65	3828	0.572	78.8%(52/66)	0.700(21/30)	0.861(31/36)
subset5	0.65	3829	0.687	84.8%(56/66)	0.815(22/27)	0.872(34/39)
average			**0.628**	**82.4%(272/330)**	**0.770(97/126)**	**0.858(175/204)**

a. λ_2 is determined by a randomly selected validation set which is one third of the training subset.
b. The number of features is 3993 including ours 3849 features and 44 SMARTS features. Features whose values do not vary in the training set were deleted leading to different numbers of features for each subset.
c. The ***MCC*** for the whole dataset is calculated by summing up ***TP***, ***TN***, ***FP*** and ***FN*** of 5 subsets.

3.2.4.1.3 The prediction results of TEM-confirmed compounds

As introduced above, the Transmission Electron Microscopy (TEM) is the gold standard method to identify positive or negative PLD properties of a compound. PLD positive compounds that are only confirmed by histopathological presence of foamy macrophages, cytoplasmic vacuolations, cytoplasmic granules, lipidosis, dyslipidosis or histiocytosis cannot be considered reliable PLD-inducers. On the other hand, many compounds in the PLD databases are considered PLD negative only due to the lack of evidence of PLD. Goracci *et al.*[54] analyzed seven popular PLD databases and removed unreliable compounds. However, the compounds in this curated database are not all confirmed by the TEM method.

Table 3.22: Comparison of predictions for TEM-confirmed and TEM-confirmed unclear compounds using the same prediction scheme as used for Table 3.19.

TEM-confirmed 178 compounds				
	true positive TP	**true negative TN**	**false positive FP**	**false negative FN**
number of compounds	78	66	9	25
mean value of scoring function	0.593	-0.778	0.306	-0.483
TEM-confirmed unclear 152 compounds				
number of compounds	19	112	17	4
mean value of scoring function	0.527	-0.628	0.451	-0.596

In the Goracci database, specific information concerning which compounds have been confirmed by the TEM method is not provided. Therefore, we tracked back to investigate the seven PLD databases used to construct the Goracci database. There are only four databases providing information of TEM identification. Of the 330 compounds of the Goracci database, 178 are TEM-confirmed while it is unclear for the other 152 compounds (TEM-confirmed unclear). Chapter 3.2.2.1 describes in detail the sources of TEM identification information.

The scoring function (Equation (2.10)) is used to predict the property assignment of a compound in the test set. The biological response representing PLD assignment in the training set is -1 and +1 for being negative and positive with respect to PLD-induction, respectively. If the value of the scoring function (***y*** value) is greater than zero, the test compound is judged to be positive else it is judged negative. The higher the positive predicted value of the scoring function is, the higher is the probability that the corresponding compound belongs to the positive set and vice versa. To observe whether TEM-confirmed compounds are more reliable, the predicted values of ***TP***, ***TN***, ***FP*** and ***FN*** of the TEM-confirmed compound set were summed up, respectively for comparison with their counterparts among the TEM-confirmed unclear compounds. The results of the comparison are shown in Table 3.22. The predicted values of the scoring function are calculated by the DemFeature-1 model built with all features (3849 features). The results for each compound are given in Appendix 3.

Table 3.22 demonstrates that the mean value of the scoring function (MVSF) of *TP*s of TEM-confirmed compounds is larger than the MVSF of *TP*s of TEM-confirmed unclear compounds. The MVSF of *TN*s for TEM-confirmed compounds is obviously lower than the MVSF of *TN*s for TEM-confirmed unclear compounds. These comparisons suggest that the reliability of TEM-confirmed compounds is higher than of the TEM-confirmed unclear compounds. On the other hand, the MVSF of *FP*s for TEM-confirmed compounds is lower than the MVSF of *FP*s for TEM-confirmed unclear compounds. The MVSF for *FN*s for TEM-confirmed compounds is larger than the MVSF of *FN*s for TEM-confirmed unclear compounds. Those results reflect that the TEM-confirmed unclear compounds have a higher probability to be judged incorrectly. However, it must be noted that the number of *FP*s for TEM-confirmed compounds and the number of *FN*s for TEM-confirmed unclear compounds are relatively small compared to the number of compounds in the other sets. In summary for the TEM-confirmed compounds correct assignments are made with a higher degree of confidence while incorrect assignments with lower degree of confidence than for the TEM-confirmed unclear compounds. However, one can also observe that the prediction results are more favorable for

the TEM-confirmed unclear dataset than for the TEM-confirmed compounds, since in the former case only 21 (13.8%) wrong assignments are made while in the latter case these are 34 (19.1%). To check whether such a comparison is robust, we also designed another way to compare the reliability (see Figure 3.26).

As shown in Figure 3.26, we used TEM-confirmed compounds as the training set to build a DemFeature-1 model to predict the TEM-confirmed unclear compounds. Then, we deleted the compounds classified incorrectly in TEM-confirmed unclear dataset and used the remaining compounds as a training set to in turn predict the TEM-confirmed compounds. In addition, instead of deleting the incorrect predicted compounds o the TEM-confirmed unclear dataset, we exchanged their PLD properties and then trained a DemFeature-1 model based on this modified TEM-confirmed unclear dataset to predict the TEM-confirmed compounds. The results are shown on Table 3.23.

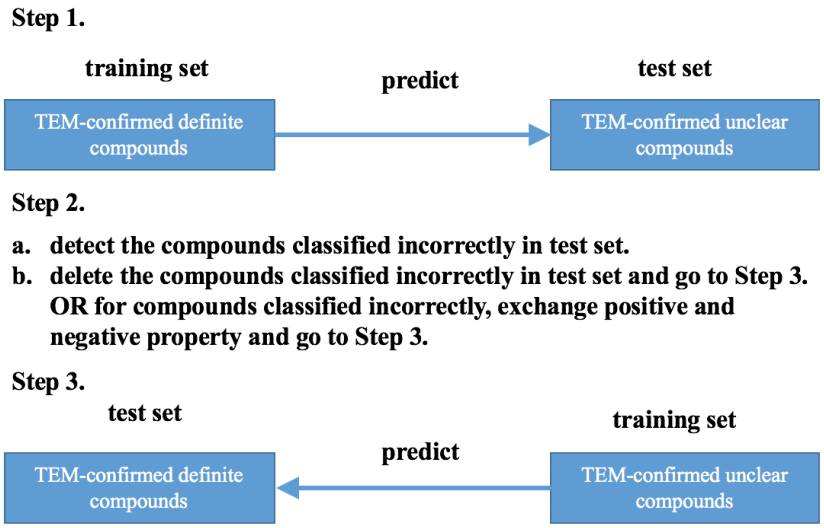

Figure 3.26: First the TEM-confirmed compounds are used to train a model to predict TEM-confirmed unclear compounds to find out which compounds are classified incorrectly in the set of TEM-confirmed unclear compounds. Then, two tests are performed to analyze whether the PLD properties of the incorrectly classified compounds are unreliable.

As shown in Table 3.23, after deleting the compounds classified incorrectly in the TEM-confirmed unclear set, the prediction performance can be improved (prediction 2). Furthermore, if we change the PLD properties of misclassified compounds, the prediction performance can be greatly improved (prediction 3). This improvement mainly comes from an enhanced ***sensitivity***. From those results, it can be deduced that the positive compounds in the TEM-confirmed unclear set include actually many negative compounds. On the other hand, the ***specificities*** of prediction 2 and prediction 3 are lower than for prediction 1. These

phenomena can explain why the quality of the negative compounds in the TEM-confirmed unclear set is relatively better than the quality of the positive compounds in the TEM-confirmed unclear set. Indeed, the majority of the negative compounds in the TEM-confirmed unclear set belong to the *high-confidence* compounds in the Orogo database[102]. The reason for those compounds being classified as negative compounds is because the absence of PLD keywords in the New Drug Application documents of the U.S. FDA. Combined with the investigation results in Table 3.23, those negative compounds in the TEM-confirmed unclear set can be trusted as relatively reliable PLD non-inducers. Therefore, when building a proper PLD prediction model, if there are too few negative compounds, those negative compounds in the TEM-confirmed unclear set can be considered as reliable PLD negative compounds.

Table 3.23: Prediction using different trainings sets of the TEM-confirmed unclear compounds.

	λ_2	MCC	accuracy	sensitivity	specificity
prediction 1	0.750	0.442	65.7%(117/178)	0.437(45/103)	0.960(72/75)
prediction 2	0.750	0.507	72.5%(129/178)	0.592(61/103)	0.907(68/75)
prediction 3	0.700	0.690	84.8%(151/178)	0.864(89/103)	0.827(62/75)

prediction 1: Use TEM-confirmed unclear set to predict TEM-confirmed set of the Goracci database[54].
prediction 2: Predict TEM-confirmed set after deleting the compounds classified incorrectly in TEM-confirmed unclear dataset.
prediction 3: Predict TEM-confirmed set after exchanging the PLD properties in the TEM-confirmed unclear dataset.

3.2.4.1.4 Measurement of predictive confidence

The predictive confidence measurement was defined in Chapter 2.13. It estimates the reliability of the classification for a compound. The predictive confidence is a probability measure that a compound is correctly predicted to be PLD inducer or not. According to this definition, for the binary classification, the minimal probability for the predicted class must be larger than 0.5. Confidence is normalized to the range of 0.0-1.0. The confidence can be considered as an internal measure of prediction performance. Thus, this confidence should correlate with prediction quality including ***MCC*** and ***accuracy***. The Figure 3.23 demonstrates the correlation between confidence and prediction quality. The prediction result was provided by DemFeature-1 model built with 3849 features (see Table 3.19).

Based on the prediction quality of the compounds in each bin of the histogram, clearly, the prediction quality is proportional to the confidence level. The compounds with high

confidence have a high probability of being classified correctly. According to Figure 3.27, from the confidence level 0.5 onwards, the prediction quality is such that ***accuracy*** and ***MCC*** values are above 0.8. Furthermore, the confidence of the majority of compounds is greater than 0.5. Moreover, it is worth noting that of those 10 bins, the bin representing the maximum confidence interval (0.9-1.0) accounts for the largest number of compounds (>24%). Overall, it can be concluded that a compound predicted with higher the confidence is more likely correctly predicted.

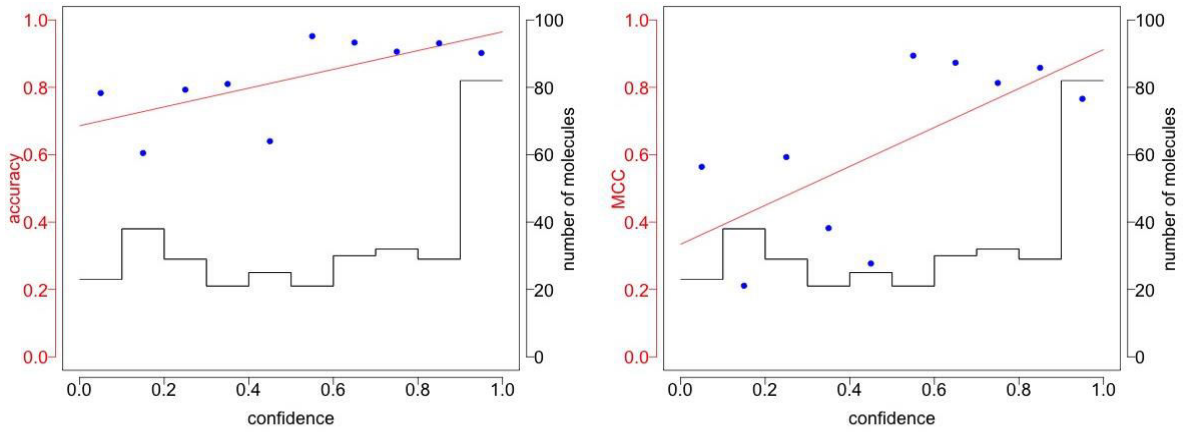

Figure 3.27: The left plot represents the correlation between the confidence and ***accuracy*** and the right plot represents the correlation between the confidence and ***MCC*** value. Each plot consists of a histogram for the number of molecules (**black**), prediction quality points (**blue**) and a linear fitted line (**red**). The histogram contains 10 bins. Of those 10 bins, each bin represents a confidence interval of 0.1. 330 compounds from the Goracci database spread over those bins in accordance with the confidence value of each compound. The number of compounds in each bin is reflected on the right vertical axis. The prediction quality points (***accuracy*** and ***MCC***) were calculated based on the compounds within each bin. The result shown on this figure is given by the DemFeature-1 model built with 3849 features. The fitting equations are: $accuracy = 0.2787 conf^{1} + 0.6861$, $MCC = 0.5780 conf^{1} + 0.3341$; This figure was made by R v3.1.3.

3.2.4.1.5 PLD prediction performance for different models

Several *in silico* prediction models classifying PLD inducers and PLD non-inducers have been introduced in the previous sections. In this section, we employ the Ploemen model (Chapter 3.2.1.1), Pelletier model (Chapter 3.2.1.2), original SMARTS model and updated SMARTS model (Chapter 3.2.1.6), to compare with the models we developed, DemPred (Chapter 2.9) and DemFeature-1(Chapter 2.10.1). The results of the Ploemen model and Pelletier model were produced by our programs. The results of the original SMARTS model and updated SMARTS model were given in the publication by Przybylak *et al.*[97] . Table 3.24 lists the results of the comparison.

