Novel Machine Learning Methods for Computational Chemistry

vorgelegt von

Master of Science
Katja Hansen
aus Hamburg

Von der Fakultät IV - Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades
Doktor der Naturwissenschaften
- Dr. rer. nat. -

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof.Dr. Manfred Opper
Gutachter: Prof.Dr. Klaus-Robert Müller
Gutachter: Prof.Dr. Gisbert Schneider
Gutachter: Prof.Dr. Graeme Henkelman

Tag der wissenschaftlichen Aussprache: 19.06.2012

Berlin 2012

D 83

Abstract

The experimental assessment of absorption, distribution, metabolism, excretion, toxicity and related physiochemical properties of small molecules is counted among the most time- and cost-intensive tasks in chemical research. Computational approaches, such as machine learning methods, represent an economic alternative to predict these properties, however, the limited accuracy and irregular error rate of these predictions restrict their use within the research process. This thesis introduces and evaluates new ideas to enhance the acceptance and usage of kernel-based machine learning models in chemical research.

The first part of the thesis investigates different approaches to improve the quality of machine learning predictions in drug discovery. By taking the precise chemical application into account we derive a new virtual screening algorithm, StructRank, which enables to focus on the correct ranking of compounds with high binding affinities. Then, the limits of single and ensemble learning methods are analyzed in the context of hERG inhibition. Since the drug discovery process often requires the assessment of new chemical series different to previously examined structures, we introduce and evaluate a clustered cross-validation scheme that stresses the extrapolation capacity of models. We present a local bias correction to incorporate new measurements efficiently and without the need for model retraining.

The second part of the thesis is concerned with two different approaches to assess the reliability and interpretability of kernel-based prediction models. The first approach builds on the visual interpretation of predictions based on the most relevant training compounds. A compact method to calculate the impact of training compounds on single predictions is derived and the resulting visualizations are evaluated in a questionnaire study. The second approach addresses interpretability in terms of chemical features. Here, local gradients are employed to measure the local influence of specific chemical features on a predicted property. The capacity of this approach to identify local as well as global trends in Ames mutagenicity data, and, to reveal unique characteristics of compound classes such as steroids is depicted. Finally, we show that the potential of the developed methods extends beyond drug discovery by using local gradients to enhance the assessment of reaction rates in transition state theory.

While computational chemistry remains a challenging field of application for machine learning, the present work introduces methods to improve and assess the quality of machine learning predictions in order to increase the usage of these methods in chemical research.

Zusammenfassung

Die Untersuchung komplexer pharmakokinetischen Eigenschaften, wie Absorption, Disposition, Metabolismus oder Toxizität, ist bei Arzneistoffen mit einem enormen experimentellen Aufwand und erheblichen Kosten verbunden. Computergestützte Vorhersageverfahren, wie maschinelle Lernverfahren, können diese Eigenschaften vorhersagen und stellen somit eine effiziente Alternative zum experimentellen Ansatz dar. Allerdings werden diese Verfahren aufgrund ihrer oft unklaren und wechselhaften Genauigkeit nur zögerlich eingesetzt. Ziel dieser Arbeit ist es, die Akzeptanz und die Anwendungsmöglichkeiten von maschinellen Lernverfahren in der chemischen Forschung zu erweitern.

Im ersten Teil der Arbeit steht die Verbesserung von kernbasierten maschinellen Lenverfahren in Bezug auf die Anwendungen in der Wirkstoffforschung im Vordergrund. Im ersten Kapitel wird ein neuer Algorithmus, StructRank, für das virtuelle Screening entwickelt. Dieser Algorithmus ist ideal an die Anforderungen des virtuellen Screenings angepasst, da er eine Rangordnung von Molekülen vorhersagt und Moleküle mit einer hohen Bindungsaffinität besonders stark berücksichtigt. Das zweite Kapitel beschäftigt sich mit dem Vergleich und der Kombination von Lernverfahren zu einem leistungsstärkeren Ensemble. Anhand von Daten zur Inhibition des hERG Rezeptors werden die Grenzen und Möglichkeiten verschiedener Verfahren untersucht. Eine lokale Bias-Korrektur kristallisiert sich hierbei als ein schnelles und effizientes Verfahren zur Einbindung neuer Messergebnisse ohne erneute Anpassung des Modells heraus. Im Rahmen dieser Studie wird auch ein neues Kreuz-Validierungs-Schema untersucht, welches das Extrapolationsvermögen von Prädiktionsmodellen stärker berücksichtigt. Das Extrapolationsvermögen ist in der chemischen Forschung von besonderer Bedeutung, da die neu zu untersuchenden Verbindungen sich oftmals deutlich von allen zuvor untersuchten Molekülen unterscheiden.