Table 3.24: Predictive statistics of PDL prediction models.

	MCC	accuracy	sensitivity	specificity
Ploemen model	0.421	66.1%(218/330)	0.452(57/126)	0.789(161/204)
Pelletier model	0.534	77.9%(257/330)	0.722(91/126)	0.814(166/204)
original SMARTS	0.56	79.4%(262/330)	0.706(89/126)	0.848(173/204)
updated SMARTS	0.608	80.6%(266/330)	**0.833(105/126)**	0.789(161/204)
DemPred[a]	0.536	78.2%(258/330)	0.706(89/126)	0.828(169/204)
DemFeature-1[b]	**0.645**	**83.3%(275/330)**	0.770(97/126)	**0.873(178/204)**

a. The result of the DemPred model is given by L2 regularization. It is based on reduced number of features, which were selected by L1 approach.
b. The result of the DemFeture-1 model is obtained with all 3849 features.

According to the comparison (Table 3.24), the Ploemen model shows a poor prediction performance with 66.1% by ***accuracy*** and 0.421 by ***MCC***. This model is based only on two simple physicochemical properties, log *P* and *pKa*. Based on the prediction performance of the Ploemen model, it seems that log *P* and pK_a alone are not sufficient to give a reasonable prediction result. However, the Pelletier model, which is a modified version of the Ploemen model and is also based on log *P* and pK_a, provides an obvious improvement in prediction performance in terms of ***accuracy***, from 66.1% to 77.9% and in ***MCC***, from 0.421 to 0.534. Despite this improvement the Pelletier does not reach the prediction quality of the SMARTS models and of our models.

The prediction results of both original SMARTS model and updated SMARTS model are better than those of the Ploemen, Pelletier and DemPred model. The updated SMARTS model improves the ***sensitivity*** compared to the original SMARTS model but the updated SMARTS model predicts more false positives than the original SMARTS model. Although the updated SMARTS model has the best ***sensitivity*** of all considered prediction models, the decrease in ***specificity*** lowers the prediction results characterized by ***MCC*** and ***accuracy***.

The DemFeature-1 model built with all 3849 features offers the highest predictive power. The 3849 features cover a large area of chemical diversity, which seems to be advantageous to identify PLD inducers correctly. The usage of a large number of features is not helpful to explain mechanism of PLD occurrence, which is still not understood. On the other hand, such *in silico* models can efficiently guarantee drug safety in the early stage of drug development.

Table 3.25: The comparison of *P*-value among different models.

Goracci database	total number of compounds: 330				
	updated SMARTS	**original SMARTS**	**DemPred[a]**	**Pelletier**	**Ploemen**
DemFeature-1[b]	0.374	0.171	0.017	0.045	0.006
updated SMARTS		0.584	0.445	0.306	0.018
original SMARTS			0.740	0.614	0.026
DemPred[a]				1.000	0.138
Pelletier					0.071

a. The result of the DemPred model is given by L2 regularization. It is based on the reduced number of features, which were selected by the L1 approach.
b. The result of the DemFeture-1 model is computed with all 3849 features.

3.2.4.1.6 Comparing prediction models by *P*-values

To further compare the difference in prediction results between two models, *P*-values were computed. As introduced in Chapter 2.12, a binomial test (Equation (2.41)) or a McNemar's test (Equation (2.42)) can be used to calculate the *P*-value. A matrix of the *P*-values is shown in Table 3.25 comparing the prediction results of 6 different models. Based on those *P*-values, only few of the considered pairs of prediction models are significantly different from each other possessing *P*-values below 0.05. These are DemFeature-1 relative to the DemPred, Pelletier and Ploemen model and on the other hand the Ploemen model relative to the two SMARTS models. All other models have a prediction quality, which in a statistical sense is similar. Since the Pelletier model uses only a simple rule referring to two features, log*P* and pK_a, the corresponding *P*-values reflect that these two features play important roles in identifying PLD inducers. The *P*-value between the updated SMARTS model and original SMARTS model is 0.584. It reveals that the addition of seven SMARTS patterns in the updated SMARTS model does not significantly change the prediction performance of the original SMARTS model. Moreover, except for the updated SMARTS model, the *P*-values between the DemFeature-1 model built with all 3849 features and all other models are relatively small. Since the DemFeature-1 model is built with all features gives the best prediction results, those *P*-values support evidence that the mechanism of PLD formation cannot be explained a few features. To efficiently identify PLD inducers in the early stage of drug development, *in silico* PLD prediction models need to consider many features.

3.2.4.2 Prediction results of independent test set

As discussed in Chapter 3.2.2.2, in this research an independent test set was prepared to further assess *in silico* models predicting PLD induction potential. This independent test set

consists of 131 compounds with 72 PLD positive and 61 PLD negative compounds. According to the dataset provider, Boehringer Ingelheim, the PLD activity of those compounds was assessed by a fluorescent phospholipid-based assay, which is an *in vitro* cell-based method[108]. Although the gold standard method TEM was not used to evaluate those compounds, the prediction results of this independent test set can still be a reference for evaluating the prediction performances of models.

To create a scenario as for previous efforts predicting the Goracci database, the 3849 features used to predict the Goracci database were also utilized to construct the DemPred model and DemFeature-1 model for predicting the independent test set. The 330 compounds in the Goracci database were used as training set to build prediction models. In addition to DemPred and DemFeature-1, the Pelletier model and updated SMARTS model were also employed to predict the independent test set and the results were compared.

For DemPred and DemFeature-1 models, identical λ candidates used to predict the Goracci database were prepared for optimizing the λ-values. The L1 & L2 two-step method was also applied to construct the DemPred model, where L1 is the feature selection step. The optimized λ_1 enables DemPred to use a reduced number of 55 features. According to the absolute weight values of the features, the 15 most important features are shown in Figure 3.27.

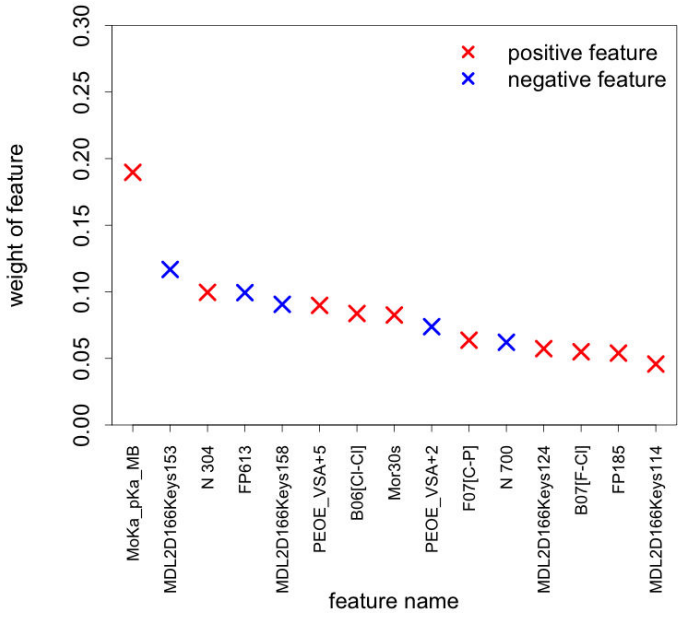

Figure 3.27: The 15 most important features are shown with their corresponding weights. The larger the absolute value of the weight is, the more important is the corresponding feature. The weights are calculated by the DemPred model with the L1 approach and λ_1 =0.075. This figure was made by R v3.1.3.

As demonstrated in Figure 3.27, the feature *Moka_pKa_MB* has the strongest correlation with the PLD property. *Moka_pKa_MB* is the most basic pK_a value, which is consistent with the prediction rule of the Ploemen and Pelletier model. It shows that the most basic pK_a value indeed is an important feature for predicting the PLD property of a compound. On the other hand, the feature *MDL2D166Keys153* is the strongest descriptor for detecting negative compounds. This feature defines a 2D substructure in a compound containing a carbon with at least three neighbors when those neighbors are two carbons as well as one oxygen. Around 64.2% negative compounds contain this substructure.

Table 3.26 shows the results of different approaches predicting the independent test set. For DemPred and DemFeature-1, the 330 compounds of the Goracci database were used as the training set. The independent test set was not used to construct the prediction models.

Table 3.26: Prediction results of the independent test set by different models.

	MCC	accuracy	sensitivity	specificity
Pelletier model	0.511	75.2%(100/133)	0.903(65/72)	0.557 (35/61)
updated SMARTS	0.442	72.2%(96/133)	0.722(52/72)	0.721 (44/61)
DemPred L1 feature selection [a]	0.410	69.9%(93/133)	0.639(46/72)	0.770 (47/61)
DemPred L2 regularization [b]	0.315	63.9%(85/133)	0.500(36/72)	0.803 (49/61)
DemFeature-1 (all features)[c]	0.255	59.4%(79/133)	0.375(27/72)	0.852 (52/61)
DemFeature-1 (800 best features)[d]	0.531	76.7%(102/133)	0.778(56/72)	0.754 (46/61)

a. DemPred with L1 feature selection, λ_1 = 0.075. 55 features were retained from L1 approach.
b. DemPred with L2 regularization was applied to the dataset with 55 features after L1 feature selection, λ_2 = 0.13.
c. DemFeature-1 built with all features (3489 features), λ_2 = 0.700.
d. The 800 best features were selected through absolute weight values of features, which were calculated by DemPred with L2 (λ_2 = 0.100). The larger the absolute weight values are, the more important the corresponding features, λ_2 = 0.410.

Overall, comparing ***MCC*** and ***accuracy***, the prediction performance for the independent test set is worse than for the Goracci database. Surprisingly, the Pelletier model produces the best result, especially in terms of ***sensitivity***. As shown in Table 3.26, the prediction results of the Pelletier model on the independent test set is even better than the updated SMARTS model. The ***sensitivity*** of the Pelletier model can reach 0.903, which is overwhelmingly better than other models. The updated SMARTS model on the independent test set does not perform as good as on the Goracci database and its prediction results are worse than Pelletier model. Only the ***specificity*** of the updated SMARTS model is better than the Pelletier model. This

result seemingly contradicts the conclusion that Ploemen and Pelletier models are insufficient to classify PLD inducers from non-inducers in the publication of Przybylak *et al* [97].

The L1 & L2 two-step method was also applied to build the DemPred models. The optimized λ_1 enables L1 regularization to give a reduced features set consisting of 55 features, which yields ***MCC***: 0.410 and ***accuracy***: 69.9%. The prediction of DemPred L2 regularization was performed on the dataset with 55 features carried over from L1 feature selection but it did not give better prediction results.

For the independent test set, DemFeature-1 with all features performs very poorly with ***MCC*** of only 0.255 and ***accuracy*** of 59.4%. However, the DemFeature-1 model built with the 800 best features greatly improves the ***MCC*** from 0.255 to 0.531 and the ***accuracy*** from 59.4% to 76.7%, which is better than other models. The 800 best features were selected by the absolute weight values, which represent the importance of corresponding features. The weight values were calculated by DemPred with the L2 approach with λ_2 = 0.10. To investigate how the prediction results of the test set depend on the number of best features, we evaluate the MCC as a function of the number of best features. There are for instance 349 features whose absolute values of the weights is above 0.01 and 2900 features with weight values above 0.001. Figure 3.28 illustrates the results.

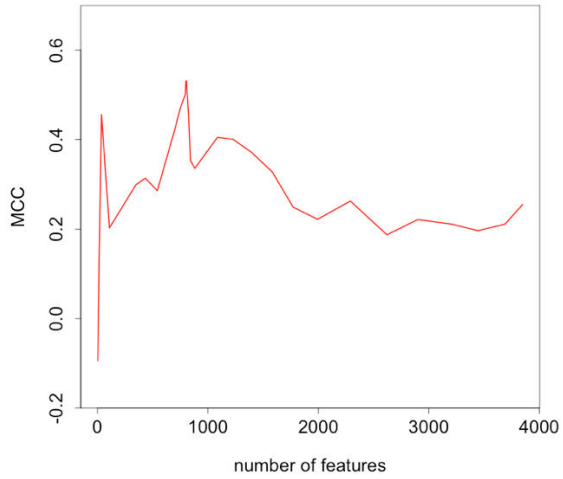

Figure 3.28: The ***MCC*** of the prediction results of DemFeature-1 is displayed for the test set as a function of the number of best features. The best features are determined by applying DemPred with L2 regularization using λ_2 = 0.10. The importance of features is ranked according to their absolute values of weight that they obtain with the DemPred computation. One starts with the list of features at the highest rank and includes all features of subsequent lower rank until the given number of best features is reached. This figure was made by R v3.1.3.

As shown in Figure 3.25, choosing around 800 of the most important features gives the best prediction result. Interestingly, with around 40 most important features, it gives the second

best prediction result. With less than 30 of the most important features, the performance of the prediction goes down drastically. A further investigation into the 40 most important features found that *Moka_pKa_MB* ranks 31st. As we introduced before, *Moka_pKa_MB* is an important feature for predicting PLD. When this feature is deleted, the prediction performance decreases markedly. It demonstrates that also for the independent test set, the most basic pK_a plays an important role in distinguishing PLD inducers from PLD non-inducers.

For DemFeature-1, the independent test set is different from the situation of the Goracci database whose best result was obtained using all 3849 features. Here the optimal prediction result was achieved with the 800 best features. The complexity of the DemFeature-1 model built with the 800 best features is smaller than the one built with all 3849 features. This indicates that the independent test set is dissimilar from the Goracci database. Moreover, for further demonstrating the dissimilarity between these two sets, the following investigation was carried out. Equation (3.8) was employed to calculate the inter-variance between the Goracci database and the independent test set and the intra-variances of both datasets. The results are shown in Table 3.27.

$$\mathrm{var}(1,2) = \sqrt{\frac{1}{N_1 N_2} \sum_{i=1,\, j=1}^{N_1 N_2} (\hat{\vec{f}}_{1,\mathrm{i}} - \hat{\vec{f}}_{2,\,j})^2} \tag{3.8}$$

Suppose that there are two datasets, namely, 1st dataset and 2nd dataset. In the Equation (3.8) $\hat{\vec{f}}_{1,i}$ represents the $\boldsymbol{i^{th}}$ normalized feature vector in 1st dataset and $\hat{\vec{f}}_{2,\,j}$ represents the $\boldsymbol{j^{th}}$ normalized feature vector in 2nd dataset. Both feature vectors are normalized by Equation (2.35). $\boldsymbol{N_1}$ and $\boldsymbol{N_2}$ are the number of compounds in 1st and 2nd datasets, respectively. var(1,2) represents the inter-variance between the two datasets. When the 2nd dataset is replaced by the 1nd dataset in Equation (3.8) it measures the intra-variance of the 1nd dataset.

As shown in Table 3.27, the intra-variance of the total set, the intra-variance of the positive set and the intra-variance of the negative set are for the independent test set smaller than for the Goracci database. The values of the intra-variances reflect that in the chemical space the distribution of compounds of the independent test set is narrower than of the Goracci database. Additionally, the inter-variances between Goracci database and independent test set considering the total sets, the positive sets and the negative sets are all larger than the corresponding intra-variances. Hence, the chemical space covered by the independent test set is not fully included in the chemical space covered by Goracci database. Therefore, using Goracci database for training can cause some problems to predict the independent test set as

has been experienced. In the independent test set 54 compounds are incorrectly classified by DemFeature-1 using all 3849 features (see Table 3.26). According to Table 3.27, the inter-variances of the 54 compounds of the independent test set that are incorrectly classified are slightly larger than the corresponding inter-variances considering all compounds of the independent test set. This indicates that the incorrectly classified compounds of the independent test set are indeed more dissimilar from Goracci database than the correctly classified compounds..

Table 3.27: Intra-variances of Goracci database (GD) and independent test set (ITS) and inter-variance of GD and ITS. All variances are measured with 3849 features.

	inter-variance of GD and ITS	intra-variance of GD	intra-variance of ITS	inter-variance of GD and incorrectly classified compounds in ITS[d]
total set [a]	1.418441	1.411507	1.264117	1.418653
positive set [b]	1.404277	1.389553	1.187447	1.408309
negative set [c]	1.422522	1.404093	1.310719	1.426637

a. variances involving positive and negative set.
b. variances of the positive set only.
c. variances of the negative set only.
d. The 54 incorrectly classified compounds in independent test set are obtained by DemFeature-1 with 3849 features (see Table 3.26).

In addition, it is worth noting that the Goracci database is a more reliable dataset, with 54% of compounds having been identified by the TEM method and many of the compounds have been evaluated by several labs independently. In contrast the independent test set was only evaluated by an *in vitro* method (Chapter 3.2.3.3) and not with the gold standard method, TEM.

3.2.5 Discussion

DemPred built with L1 & L2 two steps method

Since the quality of the PLD dataset has an important influence on the prediction performance of the models, the carefully curated database made by Goracci *et al.*[54] was employed in this study to construct *in silico* models for predicting PLD. To enable each compound of the Goracci database to be predicted, a 5-cross validation was used to predict the Goracci database (see Table 3.15). Moreover, since the mechanism causing PLD is still not clear, a large number of features were generated to cover as large a chemical space as possible.

The L1&L2 two-step method was used to build the DemPred models. L1 is the step for feature selection. With optimized λ_1 value less than 0.05% of features are kept yielding prediction results of 0.532 for ***MCC*** and 77.9% for ***accuracy***, which is equivalent to the prediction performance of the Pelletier model on the Goracci database (see Table 3.16 & Table 3.24). Seemingly, a small number of features have a strong capacity to classify PLD inducers from PLD non-inducers. This phenomenon proves that a large fraction of the 3849 features have no predictive power or may even disturb results for predicting PLD.

Furthermore, based on the reduced number of features given by the L1 approach, the L2 regularization was used to build the DemPred models. L2 regularization slightly improved the prediction performance reaching 0.536 for ***MCC*** and 78.2% for ***accuracy***, which is not an significant improvement. This result may be explained by L1 feature selection having made a rigorous feature selection so that the L2 regularization cannot significantly increase prediction performance.

DemFeature-1

The reduced number of features selected by DemPred with the L1 approach were also used to build the DemFeature-1 model. The prediction results of the DemFeature-1 model based on reduced number of features are 0.552 for ***MCC*** (see Table 3.18), which is better than the results (***MCC*** = 0.536) of the DemPred model. Comparing with DemPred, the improved prediction result with DemFeature-1 is mainly due to the ***sensitivity*** enhancing from 0.706 to 0.778, while the ***specificity*** decreases from 0.828 to 0.784 (see Table 3.17 & Table 3.18). Despite this behavior, the improvement in the correct identification of PLD inducers is very important for drug development. It is worth noting that the ***sensitivity*** of subset 4 obtained with DemPred is only 0.466, which is the worst one among all five subsets. With DemFeature-1 ***sensitivity*** increases from 0.466 to 0.700. This result clearly manifests the advantage to use DemFeature-1 for drug development.

In addition to the DemFeature-1 model built with the reduced number of features, we also employed all features (3849 features) to build the DemFeature-1 model. The prediction results of the DemFeature-1 model built with all features improve the prediction performance measured by ***MCC*** and ***accuracy*** (see Table 3.19) considerably compared to the DemFeature-1 model built with the reduced number of features (see Table 3.18). The prediction performance was improved, since the ***specificity*** increases from 0.784 to 0.873 while the ***sensitivity*** almost stays on the same level. This phenomenon shows that a reduced features set

selected by the L1 approach is sufficiently capable of identifying PLD-inducers but still allows to many false positives. After adding more features to build the DemFeature-1 model, the number of false positives goes down (see Figure 3.21). Seemingly, the majority of features are needed to identify PLD non-inducers.

Analysis of SMARTS features used with DemFeature-1

The decision tree-like SMARTS model built by Przybylak *et al*. offers a good prediction performance with 44 SMARTS features (see Table 3.24). However, using the same SMARTS features, the DemFeature-1 model does not offer a reasonable prediction performance (see Table 3.20). It seems that those SMARTS features are unsuitable for a statistics-based model. It could be explained that the SAMRTS model is a rule-based model in which an unknown compound need to be judged through several steps starting from the top of the decision tree and going downward. Hence, in the SMARTS approach the features are not considered simultaneously, in contrast to the prediction algorithms used in the present study, which may make a difference. On the other hand, the prediction results of DemFeature-1 model built with the combination features set including our 3849 features and 44 SMARTS features is almost equivalent to the DemFeature-1 model only using all 3849 features (see Table 3.21). From this result one may conclude that the predictive capacities of the 3849 features have covered the predictive capacities contributed by the 44 SMARTS features.

Analysis of subset 4

Of all five subsets of the Goracci database, the subset 4 is the hardest one to predict (see Figure 3.19). The best result for subset 4 given by DemPred is only 0.398 for ***MCC*** and 69.7% for ***accuracy*** (see Table 3.17). Although the prediction result for subset 4 of DemFeature-1 improves to 0.572 for ***MCC*** and 78.8% for ***accuracy***, the prediction result for subset 4 is still slightly worse than for the other subsets. A method illustrated in Figure 3.22 (results in Table 3.23) was also used to investigate subset 4. The investigation results for subset 4 are shown in Table 3.28.

It can be seen that both methods (1) deleting compounds that are classified incorrectly in subset 4 and (2) inverting the PLD properties of those compounds can produce a better prediction result than using the original subset 4 to predict the remaining compounds in the Goracci database. The good results of both methods are mainly contributed to an improved ***specificity*** (see prediction 1 & 2 in Table 3.27), which elucidates that quite a number positive compounds in subset 4 may actually be negative compounds. When we put those compounds

into the negative set or delete them, the prediction performance can be improved by an increase in ***specificity***.

Table 3.28: Prediction of the Goracci dataset with DemFeature-1 using different training sets of subset 4.

	λ_2	MCC	accuracy	sensitivity	specificity
prediction 1	0.600	0.518	0.769(203/264)	0.750(72/96)	0.780(131/168)
prediction 2	0.400	0.599	0.818(216/264)	0.677(65/96)	0.899(151/168)
prediction 3	0.300	0.598	0.818(216/264)	0.656(63/96)	0.911(153/168)

prediction 1: Use subset 4 to train a DemFeature-1 model and predict remaining compounds of the Goracci database[54] belonging to the subsets 1, 2, 3, 5.
prediction 2: Predict remaining compounds of the Goracci database after deleting compounds in training subset 4, which were incorrectly classified before.
prediction 3: Predict remaining compounds of the Goracci database using a subset 4 for training where the PLD properties of incorrectly classified compounds in subset 4 were inverted.

Prediction 2 & 3 give a relatively worse ***sensitivity*** than prediction 1. We can deduce that the quality of the negative compounds in original subset 4 is more reliable than of the positive compounds. In addition, another piece of evidence to support this deduction is that the ***specificity*** of subset 4 predicted by the DemPred model built with the L1 approach is 0.861, while the ***sensitivity*** is only 0.500 (see Table 3.16). After L2 regularization, the ***specificity*** of subset 4 predicted by the DemPred model is 0.889, whereas, the ***sensitivity*** even decreases to 0.466. Hence, the main reason that DemPred models do not return good prediction results for subset 4 is the low quality of the positive compounds. Moreover, it is worth noting that the DemFeature-1 model (see Table 3.18 and Table 3.19) has significantly improved the ***sensitivity*** of subset 4 to 0.700, which demonstrates the advantage of DemFeature-1 prediction model.

Analysis of the independent test set

The Pelletier model gives an excellent prediction result for the independent test set. The prediction performance of the Pelletier model on this dataset is obviously better than other datasets. To explain the reason, we need to consider the principle of the Pelletier model. The majority of PLD inducers are CADs, which have the property of lysosomotropism. Those compounds are membrane penetrable and partition across the lysosomal membrane based on a concentration gradient. In addition, the property of weak base of CADs makes them

protonated in the acidic environment. As a result, the pH in lysosome would increase toward neutrality, which is less favorable for lysosome hydrolases. The prediction rules of the Pelletier model judge a compound with high log *P* and high pK_a as a PLD inducer (see Chapter 3.2.1.2), which is based on the lysosomotropic properties of CADs. In the independent test set, there are 91 compounds out of 133 compounds that fulfill the rules of the Pelletier model to be judged as PLD-inducers. Of those 91 compounds, 65 compounds are experimental PLD-inducers. This is the reason why the Pelletier model offers a high ***sensitivity***, resulting in the Pelletier model's good ***MCC*** and ***accuracy*** (see Table 3.26).

Despite the very good prediction performance of Pelletier model on the independent test set is very good, only two features, log *P* and pK_a, are still not sufficient to explain PLD. Some PLD inducers' log *P* are not high, for example, aminoglycoside antibiotics, but they also can go into cells[132]. Possibly, the hydrophilic channels in the cell membrane can bring the compounds with low log *P* into cells. Moreover, not all PLD inducers belong to CADs. Therefore, only considering two simple features is not reliable for detecting PLD in drug development. Nevertheless, the advantage of the Pelleter model is relatively faster than other *in silico* models. In drug discovery, Pelletier model can be used to screen a huge number of unknown compounds in the first step and then use other powerful yet time-consuming models such as DemFeature-1 to investigate the compounds survived from Pelletier model screening.

For the independent test set, the Pelletier model performs even better than the SMARTS model. This situation is inconsistent with the results in the publication of Przybylak *et al*[97]. In their publication, the SMARTS model is significantly better than the Pelletier model. In our research on the Goracci database, the prediction results of the SMARTS models are also better than Pelletier model (see Table 3.24). The limitation of the SMARTS model is one reason to explain this phenomenon. SMARTS features were concluded from available public PLD datasets. Those features were particularly chosen to give the "possible best" prediction results so there may be some loss of generality. Some of the SMARTS features are very specific to identifying a single compound. Thus, for an undisclosed dataset, namely, independent test set, the SMARTS model possibly fails to predict it with a good prediction result.