Im zweiten Teil der Arbeit werden neue Ansätze zur Bewertung und Interpretation computergestützter Vorhersagen untersucht. Zunächst wird ein Verfahren zur Berechnung des Einflusses einzelner Trainingsdatenpunkte auf eine Vorhersage hergeleitet. Anschließend werden die einflussreichsten Verbindungen als Erklärungshilfen zusammen mit der Vorhersage visualisiert und dieser Erklärungsansatz in einer empirischen Studie evaluiert. Lokale Gradienten repräsentieren einen zweiten neuen Ansatz zur Interpretation von Vorhersagen. Sie messen den lokalen Einfluss einzelner chemischer Eigenschaften auf die Vorhersage. Mit diesem Verfahren werden sowohl globale als auch lokale Tendenzen auf einem Datensatz zur Ames Mutagenität erfasst und Besonderheiten von Verbindungsklassen, wie z.B. Steroiden identifiziert. Eine Studie zur Berechnung von Reaktionsraten mit Hilfe von lokalen Gradienten im Rahmen der Theorie des Übergangszustandes verdeutlicht abschießend die Relevanz der erarbeiteten Verfahren außerhalb der Wirkstoffforschung.

Insgesamt beinhaltet diese Arbeit neue Ideen und Methoden zur Beurteilung und Verbesserung von maschinellen Lernverfahren, um die Anwendungsmöglichkeiten dieser Verfahren in der chemischen Forschung nachhaltig zu erweitern.

Acknowledgements

First of all, I would like to thank Prof. Dr. Klaus-Robert Müller for giving me the opportunity to work in his group. His infectious optimism and encouragement was of indispensable value in the course of my research. Prof. Dr. Gisbert Schneider supported me with his expertise in the area of chemoinformatics and I am very grateful to him and his group, including Dr. Petra Schneider, Tim Geppert and Felix Reisen, for introducing me to the experimental side of chemoinformatics.

This work would not have been possible without the contributions of my coworkers and colleagues. Dr. Timon Schroeter, who talked me into chemoinformatics, gave me inspiration and motivation for my research. I would like to thank Dr. Matthias Rupp for his thorough review of my work and his reliable assistance within the past year. A great thanks goes to David Baehrens, Fabian Rathke, Dr. Ulf Brefeld and Peter Vascovic for their help on designing and implementing new algorithms. I would like to thank my collaborators at Idalab, Bayer Schering Pharma and Boehringer Ingelheim, especially Dr. Sebastian Mika, Dr. Antonius ter Laak, Dr. Nikolaus Heinrich, Dr. Andreas Sutter and Dr. Jan Kriegl, for many fruitful discussions and the open exchange of experience. The participants of "Navigation Chemical Compound Space for Materials and Bio Design" program at IPAM, UCLA inspired my work and I would like to thank Zachary Pozun, John Snyder, Prof. Dr. Kieron Burke, Dr. Daniel Sheppard and Prof. Dr. Graeme Henkelman for their calculations and willingness to share their knowledge.

I deeply enjoyed working at the Machine Learning Group of the Berlin Institute of Technology and would like to thank all members for generating a kind and open-minded research atmosphere. A special thanks goes to my former office mates Dr. Tobias Lang, Stanimír Dragíev and Martijn Schreuder. Their humor and friendship was always a great support.

I gratefully acknowledge the funding from the German Research Foundation (MU 987/4-2), the German Academic Exchange Service and the European Commission under the PASCAL2 Network of Excellence of the FP7-ICT program.

Finally, I would like to thank my family and friends who, non-scientifically but to the same degree, supported my work. In particular I would like to express my gratitude to the group of women which originated from the ProMotion program, to Annika who fostered my interest in life science and, most of all, to Marcel for his patience and understanding.

Contents

Preface

Machine learning (ML) is a branch of artificial intelligence concerned with the development of cost-efficient computing systems with learning capability. Nowadays these systems can be found in automated mail sorting programs, voice-controlled writing applications, recommender system of online stores like Amazon, computer firewalls and many other parts of our everyday live. Especially data-rich areas of science like bioinformatics, image processing or neuroscience benefit from the automated knowledge acquisition of machine learning algorithms.
In chemistry, machine learning algorithms are predominantly employed in the area of drug discovery. Nonlinear machine learning methods like support vector machines, neural networks, random forests, and Gaussian processes can be used to predict ADMET (absorption, distribution, metabolism, excretion, and toxicity) and related physicochemical properties [89, 48, 29, 115, 113]. In drug discovery an early assessment of these properties is highly important in order to prioritize compounds thereby save costs, time and controversial animal experiments.
Due to the complexity of chemical systems, the vast space of chemical compounds and the limited amount of experimental data, however, current approaches fail to provide prediction models of general accuracy for complex chemical properties. Moreover, there exist no accurate confidence estimates for the single predictions of these systems. Thus, machine learning methods find only limited application in industrial research and can support, but not substitute laboratory experiments.

This thesis aims to extend the acceptance and usability of kernel-based machine learning methods in chemical research. Based on a close analysis of chemical problems and experimental data, we derive new algorithms and methods to enhance machine learning predictions and to asses their reliability.

Roadmap

CHAPTER 1 The first chapter gives a brief introduction to chemoinformatics, drug discovery and machine learning. The special requirements of machine learning methods arising in chemical applications are discussed and related to the author's contributions.

CHAPTER 2 In the second chapter we build on the framework of structured-learning and a metric used in information retrieval to derive a novel algorithm for virtual screening. In contrast to previous approaches this algorithm directly ranks compounds and focuses strongly on high binding affinities. The capability of the new algorithm StructRank is illustrated on three different screening datasets with different label distributions.

CHAPTER 3 This chapter is concerned with ensemble models and local correction approaches in the context of hERG inhibition. Moreover, a typical research scenario, where the compounds of interest are not directly related to the training data, is simulated in order to examine the differences between standard and the newly introduced clustered cross-validation framework.

The second part of this thesis investigates methods to assess the reliability and applicability of ML predictions:

CHAPTER 4 A visual approach to explaining kernel-based predictions is introduced. The most relevant training compounds are visualized along with the predicted value in order to allow for an intuitive understanding and interpretation of single predictions. A questionnaire study on Ames mutagenicity prediction is performed to illustrate how this approach can help to spot wrong labels, detect poor model applicability and discover important chemical characteristics of the training data.

CHAPTER 5 In the fifth chapter local gradients are used to measure the local importance of chemical features in nonlinear classification models. The relevance of local gradients for chemical prediction models is illustrated on Ames mutagenicity data. The results reveal that, in contrast to common feature importance measures, this new approach allows to detect local as well as global trends within chemical data.

CHAPTER 6 Support vector machines and local gradients are introduced to transition state theory. We demonstrate how these methods can improve the assessment of the transition surface and reaction rates.

Previously Published Work

Parts of this thesis have been discussed in previous journal publications (co)authored by the author of this thesis:

- Anton Schwaighofer, Timon Schroeter, Sebastian Mika, Katja Hansen, Antonius ter Laak, Philip Lienau, Andreas Reichel, Nikolaus Heinrich and Klaus-Robert Müller *A Probabilistic Approach to Classifying Metabolic Stability*. Journal of Chemical Information and Modelling, 48(4): 785-796, **2008**.
- Katja Hansen, Sebastian Mika, Timon Schroeter, Andreas Sutter, Antonius ter Laak, Thomas Steger-Hartmann, Nikolaus Heinrich and Klaus-Robert Müller. *Benchmark Data Set for in Silico Prediction of Ames mutagenicity*. Journal of Chemical Information and Modelling, 49(9): 2077-2081, **2009**. (Chapter 4 and 5)
- Katja Hansen, Fabian Rathke, Timon Schroeter, Georg Rast, Thomas Fox, Jan M. Kriegl and Sebastian Mika. *Bias-Correction of Regression Models: A Case Study on hERG Inhibition*. Journal of Chemical Information and Modelling, 49(6):1486-1496, **2009**. (Chapter 1 and 3)
- Matthias Rupp, Timon Schroeter, Ramona Steri, Heiko Zettl, Ewgenij Proschak, Katja Hansen, Oliver Rau, Oliver Schwarz, Lutz Müller-Kuhrt, Manfred Schubert-Zsilavecz, Klaus-Robert Müller and Gisbert Schneider. *From machine learning to natural product derivatives selectively activating transcription factor PPARγ* . ChemMedChem,Wiley, 5(2): 191-194, **2010**.
- Ramona Steri, Matthias Rupp, Ewgenij Proschak, Timon Schroeter, Heiko Zettl, Katja Hansen, Oliver Schwarz, Lutz Müller-Kuhrt, Klaus-Robert Müller, Gisbert Schneider and Manfred Schubert-Zsilavecz. *Truxillic acid derivatives act as peroxisome proliferator-activated receptor y activators*. Bioorganic & Medicinal Chemistry Letters, 20: 29202923 **2010**.
- David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen and Klaus-Robert Müller. *How to Explain Individual Classification Decisions*. Journal of Machine Learning Research, 11: 1803-1831, **2010**. (Chapter 5)
- Iurii Sushko, Sergii Novotarskyi, Robert Korner, Anil Kumar Pandey, Artem Cherkasov, Jiazhong Li, Paola Gramatica, Katja Hansen, Timon Schroeter, Klaus-Robert Müller, Lili Xi, Huanxiang Liu, Xiaojun Yao, Tomas Oberg, Farhad Hormozdiari, Phuong Dao, Cenk Sahinalp, Roberto Todeschini, Pavel Polishchuk, Anatoliy Artemenko, Victor Kuzmin, Todd M. Martin, Douglas M. Young, Denis Fourches, Eugene Muratov, Alexander Tropsha, Igor Baskin, Dragos Horvath, Gilles Marcou, Christophe Muller, Alexander Varnek, Volodymyr V. *Applicability domains for classification problems: Benchmarking of distance to models for Ames mutagenicity set*. Journal of Chemical Information and Modelling, 50(12): 2094-111, **2010**.
- Katja Hansen, David Baehrens, Timon Schroeter, Matthias Rupp and Klaus-Robert Müller. *Visual Interpretation of Kernel-Based Prediction Models*. Molecular Informatics, 30: 817-826 **2011**. (Chapter 4)
- Fabian Rathke, Katja Hansen, Ulf Brefeld and Klaus-Robert Müller. *StructRank: A New Approach for Ligand-Based virtual screening*. Journal of Chemical Information and Modelling, 51(1): 83-92 **2011**. (Chapter 2)
- Zachary D. Pozun, Katja Hansen, Daniel Sheppard, Matthias Rupp, Klaus-Robert Müller and Graeme Henkelman. *Optimizing Transition State Dividing Surfaces via Kernel-Based Machine Learning*, submitted. (Chapter 6)

- Grigorios Skolidis, Katja Hansen, Guido Sanguinetti and Matthias Rupp. *Multi-task learning for pK_a prediction*, submitted.
- John C. Snyder, Matthias Rupp, Katja Hansen, Klaus-Robert Müller and Kieron Burke. *Finding Density Functionals with Machine Learning*, submitted.

Chapter 1

Introduction

1.1 Machine Learning

The field of machine learning (ML) seeks to infer and generalize dependencies from data using computing systems with learning capability. ML is concerned with many research questions arising in the field of statistics, data mining and psychology, but with differences of emphasis. Statistics focuses on understanding the data generating process, often with the goal of testing hypotheses, whereas data mining seeks to find patterns in the data and psychological studies aspire to understand the mechanisms underlying human learning behaviors [26]. In contrast, the machine learning methods investigated in the following are primarily concerned with predictive modeling, where a model is built to mimic and generalize a certain characteristic of the data generating process. More precisely, the goal is to predict the label y for some test sample $\mathbf{x}$ by taking a given training set $\mathbf{D}$ into account.

Types of Learning

The training set $\mathbf{D}$ may consist of

- samples with associated labels (*supervised learning scenario*),
- solely of samples without labels (*unsupervised learning scenario*),
- labeled and unlabeled samples (*semi-supervised learning scenario*).