3.2.6 Conclusion

Drug-induced PLD is a side effect, which can impair lipid metabolism and the accumulation of phospholipid and drugs in cells. To date, the mechanism of drug-induced PLD is still not

clear yet. Regarding the safety of drugs, PLD needs to be detected as early as possible. *in silico* models demonstrate an important high-throughput method to detect toxicity in the early stage of drug development at a low cost. Moreover, it can be used to test new compounds before chemical synthesis. Recently, several *in silico* models have been proposed to identify the PLD induction potential of compounds such as the Pelletier model and SMARTS model. In this study, we utilized our *in silico* methods, DemPred and DemFeature-1 to investigate the prediction power of models on PLD inducers. The carefully curated Goracci database and a proprietary independent dataset provided by Boehringer Ingelheim Inc. were used as research datasets.

The Pelletier model, which is based on only two physicochemical features, can provide good prediction results on CADs compounds with high *pK*a values and log *P*. However, those simple properties are not enough to fully explain PLD-inducers. Recent research reports have proved that not all CADs compounds can induce PLD and not all PLD-inducer are CADs. The SMARTS model proposed by Przybylak *et al*. considered 44 structural features capable of inducing PLD. Although this method improved statistical predictivity on the Goracci database compared with the Pelletier model, it does not offer better prediction performance than the Pelletier model on the independent test set. Since the SMARTS features were particularly chosen from available PLD datasets, the SMARTS model seemingly does not offer a good generality to properly predict proprietary dataset. In addition, our methods, DemPred and DemFeature-1 were used to predict the Goracci database and independent dataset, respectively. DemFeature-1 gives the best prediction performance compared with DemPred and other *in silico* methods. Our methods considered as many molecular descriptors as possible to build models and obtained excellent prediction results. Based on a comparison of results, it seems that for properly predicting PLD induction potential, which has complex induction mechanism, researchers need to consider more factors, not only a few simple features.

The purpose of this research was not only to propose a good method to predict PLD, but also to demonstrate the prediction power of DemFeature-1. Comparing to other models, the predictive power of DemFeature-1 for predicting PLD in the Goracci database and the independent test set are comparable to the best results. DemFeature-1 is fully automated and no additional programming works is needed on the part of the users. Hence, DemFeature-1 could be used as a general prediction platform for solving many (Q)SAR tasks in drug development.

4 Summary

Drug development is a very complex project, which is not only high-cost and time-consuming but also has high failure rate. Therefore, the techniques of computer-aided drug design are very critical for pharmaceutical industry. (Q)SAR is a popular technique of computer-aided drug design. (Q)SRA models correlating biological activity with molecular structures can dramatically reducing development time; it can also drains on manpower and material resources, which would be impossible in a wet lab. On the other hand, (Q)SRA models also can be alternative methods to *in vivo* tests when consider the ethnic reasons.

However, the implementation of (Q)SAR prediction techniques needs professional programming skills and related expert knowledge of mathematics. Those requests would hamper the majority of medical researchers if they start to develop a program from scratch. Hence, there is a demand for a powerful yet easy to operate (Q)SAR building software program for which users can simply customize their (Q)SAR model to a research target. Previously, our group has developed a (Q)SAR prediction package, DemPred, which has been used to solve various classification and regression problems such as prediction of human volume of distribution and clearance and predicting major histocompatibility complex II epitopes. In my doctoral research, based on the DemPred, we developed an updated prediction algorithm, DemFeature. The core of DemFeature is also a linear discrimination scoring function. In contrast to the DemPred, the distinct feature of DemFeature is that the it independently construct a specific training subset for each compound in the test set. It means that each compound in the test set would be predicted by a specific scoring function. The rule of constructing a training subset is referred to the similarity between the compounds in the training set and the compound to be predicted. DemFeature has two versions: DemFeature-1 and DemFeature-2. DemFeatur-1 utilized a ***cutoff*** value to decide how similar compounds in training set can be selected to constitute the specific training subset for a compound to be predicted, while the DemFeature-2 gives a fixed number of training subset including most similar and most dissimilar compounds for a compound to be tested.

In my doctoral research, two datasets were utilized to test prediction ability of DemFeature. The first one was a contest on Kaggle™ platform launched by Boehringer Ingelheim whose dataset is related to gentoxcitiy, an important property in drug development. The other one is drug-induced phospholipidosis, which is a side effect of drugs. Recent years, it has been interested in pharmaceutical research community for drug safety. Compared with DemPred, the prediction performance of the DemFeature has been improved. The prediction results were

even better than some state of art prediction models on both cases. Therefore, DemFeature could be employed as a computer-aided tool used in the early stage of drug development.

4 Zusammenfassung

Wirkstoffentwurf ist ein anspruchsvolles Thema, welches nicht nur zeit- und kostenintensiv ist, sondern auch eine hohe Misserfolgsquote aufweist. Aus diesem Grund ist der computergestützte Wirkstoffentwurf sehr entscheidend in der Pharmaindustrie. (Q)SAR-Modelle, welche biologische Aktivität von Molekülen aufgrund ihrer Struktur beschreiben, sind in der Lage, den Bedarf von Arbeitskräften und Materialien signifikant zu reduzieren. Zusätzlich bieten (Q)SAR-Modelle eine alternative Herangehensweise, falls *in-vivo*-Studien aus moralischen Gründen nicht in Frage kommen.

Das Entwickeln von (Q)SAR-Vorhersagemethoden erfordert professionelle Programmierkenntnisse sowie tiefgehendes mathematisches Verständnis. Diese Anforderungen erschweren Wissenschaftlern in der medizinischen Forschung die Entwicklung eigener Software. Aus diesem Grund, besteht ein hoher Bedarf an leistungsstarker und einfach zu benutzender Software zur Erstellung von (Q)SAR-Modellen. Vorab wurde in unserer Arbeitsgruppe die Software-Bibliothek „DemPred“ entwickelt, welche benutzt wurde, um verschiedene Klassifikations- und Regressionsfragestellungen zu lösen. Beispiele sind die Vorhersage von Verteilungsvolumen im menschlichen Körper sowie Vorhersage von Epitopen des Haupthistokompatibilitätskomplexes. Während meiner Doktorarbeit habe ich, auf DemPred basierend, einen aktualisierten Algorithmus („DemFeature“) entwickelt. Der Kern von DemFeature und auch DemPred ist eine lineare Bewertungsfunktion, welche zur Klassifizierung benutzt wird. Der wesentliche Unterschied zu DemPred ist die Fähigkeit von DemFeature für jede Verbindung aus den Testdaten einen eigenen Lerndatensatz erstellen zu können. Das bedeutet, dass jede Verbindung in den Testdaten von einer spezifischen Scoring-Funktion vorhergesagt wird. Der Algorithmus zur Zusammenstellung des spezifischen Lerndatensatzes richtet sich nach der Ähnlichkeit der Verbindungen aus dem Lerndatensatz zu der Verbindung, für welche vorhergesagt wird. DemFeature beinhaltet zwei Versionen: DemFeature-1 und DemFeature-2. DemFeature-1 verwendet einen Grenzwert bezüglich der Ähnlichkeit um zu entscheiden, ob eine Verbindung dem spezifischen Lerndatensatz zugeordnet wird. Im Unterschied dazu wird bei DemFeature-2 eine feste Anzahl von Verbindungen im spezifischen Lerndatensatz vorgegeben, so dass diesem nur die ähnlichsten und unähnlichsten Verbindungen zugeordnet werden.

In dieser Doktorarbeit wurden zwei Datensätze verwendet, um die Vorhersagekraft von DemFeature zu prüfen. Der eine Datensatz bezieht sich auf Genotoxizität, welche im Wirkstoffentwurf eine wichtige Rolle spielt, und resultiert aus einem, durch Boehringer Ingelheim gegründeten, Wettbewerb. Der zweite Datensatz konzentriert sich auf Nebenwirkungen von Medikamenten (arzneimittelbedingte Phospholipidose), welche in den letzten Jahren hinsichtlich Arzneimittelsicherheit für Wissenschaftler aus dem pharmazeutischen Bereich interessant geworden sind. Verglichen mit DemPred konnte die Vorhersagekraft der generierten Modelle mit DemFeature verbessert werden. Die Vorhersagekraft übertraf sogar jene anderer aktueller Modelle zur Vorher.

Appendix 1.

The ID index of 1108 features

The original number of features in the Kaggle competition is 1776. Using DemPred with the L1 approach with λ_1=0.002, 1108 features remain.

1	248	524	763	979	1159	1349	1630
2	250	526	764	980	1160	1352	1631
5	251	528	765	981	1162	1353	1638
8	252	530	766	982	1163	1354	1639
10	253	534	767	983	1164	1356	1641
11	254	535	768	985	1165	1357	1642
13	256	539	769	986	1166	1358	1644
14	257	543	770	987	1167	1359	1645
16	258	544	772	988	1168	1361	1648
20	263	545	773	990	1169	1363	1649
21	266	546	774	991	1170	1364	1652
22	268	547	775	992	1172	1365	1656
24	272	548	776	993	1173	1366	1657
26	273	551	780	994	1174	1367	1659
27	275	552	782	995	1175	1368	1660
29	276	555	784	996	1176	1369	1663
31	277	561	785	997	1178	1370	1667
33	278	564	786	998	1179	1371	1669
36	279	566	788	1001	1180	1372	1675
37	280	568	791	1002	1182	1373	1676
39	281	569	792	1003	1184	1374	1678
40	282	570	793	1004	1185	1375	1682
41	284	571	794	1005	1186	1376	1684
42	287	573	795	1006	1187	1377	1686
43	288	574	796	1007	1188	1378	1687
44	289	575	797	1008	1190	1379	1692
45	291	576	798	1009	1191	1380	1695
46	292	577	799	1011	1192	1381	1696
50	293	578	800	1012	1193	1382	1700
51	294	579	801	1013	1194	1383	1701
52	296	580	802	1014	1195	1384	1703
54	297	585	804	1015	1196	1385	1704
55	298	586	805	1016	1197	1386	1706
57	300	588	806	1017	1198	1387	1707
59	301	589	807	1018	1202	1388	1715
63	303	591	808	1019	1203	1390	1727
65	304	594	809	1021	1204	1391	1730
66	307	596	810	1022	1205	1392	1731
68	310	597	811	1023	1206	1393	1732
71	311	598	813	1024	1207	1394	1734

72	312	599	819	1025	1208	1395	1735
75	313	601	820	1026	1209	1398	1737
77	314	602	821	1027	1210	1399	1739
78	315	603	823	1030	1212	1401	1740
79	317	604	825	1031	1213	1402	1746
80	319	605	826	1032	1214	1404	1751
81	320	606	829	1033	1215	1405	1753
82	321	608	830	1034	1217	1407	1755
83	322	609	831	1035	1218	1408	1758
84	323	610	832	1036	1219	1410	1759
86	324	611	833	1037	1221	1411	1760
90	327	612	834	1039	1222	1412	1762
91	328	615	835	1040	1223	1414	1766
92	329	619	837	1043	1224	1415	1768
94	330	620	838	1045	1225	1416	1769
98	331	622	839	1046	1226	1417	1770
99	336	625	844	1047	1227	1418	1772
100	338	629	845	1049	1228	1419	1773
101	342	632	846	1050	1229	1420	
103	343	634	847	1051	1231	1421	
105	344	635	851	1053	1232	1422	
107	346	636	857	1055	1233	1423	
108	348	637	858	1057	1234	1424	
109	350	641	859	1058	1236	1426	
110	351	642	860	1059	1237	1429	
112	352	643	861	1060	1238	1430	
113	354	644	862	1061	1239	1431	
116	356	645	865	1062	1240	1432	
118	358	647	868	1063	1241	1434	
119	359	648	870	1064	1242	1435	
121	364	649	871	1065	1243	1436	
123	365	650	872	1066	1245	1437	
124	366	651	873	1067	1246	1438	
127	367	652	875	1068	1247	1439	
128	369	653	878	1070	1249	1441	
129	370	655	880	1071	1250	1442	
132	371	656	881	1072	1251	1443	
134	372	657	882	1073	1252	1444	
136	373	658	883	1074	1254	1445	
140	374	659	884	1075	1255	1451	
143	375	660	885	1076	1256	1454	
144	376	661	886	1078	1258	1455	
145	378	663	887	1079	1261	1456	
146	379	664	888	1080	1262	1460	

149	382	668	889	1081	1264	1463
150	383	669	894	1082	1265	1466
151	384	670	895	1085	1266	1468
152	390	671	896	1086	1267	1472
154	391	673	897	1087	1268	1473
156	393	674	899	1088	1270	1474
159	394	675	900	1089	1271	1475
162	397	676	901	1090	1272	1476
166	400	677	902	1091	1273	1477
170	401	678	903	1092	1274	1478
171	405	679	904	1093	1275	1479
172	406	680	905	1094	1276	1486
174	408	681	906	1095	1277	1487
175	409	685	907	1096	1278	1488
176	410	688	909	1098	1280	1490
178	411	689	910	1099	1281	1496
179	412	690	911	1100	1282	1497
181	413	691	913	1101	1283	1499
182	414	693	914	1102	1284	1506
184	416	695	916	1103	1286	1507
185	417	696	917	1104	1287	1508
186	419	697	921	1105	1289	1510
188	421	698	922	1106	1290	1513
189	423	700	923	1107	1291	1516
192	427	701	924	1109	1292	1517
194	428	702	926	1110	1293	1525
195	430	703	928	1111	1294	1526
196	431	704	931	1113	1295	1538
197	432	705	934	1114	1296	1547
198	433	706	937	1115	1297	1552
199	435	707	938	1116	1298	1554
200	436	709	939	1117	1299	1555
201	439	710	940	1120	1301	1558
202	440	714	941	1121	1302	1560
204	441	716	942	1122	1303	1563
206	443	717	943	1124	1304	1564
207	444	718	944	1125	1305	1568
208	447	722	945	1126	1306	1570
212	450	723	946	1127	1307	1571
217	452	725	948	1128	1308	1572
218	455	726	949	1129	1309	1575
219	458	727	952	1130	1310	1576
220	461	728	953	1131	1312	1583
222	462	729	955	1132	1313	1585

223	463	730	956	1133	1314	1590
225	469	731	957	1134	1316	1591
227	470	732	958	1135	1317	1593
228	472	733	959	1137	1318	1595
230	478	734	960	1138	1319	1596
231	484	735	961	1139	1320	1597
232	488	740	962	1142	1322	1599
233	490	741	963	1143	1323	1601
234	491	742	964	1144	1325	1602
235	495	743	965	1145	1326	1604
236	498	744	966	1146	1327	1605
237	501	745	967	1147	1328	1608
238	502	746	969	1149	1330	1613
239	503	748	970	1150	1333	1616
240	504	750	971	1151	1334	1617
241	506	751	972	1152	1335	1618
242	507	753	973	1153	1336	1622
243	508	754	974	1154	1338	1623
244	510	755	975	1155	1339	1625
245	511	756	976	1156	1343	1627
246	517	758	977	1157	1345	1628
247	523	761	978	1158	1348	1629

Appendix 2.