Within this thesis, we focus on supervised learning scenarios. The training sets usually consist of small molecules represented as vectors of chemical descriptors with associated experimental measurements of physical or chemical properties. The constitution of the measurement values y determines the type of learning task: Quantitative measurements result in *regression tasks*, whereas *classification tasks* are determined by qualitative measurements. The class of *ranking tasks* discussed in Chapter 2 forms an exception where the labels are continuous values as in regression tasks but the prediction is a ranking of compounds according to their unknown labels.

The Frequentist and the Bayesian Approach

A fundamental basis of machine learning is statistical inference, i.e. the process of drawing conclusions from observable data that are subject to random variation. There are two different views on inference in statistics resulting from different definitions of probability.

For a *frequentist* a statistical model equals a function $y = f(\mathbf{x}; \theta)$, where f represents a class of parametric functions with parameters θ processing input sample $\mathbf{x}$. The strategy for learning is based on determining the parameters θ such that f is optimal with respect to certain likelihood terms or loss functions.

From a *Bayesian* point of view the parameters θ_i are not fixed but distributed according to a known prior distribution $P(\theta)$. An initial model is designed based on this prior information and then adapted in light of the observed data. The model provides a representation of our prior knowledge about the system and the information derived from the data [8]. Thus the prediction takes the form of a predictive distribution $P(y|\mathbf{x})$ which can be interpreted as an expression of our degrees of belief in the various possible outcomes y [83].

Over the last decades a huge amount of research publications addressed machine learning and respective theory, building on the work of Vapnik [138] and Rosenblatt [100]. The Appendix provides an overview of the main concepts and ideas of the machine learning methods implemented in this thesis. For a comprehensive introduction into the field of machine learning we refer to the literature [46, 8, 81, 28] for further reading.

1.2 Chemoinformatics and the Drug Discovery Process

Chemoinformatics

Chemoinformatics (also cheminformatics) is a young research area at the interface of chemistry and informatics [148, 36, 13] that employs mathematical and statistical methods to extract information from chemical data [35]. In the 1970's chemists started to discover computers as tools for chemical research. From the very beginning, however, it could be observed that there was a split between theoretical chemists using computers for quantum mechanical calculations and chemists using computers for information processing and data analysis [36]. The first group of theoretical chemists founded the research area of computational chemistry, while the latter group denominated their research as chemoinformatics. The two fields, however, are highly related and the differences only vaguely defined.

In chemoinformatics machine learning methods are predominantly applied to problems arising in the context of drug discovery and design. However, as exemplified in Chapter 6, the field of theoretical chemistry provides opportunities for machine learning as well.

Drug Discovery Process

The drug discovery and development process of a single new drug encompasses between ten and fifteen years of research and costs about 1.8 billion dollars (Paul et al. [91]).
The first step in drug discovery comprises the selection and confirmation of a *target*, a single molecule, often a protein, which is involved in the disease mechanism and can potentially interact with a drug molecule to affect the disease. In the following screening step one seeks to find *hits*, small molecules that interact with the target. The most common approach is high-throughput screening (HTS) where screening robots are used to perform a chemical assay on millions of compounds. Alternatively, virtual screening or molecular modeling techniques are applied. Here, computational methods (*in silico methods*) are implemented to assess the potential of compounds.
After the effect of the identified hits on the target has been confirmed in laboratory experiments, the first safety tests are performed on the most promising compounds (hit-to-lead phase). ADMET (absorption, distribution, metabolism, excretion, and toxicity) and related

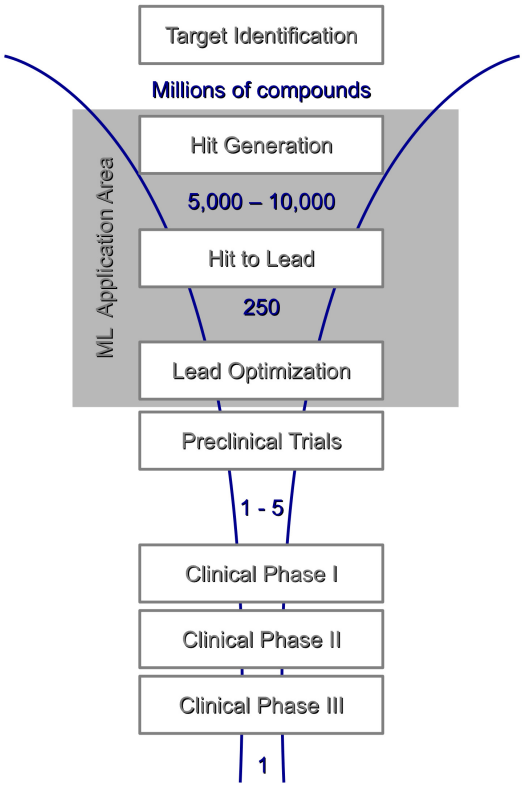


Figure 1.1: Sketch of the drug discovery process.