39 structural patterns used in SMARTS model. The SMARTS features were designed by Przybylak et al., which were published in their article: How does the quality of phospholipidosis data influence the predictivity of structural alerts? Journal of Chemical Information and Modeling.,2014, 54(8), pp 2224-2232 DOI: 10.102/ci500233k.
Note: There are several typo errors in original publication. According to the suggestions from Dr. Jörg Bentzien of Boehringer-Ingelheim Inc., we have corrected them in this table.

structural group	**SMARTS pattern**
primary amine	[NH2][CX4;!R][CX4]
	[NH2][CX4](C)(C)(C)
	[NH2][C;R]([C;R][OH])[C;R][OH]
	[NH2][C;R]([CH;R])[C;R]O
	[NH2]CC1OCCCC1
	[NH2][C;R]([C;R])[C;R](O)[O;R]
	[NH2]c1c(Br)cc(Br)cc1
	[NH]-C([NH2])c1ccccc1
	[NH]-C([NH2])[NH]C[NH]
secondary amine	c[CX4;!R][CX4;!R][NH][CH2][CH3]
	[C;!R][NH][C;R]([C;R])c
	[C;!R][C;!R][C;!R][NH][C;!R][C;!R]c
	c1[cH1]c[cH1][cH1]c1[NH]c1[cH1][cH1][cH1][cH1]c1
	c[CH2][NH][C;!R][C;R]
	[CH3][NH][CH2;!R][CH2;!R]
	[CH3][NH][C;R]([C;R][OH])[C;R][OH]
	[CX4;!R][NH][CX4;!R][CX4;!R][CX4;!R]Oc1c2ccccc2ccc1
	[CX4][NH][CX4;R]
	c[OX2][CX4;!R][CX4;!R]([OH])[CX4;!R][NH][CH]([CH3])[CH3][a]
tertiary amine	[CX4;!R][N;!+]([CX4;!R])[CX4;!R][CX4]
	[CH3][N;!+]([CH3])[CH2][CH]=C(c)c
	[CH3][N;!+]([CH3])[CX4;R][CX4;R]([OH])[CX4;R][OX2;R][a]
	[CH3][N;!+]([CH3])[CX4;R][CX4;R]([CH3])[OX2;R][a]
cyclic amine	[nH]1n = ccc1
	cO[CH2][CH2][N;!+]1CCCC1
	[CH3][N;!+]1CC[N;!+]([C;!R])CC1
	[NH;R][C;R](C)[C;R]cc[C;R]
	[NH]1C(C)CCCC1
	[N;!+]1[CH2][CH2][CH]([NH])[CH2][CH2]1
	[N;!+]1[CH2][CH2]C(c2ccc(Cl)cc2)[CH2][CH2]1
	[n]c[OH][a]
	c[N;R][CX4;R][CX4;R][N;R][CX4][a]
	[CH3][N;R!+]([CX4;R])[CX4;R][a]
aromatic system	cN([CX4][CX4][CX4][NX3;!R])c
	cN([CX4][CX4][NX3])c

	cN([CX4][CX4][CX4;R][N;R])c
	c1ccccc1[CH2]c1ccccc1
	c[CH]([N;R])c
ring system	[R;a]
	[R;!a]
acidic groups	[#6,#1]C(=O)[OH]
	[CH](=O)[OH]
	[#6]N(=O)=O
	[#6][N+](=O)[O-]

Appendix 3.

prediction results for the 330 compounds of the Goracci PLD dataset

Gorraci PLD dataset[54], of which pipermide has been removed leading to 330 remaining compounds.

The Gorraci PLD dataset uses the dataset sources
T: Tomizawa *et al.* (2006); P: Pelletier *et al.* (2007); V: Vitovic *et al.* (2008);
H: Hanumegowda *et al.* (2010); F: Fisher *et al.* (2012); L: Lowe *et al.* (2010)
O: Orogo *et al.* (2012)

a: Dataset sources to which this compound belongs to.
b: compound generates PLD (+1) or not (-1).
c: Confirmed (yes) by transmission electron microscopy (TEM) or not (unclear) as stated in the above mentioned literature.
d: The Gorraci PLD dataset was divided into five subset (1, 2, 3, 4, 5). The four subsets, which do not contain the compound to be predicted, were used for training. The subset number containing the compound to be predicted is listed.

compound name	dataset source[a]	PLD property[b]	confirmed by TEM[c]	subset[d]	predicted value[e]
1-Chloroamitriptyline	P,L,O	1	yes	1	1.0691
Acetylcysteine	O	-1	unclear	1	-1.5109
Acitretin	O	-1	unclear	1	-0.5405
Amlodipine	H	-1	unclear	1	0.0848
Anagrelide hydrochloride	O	-1	unclear	1	0.1477
Aripiprazole	O	1	yes	1	0.0571
Bepotastine besilate	O	1	yes	1	0.2431
Bromocriptine mesylate	O	-1	unclear	1	-0.0711
Budesonide	O	-1	unclear	1	-0.6546
Bupivacaine	V,O	-1	unclear	1	0.7957
Calcitriol	O	-1	unclear	1	0.1848
Carisoprodol	O	-1	unclear	1	-0.3827
Ceftazidime	P,L,O	-1	yes	1	-1.7492
Chloroquine	T,P,V,H,L,F,O	1	yes	1	1.0623
Cimetidine	T,V,H,L,O	-1	yes	1	-1.2898
Clofibrate	P,L,O	-1	yes	1	-0.4146
Clomipramine	T,P,V,H,L,F,O	1	yes	1	1.1704
Deferoxamine (desferal)	P,L,O	-1	yes	1	-1.7229
Dexamethasone	O	-1	unclear	1	-1.0615
Diazepam	V,H,L,F,O	-1	yes	1	0.2892
Diclofenac	P,H,L,O	-1	yes	1	-0.336
Didanosine	O	-1	unclear	1	-0.9397
Dirithromycin	O	1	yes	1	1.1455
Disobutamide	P,L,O	1	yes	1	0.0196
Erlotinib	O	-1	unclear	1	-0.1547
Estradiol acetate	O	-1	unclear	1	-0.6177

Fenfluramine	T,P,V,H,L,O	1	yes	1	0.1798
Fenofibrate	T,P,V,L,O	-1	yes	1	-0.2765
Fenoterol	L,O	-1	yes	1	0.0955
Fluticasone propionate	O	-1	unclear	1	-1.0527
Fosinopril	O	-1	unclear	1	-0.2655
Gemfibrozil	P,H,L,O	-1	yes	1	-0.6949
Gentamicin-C1a	P,L	1	yes	1	0.7291
Guaifenesin	O	-1	unclear	1	-0.4351
Hydroxyurea	P,L,O	-1	yes	1	-1.8584
Iloprost	O	-1	unclear	1	-0.7258
Ipratropium	O	-1	unclear	1	0.9656
Isosorbide mononitrate	O	-1	unclear	1	-0.4382
Levothyroxine sodium	O	-1	unclear	1	1.6869
Metformin	P,L,O	-1	yes	1	-1.113
Metoclopramide	O	-1	unclear	1	-0.5525
Midodrine	O	-1	unclear	1	-0.5548
Montelukast sodium	O	-1	unclear	1	-0.0786
Naloxone hydrochloride dihydrate	O	-1	unclear	1	-1.0766
Nelfinavir mesylate	O	-1	unclear	1	-0.2311
Physostigmine	P,L,O	-1	yes	1	-0.2885
Prenylamine	L,O	1	yes	1	0.1991
Proguanil	O	-1	unclear	1	0.4175
Promazine	V,L,F,O	1	unclear	1	0.8273
Propafenone hydrochloride	O	-1	unclear	1	-0.0037
Rifabutin	O	1	yes	1	-0.167
Rifampin	P,H,L,O	-1	yes	1	-0.5247
RMI-10.393	P,L,O	1	yes	1	0.221
Rolitetracycline	P,H,L,O	-1	yes	1	-0.962
SDZ 200-125	P,L,O	1	yes	1	0.579
Simvastatin	O	-1	unclear	1	-0.3162
Stilbamidine	P,L,O	1	yes	1	0.2184
Sulindac	P,H,L,O	-1	yes	1	-0.3314
Tiagabine	O	-1	unclear	1	-0.1135
Tramadol hydrochloride	O	-1	unclear	1	-0.4898
Trimipramine	H,L,O	1	unclear	1	1.0494
Valproic acid (valproate)	T,P,V,H,L,F,O	-1	yes	1	-1.4324
Warfarin	T,V,H,O	-1	yes	1	-0.8676
WY-14643 (pirinixic acid)	P,L	-1	yes	1	-0.3969
Zidovudine	T,P,H,L,O	-1	yes	1	-1.2384
Zimelidine	T,P,H,L,O	1	yes	1	0.6298
1-Chloro-10-11-dehydroamitriptyline	P,L,O	1	yes	2	0.9263
3-Methylcholanthrene	P,L,O	-1	yes	2	0.4975

Adefovir dipivoxil	O	-1	unclear	2	-0.9381
Alosetron hydrochloride	O	-1	unclear	2	-0.1612
ANIT (1-naphthyl isothiocyanate)	P,L,O	-1	yes	2	0.3656
Anticoman	P,L,O	-1	yes	2	-0.2322
Atazanavir	O	-1	unclear	2	-0.2644
Atovaquone	O	-1	unclear	2	-0.3506
Azacosterol (20-25diazacholesterol)	P,L,O	1	yes	2	-0.115
Benzamide (BZ-1)	L,O	1	unclear	2	0.1369
Bicalutamide	P,L,O	-1	yes	2	-1.0079
Capsaicin	O	-1	unclear	2	-0.5048
Carbamazepine	P,H,LO	-1	yes	2	0.1996
Carbidopa	O	-1	unclear	2	-0.726
Carbon tetrachloride	P,L,O	-1	yes	2	-0.0707
Chloroform	P,L,O	-1	yes	2	-0.9616
Chlorpromazine	T,P,V,H,L,F,O	1	yes	2	0.005
Citalopram	T,P,H,L,F,O	1	yes	2	0.1185
Clociguanil	T,L,O	1	unclear	2	0.149
Cloforex	L,O	1	yes	2	-0.1563
Codeine sulfate	O	-1	unclear	2	-0.7478
Dibekacin	P,L,O	1	yes	2	0.9283
Diflunisal	P,L,O	-1	yes	2	-0.3135
Donepezil (aricept)	P,L,O	-1	yes	2	0.1765
Dronedarone	O	1	yes	2	0.2059
Emtricitabine	O	-1	unclear	2	-1.033
Galactosamine	P,L,O	-1	yes	2	-0.2486
Gentamicin-C2	P,L	1	yes	2	0.95
Glimepiride	O	-1	unclear	2	-0.1993
Hydrazine	P,L,O	-1	yes	2	-2.2344
Hydromorphone hydrochloride	O	-1	unclear	2	-1.1231
Hydroxychloroquine	H	1	unclear	2	0.6748
Hydroxyzine	P,V,H,L,F,O	1	yes	2	0.7408
IA3	P,L,O	1	yes	2	0.3116
Ibuprofen	O	-1	unclear	2	-0.7828
Imipramine	T,P,V,H,L,F,O	1	yes	2	1.0869
Ketotifen	O	1	yes	2	0.0703
Levalbuterol tartrate	O	-1	unclear	2	-0.7215
Methadone	P,H,L,O	-1	yes	2	-0.5551
Methapyrilene	P,H,L,O	-1	yes	2	-0.3604
Mianserin	T,V,H,L,O	1	unclear	2	0.4106
Midazolam	O	1	yes	2	0.4057
N-deacetyl-ketoconazole (DAKC)	P,V,H,L,O	1	unclear	2	0.425
Nortriptyline	T,V,L,O	1	yes	2	0.7357

Oxamniquine	O	-1	unclear	2	-0.3197
Perhexiline	T,P,V,H,L,F,O	1	yes	2	-0.3907
Phentermine	L,O	1	yes	2	0.1673
Pindolol	O	1	yes	2	0.1554
Quinacrine	V,H,L,F,O	1	yes	2	0.842
Sapropterin dihydrochloride	O	1	yes	2	-0.495
Sildenafil citrate	O	-1	unclear	2	-0.088
Streptomycin	F	-1	unclear	2	0.7814
Sumatriptan	L,O	-1	yes	2	0.4923
Tamoxifen	T,P,V,H,L,F,O	1	yes	2	0.7092
Telbivudine	O	-1	unclear	2	-1.0636
Temozolomide	O	-1	yes	2	-0.913
Tetrabenazine	O	-1	unclear	2	-0.0154
Tetracycline	T,V, H, L, O	-1	yes	2	-1.1793
Thioacetamide	T,P,V,L,O	-1	yes	2	-0.7919
Tiapride	O	-1	unclear	2	-0.6138
Tinidazole	O	-1	unclear	2	-0.9285
Tripelennamine	L,O	1	unclear	2	0.2121
Trospectomycin	P,L,F,O	1	yes	2	0.9734
Ursodiol	O	1	yes	2	-1.1014
Voriconazole	O	-1	unclear	2	-0.9644
Zolpidem tartrate	O	-1	unclear	2	-0.2405
Abacavir	P,H,L,O	-1	yes	3	-0.2329
ABT-518 formamide	L,O	-1	yes	3	-0.811
ABT-770 (parent)	L,O	-1	yes	3	-1.0854
Acetaminophen	T,P,V,H,L,O	-1	yes	3	-0.5231
Acetylsalicylic acid	P,L,O	-1	yes	3	-1.3426
Acyclovir	O	-1	unclear	3	-0.8649
Aliskiren hemifumarate	O	-1	unclear	3	-0.0619
Allopurinol	L,O	1	unclear	3	-0.8627
Ambroxol	L,O	1	yes	3	0.658
Amikacin	P,H,L,F,O	1	yes	3	0.5082
Amine metabolite of ABT-770	L,O	1	yes	3	-0.5775
Amiodarone	T,P,V,H,L,F,O	1	yes	3	0.4927
Atropine	T,V,H,L,O	1	yes	3	-0.1579
AY-9944	P,L,O	1	yes	3	0.3728
Beclomethasone dipropionate	O	-1	unclear	3	-0.8569
Bepridil	O	1	yes	3	0.4507
Betahistine	O	-1	unclear	3	-0.0195
Bisacodyl	O	-1	unclear	3	-0.381
Caffeine	P,L,O	-1	yes	3	-1.2237
Chlorphentermine	T,P,L,F,O	1	yes	3	0.9891
Chlortetracycline	P,H,L,O	-1	yes	3	-1.0473

Ciprofibrate	P,L,O	-1	yes	3	-1.2695
Clindamycin	P,L,O	1	yes	3	-0.1951
Clomacran (SKF-14336-D)	P,L,O	1	yes	3	0.7651
Clopidogrel bisulfate	O	-1	unclear	3	-0.5958
Colchicine	P,L,O	-1	yes	3	-0.6219
Demeclocycline	P,H,L,O	-1	yes	3	-1.0671
Dextromethorphan hydrobromide	O	-1	unclear	3	-0.1626
Digoxin	O	-1	unclear	3	0.4842
Doxorubicin	V,L,O	-1	yes	3	-1.0553
Drospirenone	O	-1	unclear	3	-0.3813
Emetine	P,L,O	1	yes	3	-0.2413
Fesoterodine	O	-1	unclear	3	-0.848
Fluoxetine	T,P,V,H,L,F,O	1	yes	3	-0.3216
Indoramin	P,H,L,O	1	yes	3	-0.202
Letrozole	O	-1	unclear	3	-0.6394
Linezolid	O	-1	unclear	3	-0.6873
Lysergide (or Lysergic-acid-diethylamide)	L,O	1	unclear	3	-0.4549
Meclizine	P,L,F,O	1	yes	3	0.6897
Methazolamide	O	-1	unclear	3	-0.872
Methylphenidate	O	-1	unclear	3	-0.1498
Moxifloxacin hydrochloride	O	-1	unclear	3	-1.0893
Naltrexone	O	-1	unclear	3	-1.3316
Nitisinone	O	-1	unclear	3	-0.9493
Omeprazole magnesium	O	-1	unclear	3	-1.224
Oxycodone	O	-1	unclear	3	-1.1387
Paliperidone	O	-1	unclear	3	0.1861
Palonosetron	O	-1	unclear	3	0.1142
Phenacetin	P,H,L,O	1	yes	3	-0.648
Phenobarbital (5-ethyl-5-phenylbarbituric acid)	P,H,L,F,O	-1	yes	3	-0.617
Pilocarpine hydrochloride	O	-1	unclear	3	-0.9969
Piroxicam	P,H,L,O	-1	yes	3	-0.7916
Prasugrel hydrochloride	O	-1	unclear	3	-0.6939
Prednisolone acetate	O	-1	unclear	3	-1.1378
Promethazine	L,O	1	unclear	3	0.2815
Quinidine	T,V,H,L,O	1	unclear	3	0.8026
Rasagiline mesylate	O	-1	unclear	3	-0.797
Salmeterol xinafoate	O	-1	unclear	3	-0.5446
Sirolimus	O	1	yes	3	0.9071
Spectinomycin	L,O	1	unclear	3	0.843
Spinosyn A	O	1	yes	3	0.7229
Spinosyn D	O	1	yes	3	0.594

Tapentadol hydrochloride	O	-1	unclear	3	-0.7757
Telithromycin	O	1	yes	3	0.6697
Vinblastine	L,O	-1	yes	3	-0.1958
Zoledronic acid	O	-1	unclear	3	-1.0596
1-Phenylpiperazine	V,O	-1	unclear	4	0.0998
2,3-deoxycytidine	T	-1	unclear	4	-0.9507
2,3-dideoxyinosine	T	-1	unclear	4	-0.8298
6-Hydroxydopamine	P,L	1	yes	4	-0.7859
ABT-518 (parent)	L,O	-1	yes	4	-0.9837
Amineptine	P,H,L,O	-1	yes	4	0.3629
Amitriptyline	T,P,V,H,L,O	1	yes	4	0.9578
Amodiaquine	P,H,L,O	1	yes	4	0.6686
Ampicillin	T,V,O	-1	yes	4	-0.9483
Atenolol	T,V,H,O	-1	yes	4	-0.4169
Atomoxetine hydrochloride	O	-1	unclear	4	0.4008
Azaserine	P,L,O	-1	yes	4	-1.0007
Azimilide dihydrochloride	L,O	1	yes	4	0.0046
Azithromycin	T,H,L,O	1	yes	4	0.7072
Chlorcyclizine	T,P,L,F,O	1	yes	4	0.9561
Chloroquine mustard	L,O	1	yes	4	0.9157
Clotrimazole troches	O	-1	unclear	4	0.1879
Cocaine	O	1	unclear	4	0.2263
Cyclizine	P,H,L,F,O	1	yes	4	0.7867
Cyproterone acetate	P,L,O	-1	yes	4	-0.8447
Darifenacin	O	-1	unclear	4	0.0006
Desvenlafaxine succinate	O	-1	unclear	4	-0.1953
DMP-777	L,O	1	unclear	4	-0.1722
Doxazosin mesylate	O	-1	unclear	4	-0.595
Doxycycline	P,H,L,O	-1	yes	4	-0.9619
Dronabinol	O	-1	unclear	4	-0.3365
Erythromycin	T,P,V,H,L,F,O	1	yes	4	0.5817
Ethyl loflazepate (ethyl flucozepate)	P,L,O	1	yes	4	-0.4705
Etoposide	P,L,O	-1	yes	4	-0.5392
Etravirine	O	-1	unclear	4	-0.0993
Fenisorex (R-800)	P,L,O	1	yes	4	0.1472
Fluconazole	O	-1	unclear	4	-0.8701
Fluvoxamine	L,O	1	unclear	4	0.1824
Furosemide	T,V,H,L,O	-1	yes	4	-0.5799
Haloperidol	V,H,L,F,O	1	yes	4	0.2662
Homochlorcyclizine	P,L,O	1	yes	4	0.9869
Isoniazide	T,V,H,O	-1	yes	4	-0.7226
L-Ethionine	P,L,O	-1	yes	4	-1.1625

Levodopa	O	1	yes	4	-0.9954
Levofloxacin (and ofloxacin)	L,O	-1	yes	4	-0.7022
Lidocaine	T,V,H,L,O	1	yes	4	-0.661
LY-281389	L,O	1	unclear	4	0.7747
Meloxicam	O	-1	unclear	4	-0.9852
Memantine	L,O	1	yes	4	0.8083
Menadione	L,O	-1	yes	4	-0.6907
Mycophenolate sodium	O	-1	unclear	4	-0.8131
Naproxen sodium	O	-1	unclear	4	-0.8145
Netilmicin	P,L,O	1	yes	4	0.7975
Nevirapine	O	1	yes	4	-0.2987
Nicotinic acid; Niacin	O	-1	unclear	4	-0.7772
Nitazoxanide	O	-1	unclear	4	-0.9742
Norethindrone acetate	O	-1	unclear	4	-0.8985
Noxiptiline	T,L,O	1	yes	4	0.6832
Paraquat	P,L,O	1	yes	4	-0.1925
Paroxetine hydrochloride	H,O	1	yes	4	0.1948
Propranolol	T,V,H,L,F,O	1	yes	4	0.1306
Quinine	T,H	1	yes	4	0.871
Rabeprazole	O	-1	unclear	4	-0.6755
Ranitidine	V,H,O	1	unclear	4	-0.8943
Saquinavir	O	-1	unclear	4	-0.6681
Selegiline hydrochloride	O	-1	unclear	4	-0.4875
Tadalafil	O	-1	unclear	4	-0.151
Tenofovir disoproxil fumarate	O	-1	unclear	4	-0.9605
Thioridazine	T,V,H,L,O	1	yes	4	0.8444
Tipranavir	O	-1	unclear	4	-0.6639
Verapamil	V,O	1	yes	4	-0.0477
17-alpha-Ethynylestradiol	P,L,O	-1	yes	5	-1.0824
4-Cyano-5-chlorophenyl-amidinourea	L,O	1	yes	5	-1.0003
ABT-770 formamide	L,O	-1	yes	5	-0.0007
AC-3579	P,L,O	1	yes	5	0.2469
Alfuzosin hydrochloride	O	-1	unclear	5	-0.7373
Alprostadil	O	-1	unclear	5	-0.4404
Amantadine	V,L,O	1	yes	5	0.8089
Amoxicillin	H,O	-1	unclear	5	-1.0027
Bilirubin	L,O	1	yes	5	-1.1661
Boxidine	P,L,O	1	yes	5	0.7614
Bromhexine	L,O	1	yes	5	0.975
Celecoxib	O	-1	unclear	5	-0.2744
Clemastine fumarate	O	1	yes	5	0.7858

Clonidine hydrochloride	O	-1	unclear	5	-0.7522
Clozapine	P,V,H,L,F,O	1	yes	5	0.8349
Coralgil	P,L,O	1	yes	5	0.5358
Dantrolene	P,L,O	-1	yes	5	-0.0826
Dasatinib	O	-1	unclear	5	-0.2288
Desipramine	V,H,LO	1	unclear	5	0.6808
Desloratadine	O	1	unclear	5	0.6368
Doxapram	P,H,L,O	-1	yes	5	-0.1706
Efavirenz	O	-1	unclear	5	-0.4596
Ethambutol hydrochloride	O	-1	unclear	5	-0.5313
Everolimus	O	1	yes	5	1.0409
Famotidine	T,P,V,H,L	-1	yes	5	-0.7893
Fexofenadine	O	-1	unclear	5	-0.1114
Flecainide	L,O	-1	yes	5	0.2733
Flucytosine	O	-1	unclear	5	-0.3424
Gatifloxacin hydrochloride	O	-1	unclear	5	-0.836
Gemifloxacin mesylate	O	-1	unclear	5	-0.6061
Gentamicin C1	P,L,O	1	yes	5	1.067
Hypoglycin-A	P,L,O	-1	yes	5	-0.6713
Iprindole	T,P,L,F,O	1	yes	5	0.473
Isoproterenol	O	-1	unclear	5	-0.4269
Ketoprofen	V,O	-1	unclear	5	-0.8353
Labetalol	T,V,H,L,O	1	yes	5	-0.4471
Lamivudine	O	-1	unclear	5	-0.975
Levonorgestrel	O	-1	unclear	5	-0.8581
Maprotiline	T,P,V,H,L,F,O	1	yes	5	0.0813
Mesoridazine	T,L,O	1	unclear	5	0.615
Methotrexate	P,L,O	-1	yes	5	-1.1029
Methyldopa	P,H,L,O	-1	yes	5	-0.0713
Morphine sulfate	O	-1	unclear	5	-0.9271
Nizatidine	O	-1	unclear	5	0.1184
Norchlorcyclizine	P,L,O	1	yes	5	0.7261
Oseltamivir phosphate	O	-1	unclear	5	-0.3941
Oxymetholone	O	-1	unclear	5	-0.2078
Pentamidine	T,V,H	1	yes	5	0.2411
PNU-177864	L,O	1	yes	5	-0.2742
Procainamide	L,O	-1	yes	5	-0.3167
Prochlorperazine maleate	O	-1	unclear	5	1.0158
Rifaximin	O	-1	unclear	5	-0.8211
Rosuvastatin calcium	O	-1	unclear	5	-0.9243
Sertindole	H	1	unclear	5	0.6397
Sertraline	H,L,O	1	unclear	5	0.4496
Spirapril	O	-1	unclear	5	-0.8447

Sulfamethoxazole	L,O	-1	yes	5	-0.762
Tilorone	T,P,L,F,O	1	yes	5	0.67
Tobramycin	P,H,L,O	1	yes	5	0.9099
Tolterodine tartrate	O	-1	unclear	5	-0.1532
Triparanol	T,P,L,O	1	yes	5	0.5609
Valganciclovir hydrochloride	O	-1	unclear	5	-0.6818
Varenicline tartrate	O	1	yes	5	0.2259
Zalcitabine	O	-1	unclear	5	-0.8841
Zileuton	P,L,O	-1	yes	5	-0.5166
Zonisamide	O	1	yes	5	-0.9573

References

[1] FDA, U.S., The drug development process, Available at www.fda.gov/ForPatients/Approvals/Drugs/default.htm, (last accessed 2016).

[2] Ciociola, Arthur A., Cohen, Lawrence B., Kulkarni, Prasad, and the, F. D. A. Related Matters Committee of the American College of Gastroenterology, How Drugs are Developed and Approved by the FDA: Current Process and Future Directions. *Am J Gastroenterol* **109** (5), 620 (2014).

[3] Suvarna, Viraj, Phase IV of Drug Development. *Perspectives in Clinical Research* **1** (2), 57 (2010).

[4] Berndt, Ernst R., Gottschalk, Adrian H. B., and Strobeck, Matthew W., in *Innovation Policy and the Economy, Volume 6* (The MIT Press, 2006), pp. 91.

[5] Adams, Christopher Paul and Brantner, Van Vu, Spending on new drug development1. *Health Economics* **19** (2), 130; Holland, John, Fixing a broken drug development process. *Journal of Commercial Biotechnology* **19** (1) (2013).

[6] Adams, Christopher P. and Brantner, Van V., Estimating The Cost Of New Drug Development: Is It Really $802 Million? *Health Affairs* **25** (2), 420 (2006).

[7] Kola, Ismail and Landis, John, Can the pharmaceutical industry reduce attrition rates? *Nature reviews Drug discovery* **3** (8), 711 (2004).

[8] Arrowsmith, John, Trial watch: Phase III and submission failures: 2007â€“2010. *Nat Rev Drug Discov* **10** (2), 87 (2010).

[9] Kapetanovic, I. M., COMPUTER-AIDED DRUG DISCOVERY AND DEVELOPMENT (CADDD): in silico-chemico-biological approach. *Chemico-biological interactions* **171** (2), 165 (2008).

[10] Boyd, Donald B., How computational chemistry became important in the pharmaceutical industry. *Reviews in computational chemistry* **23**, 401 (2007).

[11] Guener, OSMAN F., *Pharmacophore Perception, Development, and use in Drug Design* (International University Line, La jolla, California, 2000).

[12] Mauser, Harald and Guba, Wolfgang, Recent developments in de novo design and scaffold hopping. *Current opinion in drug discovery & development* **11** (3), 365 (2008).

[13] Hansch, Corwin, Quantitative approach to biochemical structure-activity relationships. *Accounts of Chemical Research* **2** (8), 232 (1969).

[14] Hansch, Corwin, Maloney, Peyton P., Fujita, Toshio, and Muir, Robert M., Correlation of biological activity of phenoxyacetic acids with Hammett substituent constants and partition coefficients. *Nature* **194**, 178 (1962).

[15] Hansch, Corwin and Fujita, Toshio, p-Ïƒ-Ï€ Analysis. A method for the correlation of biological activity and chemical structure. *Journal of the American Chemical Society* **86** (8), 1616 (1964).

[16] Fujita, Toshio, Iwasa, Junkichi, and Hansch, Corwin, A new substituent constant, Ï€, derived from partition coefficients. *Journal of the American Chemical Society* **86** (23), 5175 (1964).

[17] Marchant, Carol A., Briggs, Katharine A., and Long, Anthony, In silico tools for sharing data and knowledge on toxicity and metabolism: derek for windows, meteor, and vitic. *Toxicology mechanisms and methods* **18** (2-3), 177 (2008).

[18] Matthews, Edwin J. and Contrera, Joseph F., A New Highly Specific Method for Predicting the Carcinogenic Potential of Pharmaceuticals in Rodents Using EnhancedMCASEQSAR-ES Software. *Regulatory Toxicology and Pharmacology* **28** (3), 242 (1998).

[19] Enslein, Kurt, Gombar, Vijay K., and Blake, Benjamin W., Use of SAR in computer-assited prediction of carcinogenicity and mutagenicity of chemicals by the TOPKAT program. *Mutation Research/Fundamental and Molecular Mechanisms of Mutagenesis* **305** (1), 47 (1994).

[20] Du, Qi-Shi, Huang, Ri-Bo, and Chou, Kuo-Chen, Recent advances in QSAR and their applications in predicting the activities of chemical molecules, peptides and proteins for drug design. *Current Protein and Peptide Science* **9** (3), 248 (2008).

[21] Ivanciuc, Teodora, Ivanciuc, Ovidiu, and Klein, Douglas J., Posetic quantitative superstructure/activity relationships (QSSARs) for chlorobenzenes. *Journal of chemical information and modeling* **45** (4), 870 (2005).

[22] Gertrudes, J. C. et al., Machine learning techniques and drug design. *Current medicinal chemistry* **19** (25), 4289 (2012).

[23] De Benedetti, Pier G. and Fanelli, Francesca, Computational quantum chemistry and adaptive ligand modeling in mechanistic QSAR. *Drug Discovery Today* **15** (19), 859 (2010).

[24] Alberto Castillo-Garit, Juan et al., A review of QSAR studies to discover new drug-like compounds actives against leishmaniasis and trypanosomiasis. *Current topics in medicinal chemistry* **12** (8), 852.

[25] Sharma, O. P. et al., Evolutionary History of QSAR: A Review. *J. Natur. Cons* **1** (4), 266 (2013).

[26] Witten, Ian H. and Frank, Eibe, *Data Mining: Practical machine learning tools and techniques*. (Morgan Kaufmann, 2005).

[27] Moss, G. P. et al., The application of discriminant analysis and Machine Learning methods as tools to identify and classify compounds with potential as transdermal enhancers. *European Journal of Pharmaceutical Sciences* **45** (1), 116 (2012).

[28] Yang, Xue-Gang, Lv, Wei, Chen, Yu-Zong, and Xue, Ying, In silico prediction and screening of gamma-secretase inhibitors by molecular descriptors and machine learning methods. *Journal of computational chemistry* **31** (6), 1249 (2009).

[29] Essa, Ali Hashem, Ibrahim, Medhat, Hameed, Ali Jameel, and Al-Masoudi, Najim A., Theoretical investigation of 3'-subtituted-2'-3'-dideoxythymidines related to AZT. QSAR infrared and substituent electronic effect studies. *Arkivoc* **13**, 255 (2008).

[30] Ibrahim, M., A Saleh, N., M Elshemey, W., and A Elsayed, A., Fullerene derivative as anti-HIV protease inhibitor: molecular modeling and QSAR approaches. *Mini reviews in medicinal chemistry* **12** (6), 447 (2012).

[31] Cortes, Corinna and Vapnik, Vladimir, Support-vector networks. *Machine Learning* **20** (3), 273 (1995).

[32] Boser, Bernhard E., Guyon, Isabelle M., and Vapnik, Vladimir N., presented at the Proceedings of the fifth annual workshop on Computational learning theory, 1992 (unpublished).

[33] William, H., Teukolsky, Saul A., Vetterling, William T., and Flannery, B. P., Section 16.5. support vector machines. *Numerical Recipes: The Art of Scientific Computing* (2007).

[34] Wu, X.D., et al. Top 10 Algorithms in Data Mining. *Know Inf Syst*, **14**(1),1 (2008)

[35] Quinlan, J. R., Simplifying decision trees. *International Journal of Man-Machine Studies* **27** (3), 221 (1987).

[36] Quinlan, J. R., *C4.5: programs for machine learning*. (Morgan Kaufmann Publishers, San Francisco, CA, 1993).

[37] Chambers, PhilipL et al., in *Receptors and Other Targets for Toxic Substances* (Springer Berlin Heidelberg, 1985), Vol. 8, pp. 173.

[38] Lira, Felipe, Perez, Pedro S., Baranauskas, Jose A., and Nozawa, Sergio R., Prediction of Antimicrobial Activity of Synthetic Peptides by a Decision Tree Model. *Applied and Environmental Microbiology* **79** (10), 3156 (2013).
[39] Breiman, Leo, Random Forests. *Machine Learning* **45** (1), 5 (2001).
[40] Ho, Tin Kam, presented at the Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14-16 Aug. 1995 (unpublished); Ho, Tin Kam, The Random Subspace Method for Constructing Decision Forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20** (8), 832 (1998).
[41] Goldstein, Benjamin A., Polley, Eric C., and Briggs, Farren B. S., Random Forests for Genetic Association Studies. *Statistical Applications in Genetics and Molecular Biology* **10** (1), 32 (2011).
[42] Chen, X., Wang, M. and Zhang, H., The use of classification trees for bioinformatics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **1** (1), 55 (2011).
[43] Bishop, C.M., *Neural Networks for Pattern Recognition*. (Oxford University Press, Oxford, 1995).
[44] Rumelhart, David E.; Hinton, Geoffrey E.; Williams, Ronald J, Learning representations by back-propagating errors. *Nature* **6088** (323), 533 (1986).
[45] Broomhead, D. S.; Lowe, David, Multivariable functional interpolation and adaptive networks. *Complex Systems* (2), 321 (1988).
[46] Beaufays, H. Sak and A. W. Senior and F., *in Google speech*, 1468 (2014).
[47] Basheer, I. A. and Hajmeer, M., Artificial neural networks: fundamentals, computing, design, and application. *Journal of Microbiological Methods* **43** (1), 3 (2000).
[48] Khan, Javed et al., Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nat Med* **7** (6), 673 (2001).
[49] Rogers, Mark A. et al., Proteomic Profiling of Urinary Proteins in Renal Cancer by Surface Enhanced Laser Desorption Ionization and Neural-Network Analysis: Identification of Key Issues Affecting Potential Clinical Utility. *Cancer Research* **63** (20), 6971 (2003).
[50] Zhang, Harry, presented at the The 17th International FLAIRS2004 Conference, Miami Beach, FL, USA, 2004 (unpublished).
[51] Caruana, R.; Niculescu Mizil. A. , presented at the 23rd International Conference on Machine Learning, 2006 (unpublished).
[52] Ripley, B, D, *Pattern Recognition and Neural Networks*. (Cambridge University Press, Cambridge, 1996).
[53] Jörg Bentzien, Ingo Muegge, Ben Hamner and David C. Thompson, Crowd computing: using competitive dynamics to develop and refine highly predictive models. *Drug Discovery Today* **18**, 472 (2013).
[54] Goracci, Laura, Ceccarelli, Martina, Bonelli, Daniela, and Cruciani, Gabriele, Modeling phospholipidosis induction: reliability and warnings. *Journal of chemical information and modeling* **53** (6), 1436 (2013).
[55] Isabelle Guyon, André Elisseeff, An Introduction to Variable and Feature Selection. *Journal of Machine Learning Researc* **3**, 1157 (2003).
[56] Pearson, Karl, Notes on regression and inheritance in the case of two parents. *Proceedings of the royal society of london* **58**, 240 (1895).
[57] Zou, Hui and Hastie, Trevor, Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **67** (2), 301 (2005).

[58] Todeschini R., Consonni V. and Manuela P., Dragon software: an easy approach to molecular descriptor calculations. *MATCH Commun. Math. Comput. Chem.* **56**, 237 (2006).
[59] Hall L.H., Kellogg G.E., Haney G.H., , (eduSoft, LC., 2002).
[60] Wegner, J.K., JOELib2, Available at http://www.ra.cs.uni-tuebingen.de/joelib/index.html, (last accessed 2016).
[61] Xue, Y., Yap, C.W., Sun, L.Z., Chen, X., Chen., Y. Z., Effect of molecular descriptor feature selection in support vector machine classification of pharmcokinetic and toxicological properties of chemical agents. *J Chem Inf Comput Sci* **44** (5), 1630 (2004).
[62] Li, Z. R., Han, L. Y., Xue, Y., Yap, C. W., Li, H., Jiang, L., Chen, Y. Z., MODEL-Molecular Descriptor Lab: A Web-Based Server for Computing Structural and Physicochemical Features of Compounds. *Biotechnology and Bioengineering* **97** (2), 389 (2007).
[63] Steinbeck, C., Hoppe, C., Kuhn, S., Floris, M., Guha, R. and Willighagen, E. L. , Recent developments of the chemistry development kit (CDK)-an open-source java library for chemo- and bioinformatics. *Curr Pharm Des.* **12** (17), 2111 (2006).
[64] Pipepline Pilot, Available at http://accelrys.com/ (last accessed 2016).
[65] VolSurf, Available at http://www.moldiscovery.com/index.php (last accessed 2016).
[66] Dong, J., Cao, D. S., Miao, H. Y., Liu, S., Deng, B.C., Yun, Y.H., Wang, N. N., Lu, A. P., Zeng, W. B. and Chen, A. F,, ChemDes: an integrated web-based platform for molecular descriptor and fingerprint computation. *J cheminform.* **7:60**, eCollection (2015).
[67] MOE: Molecular Operating Environment, Available at http://www.chemcomp.com/ (last accessed 2016).
[68] Geisser, Seymour *Predictive inference: an introduction*. (Chapman and Hall, New York, NY, 1993).
[69] Kohavi, Ron, A study of cross-validation and boostrap for accuracy estimation and model selection. *Proceedings of Fourteenth International Joint Conference on Artificial Intelligence* **2**, 1137 (1995).
[70] McLachlan, Geoffrey; Do, Kim-Anh; Ambroise, Christophe, *Analyzing Microarray Gene Expression Data*. (John Wiley & Sons, 2004).
[71] Mattews, B. W., Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure* **405** (2), 442 (1975).
[72] Wikipedia, Sensitivity and specificity, Available at https://en.wikipedia.org/wiki/Sensitivity_and_specificity (last accessed 2016).
[73] Duda, Hart and Hart, Peter, (John Wiley & Sons, 2001).
[74] Webb, A. R., *Statistical Pattern Recognition 2nd Edition*. (John Wiley & SOns, Malvern, UK, 2002).
[75] Riedmiller, Martin and Braun, Heinrich, presented at the Proc. of ISCIS VII), Universitat, 1992 (unpublished); Riedmiller, Martin and Braun, Heinrich, presented at the Neural Networks, 1993., IEEE International Conference on, 1993 (unpublished).
[76] Goodman, Joshua, presented at the HLT-NAACL, 2004 (unpublished).
[77] Andrew, Galen and Gao, Jianfeng, presented at the Proceedings of the 24th international conference on Machine learning, 2007 (unpublished).
[78] Hoerl, Arthur E. and Kennard, Robert W., Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* **12** (1), 55 (1970).
[79] DemPred, Available at http://agknapp.chemie.fu-berlin.de/agknapp/?menu=software&page=dempred (last accessed 2016) .

[80] Demir-Kavuk, Ozgur, Riedesel, Henning, and Knapp, Ernst-Walter, Exploring classification strategies with the CoEPrA 2006 contest. *Bioinformatics* **26** (5), 603 (2010).
[81] Demir-Kavuk, Ozgur, Bentzien, JÃ¶rg, Muegge, Ingo, and Knapp, Ernst-Walter, DemQSAR: predicting human volume of distribution and clearance of drugs. *Journal of computer-aided molecular design* **25** (12), 1121 (2011).
[82] Demir-Kavuk, Ozgur, Krull, Florian, Chae, M. H., and Knapp, Ernst-Walter, presented at the Genome informatics. International Conference on Genome Informatics, 2009 (unpublished).
[83] Salzberg, Steven L., On comparing classifiers: Pitfalls to avoid and a recommended approach. *Data mining and knowledge discovery* **1** (3), 317 (1997).
[84] Everitt, Brian S., *The analysis of contingency tables*. (CRC Press, 1992).
[85] Hansen, Katja et al., Benchmark data set for in silico prediction of Ames mutagenicity. *Journal of chemical information and modeling* **49** (9), 2077 (2009).
[86] Sushko, Iurii et al., Applicability domains for classification problems: benchmarking of distance to models for Ames mutagenicity set. *Journal of chemical information and modeling* **50** (12), 2094 (2010).
[87] BioByte clogP v. 5.3, (2010). Availabl from: http://www.biobyte.com/index.html
[88] Carhart, Raymond E., Smith, Dennis H., and Venkataraghavan, R., Atom pairs as molecular features in structure-activity studies: definition and applications. *Journal of Chemical Information and Computer Sciences* **25** (2), 64 (1985).
[89] Daylight v4.49. Available from: http://accelrys.com/ (last accessed 2016)
[90] Demir-Kavuk, Ozgur, Kamada, Mayumi, Akutsu, Tatsuya, and Knapp, Ernst-Walter, Prediction using step-wise L1, L2 regularization and feature selection for small data sets with large number of features. *BMC bioinformatics* **12** (1), 412 (2011).
[91] CoEPrA, Available at http://www.coepra.org (last accessed 2016).
[92] Brown, Michael P. S. et al., Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proceedings of the National Academy of Sciences* **97** (1), 262 (2000).
[93] Pedregosa, Fabian et al., Scikit-learn: Machine learning in Python. *Journal of machine learning research* **12** (Oct), 2825 (2011).
[94] Reasor, Mark J. and Kacew, Sam, Drug-induced phospholipidosis: are there functional consequences? *Experimental Biology and Medicine* **226** (9), 825 (2001).
[95] Kruhlak, Naomi L. et al., Development of a phospholipidosis database and predictive quantitative structure-activity relationship (QSAR) models. *Toxicology mechanisms and methods* **18** (2-3), 217 (2008).
[96] Fischer, Holger et al., In silico assay for assessing phospholipidosis potential of small druglike molecules: training, validation, and refinement using several data sets. *Journal of medicinal chemistry* **55** (1), 126 (2012).
[97] Przybylak, Katarzyna R., Alzahrani, Abdullah Rzgallah, and Cronin, Mark T. D., How does the quality of phospholipidosis data influence the predictivity of structural alerts? *Journal of chemical information and modeling* **54** (8), 2224 (2014).
[98] Chatman, Linda A., Morton, Daniel, Johnson, Theodore O., and Anway, Susan D., A strategy for risk management of drug-induced phospholipidosis. *Toxicologic pathology* **37** (7), 997 (2009).
[99] Nelson, A. A. and Fitzhugh, O. G., Chloroquine (SN-7618) pathologic changes observed in rats which for 2 years had been fed various proportions. *Archives of pathology* **45** (4), 454 (1948).

[100] Liu, Yang, Kam, Wendy R., Ding, Juan, and Sullivan, David A., One man's poison is another man's meat: using azithromycin-induced phospholipidosis to promote ocular surface health. *Toxicology* **320**, 1 (2014).

[101] Goracci, Laura et al., Evaluating the risk of phospholipidosis using a new multidisciplinary pipeline approach. *European journal of medicinal chemistry* **92**, 49 (2015).

[102] Orogo, Amabel M., Choi, Sydney S., Minnier, Barbara L., and Kruhlak, Naomi L., Construction and consensus performance of (Q) SAR models for predicting phospholipidosis using a dataset of 743 compounds. *Molecular Informatics* **31** (10), 725 (2012).

[103] Reasor, Mark J., Hastings, Kenneth L., and Ulrich, Roger G., Drug-induced phospholipidosis: issues and future directions. *Expert opinion on drug safety* **5** (4), 567 (2006).

[104] Slavov, Svetoslav H. et al., Computational identification of a phospholipidosis toxicophore using 13 C and 15 N NMR-distance based fingerprints. *Bioorganic & medicinal chemistry* **22** (23), 6706 (2014).

[105] Choi, Sydney S., Kim, Jae S., Valerio, Luis G., and Sadrieh, Nakissa, In silico modeling to predict drug-induced phospholipidosis. *Toxicology and applied pharmacology* **269** (2), 195 (2013).

[106] Hanumegowda, Umesh M. et al., Phospholipidosis as a function of basicity, lipophilicity, and volume of distribution of compounds. *Chemical research in toxicology* **23** (4), 749 (2010).

[107] Law, Vivian et al., DrugBank 4.0: shedding new light on drug metabolism. *Nucleic acids research* **42** (D1), D1091 (2014).

[108] Nioi, Paul et al., In Vitro Detection of Drug-Induced Phospholipidosis Using Gene Expression and Fluorescent Phospholipidâ€“Based Methodologies. *Toxicological sciences* **99** (1), 162 (2007).

[109] Sawada, Hiroshi, Takami, Kenji, and Asahi, Satoru, A toxicogenomic approach to drug-induced phospholipidosis: analysis of its induction mechanism and establishment of a novel in vitro screening system. *Toxicological sciences* **83** (2), 282 (2005).

[110] Stebbins, K. E., Bond, D. M., Novilla, M. N., and Reasor, M. J., Spinosad insecticide: subchronic and chronic toxicity and lack of carcinogenicity in CD-1 mice. *Toxicological sciences* **65** (2), 276 (2002);

[111] Vonderfecht, Steven L. et al., Myopathy related to administration of a cationic amphiphilic drug and the use of multidose drug distribution analysis to predict its occurrence. *Toxicologic pathology* **32** (3), 318 (2004).

[112] Morelli, J. K. et al., Validation of an in vitro screen for phospholipidosis using a high-content biology platform. *Cell biology and toxicology* **22** (1), 15 (2006).

[113] Vitovic, Pavol, Alakoskela, Juha-Matti, and Kinnunen, Paavo K. J., Assessment of Drugâˆ’ Lipid Complex Formation by a High-Throughput Langmuir-Balance and Correlation to Phospholipidosis. *Journal of medicinal chemistry* **51** (6), 1842 (2008).

[114] Jiang, Zhengjin and Reilly, John, Chromatography approaches for early screening of the phospholipidosis-inducing potential of pharmaceuticals. *Journal of pharmaceutical and biomedical analysis* **61**, 184 (2012).

[115] Ploemen, Jan-Peter H. T. M. et al., Use of physicochemical calculation of pKa and CLogP to predict phospholipidosis-inducing potential: a case study with structurally related piperazines. *Experimental and Toxicologic Pathology* **55** (5), 347 (2004).

[116] Tomizawa, Kaori, Sugano, Kiyohiko, and Yamada, Hiroshi, Physicochemical and cell-based approach for early screening of phospholipidosis-inducing potential. *The Journal of toxicological sciences* **31** (4), 315 (2006).

[117] Pelletier, Dennis J. et al., Evaluation of a published in silico model and construction of a novel Bayesian model for predicting phospholipidosis inducing potential. *Journal of chemical information and modeling* **47** (3), 1196 (2007).

[118] Przybylak, Katarzyna R. and Cronin, Mark T. D., In silico studies of the relationship between chemical structure and drug induced phospholipidosis. *Molecular Informatics* **30** (5), 415 (2011).

[119] Ivanciuc, Ovidiu, Weka machine learning for predicting the phospholipidosis inducing potential. *Current topics in medicinal chemistry* **8** (18), 1691 (2008).

[120] Lowe, Robert, Glen, Robert C., and Mitchell, John B. O., Predicting phospholipidosis using machine learning. *Molecular pharmaceutics* **7** (5), 1708 (2010).

[121] Valerio, Luis G., In silico toxicology for the pharmaceutical sciences. *Toxicology and applied pharmacology* **241** (3), 356 (2009).

[122] Milletti, Francesca, Storchi, Loriano, Sforna, Gianluca, and Cruciani, Gabriele, New and original p K a prediction method using grid molecular interaction fields. *Journal of chemical information and modeling* **47** (6), 2172 (2007).

[123] Rogers, David and Hahn, Mathew, Extended-connectivity fingerprints. *Journal of chemical information and modeling* **50** (5), 742 (2010).

[124] Kier, L. B. and Hall, L. H., (San Diego: Academic Press 1999).

[125] Viswanadhan, Vellarkad N., Ghose, Arup K., Revankar, Ganapathi R., and Robins, Roland K., Atomic physicochemical parameters for three dimensional structure directed quantitative structure-activity relationships. 4. Additional parameters for hydrophobic and dispersive interactions and their application for an automated superposition of certain naturally occurring nucleoside antibiotics. *Journal of Chemical Information and Computer Sciences* **29** (3), 163 (1989).

[126] Pipeline Pilot v9.1.0.13, Accelrys, San Diego USA, Cited; Available from http://accelrys.com (last accessed 2016).

[127] Durant, Joseph L., Leland, Burton A., Henry, Douglas R., and Nourse, James G., Reoptimization of MDL keys for use in drug discovery. *Journal of Chemical Information and Computer Sciences* **42** (6), 1273 (2002).

[128] Verloop, A., *The STERIMOL approach to drug design*. (Marcel Dekker, New York, 1987).

[129] Talete: Talete srl, DRAGON for Linux, v6.0.32, Milano, Italy (2013).

[130] Lewell, Judd, Watson and Hann MM, RECAP--retrosynthetic combinatorial analysis procedure: a powerful new technique for identifying privileged molecular fragments with useful applications in combinatorial chemistry. *Journal of Chemical Information and Computer Sciences* **38** (3), 511-22 (1998).

[131] Schuffenhauer, Ansgar, et al., The scaffold tree-visualization of the scaffold universe by hierarchical scaffold classification. *Journal of chemical information and modeling* **47**(1), 47-58 (2007).

[132] Michael, Berthold, et al., KNIME: The Konstanz Information Miner. *Studies in Classification, Data Analysis, and Knowledge Organization. GfKL* (2007).