properties are investigated in order to exclude dangerous side effects. The most promising compounds are then determined as *leads*.
During lead optimization these lead structures are modified to improve efficacy and enhance their ADMET profile. A successful drug needs to be absorbed into the bloodstream, distributed to the proper organs, metabolized efficiently and effectively and then be excreted from the body without any toxic side effects. The mutual optimization of these highly related drug properties is the key challenge in lead optimization. Finally, about 2% of the originally identified hits enter the preclinical trials where laboratory and animal experiments are performed to determine if the compound is safe enough for human testing.
The subsequent phases of clinical trials are denoted as the drug *development* process. Starting from groups of 20 to 100 healthy volunteers (phase 1) over small groups of patients (phase 2) to large studies including 1,000 to 5,000 of patients (phase 3) the drug candidate is extensively tested to determine risks, efficacy and dosing. A successful drug candidate is then approved as new drug and enters the market.

A key problem of todays pharmaceutical industry is the productivity of this drug development process [91]. While the number of newly approved innovative drugs is decreasing the loss of revenues due to patent expirations for successful products and rising development costs are limiting research activities [39]. Following the "fail early—fail cheap" paradigm, companies now try to consider ADMET properties (using e.g. ML methods) as early as possible in the drug development process.

1.3 Machine Learning in Drug Discovery

Different areas within drug discovery benefited from utilizing machine learning technologies (see Wale [142] for a review). Among them are:

- Virtual screening: In virtual screening machine learning techniques are applied to rank or filter compounds with respect to different properties (Melville et al. [79] reviews applications in ligand-based and structure-based virtual screening). Especially in ligand-based virtual screening, where no structural information about the target is available, machine learning methods enhance similarity search—even if only very few reference compounds are given [50]. Additionally, ML methods may be applied to create diverse compound libraries that can serve as input for virtual screening (library design) [105].
- Quantitative structure-activity relationship (QSAR) and quantitative structure-property relationship (QSPR): QSAR and QSPR models are statistical models used to infer dependencies between chemical structures and their biological activity or physicochemical properties. Within the last decades machine learning models like neural networks or support vector machines became popular in this area of research.
- Prediction of protein structure, function and interaction: Machine learning methods have found extensive applications in biochemical tasks like protein structure prediction, protein function prediction and characterization of protein-protein interaction. Though these problems are related to drug discovery, we will subsequently restrict ourselves solely to machine learning applications on drug-like compounds. However, most of the challenges discussed in the following occur in all three areas of application.

Challenges of ML in Drug Discovery

In order to generate an operational prediction model for chemical applications multiple challenges of chemical data have to be addressed, mainly:

Representation of chemical structures
Chemical compounds are flexible three-dimensional structures that change shape and conformation as they interact with the environment. In order to apply statistical methods, the compounds are represented as vectors of molecular descriptors. These numbers reflect shape, charge, connectivity, weight and various other properties derived from the 2D and/or 3D structure of the molecules [131]. Some methods also allow to work directly on the 2D or 3D graph structures by using, e.g., graph kernels [101]. Unfortunately, both approaches are not able to capture the flexibility of molecules and the special characteristics of chemical compounds exhibiting several graph structures (tautomers) or several 3D structures (conformers) are not considered.

Constitution of empirical measurements:
A collection of empirical data like laboratory measurements is exposed to different sources of error: inaccurate labeling, systematic measurement errors and the inherent error of the measuring system. Thus, each modeling approach requires a thorough pre-processing including outlier analysis, visual data inspection and if necessary normalization.

Amount and distribution of data:
In early stages of the drug discovery process there is little knowledge about the target and the available datasets are small and of low diversity. Prediction models build on this kind of data are prone to overfitting and show a low generalization capacity.

Though the dataset grows stepwise with the ongoing development, the newly investigated compounds commonly lie beyond previously examined series of compounds. Thus, prediction results stay inaccurate due to missing information in the chemical space of interest.

Complexity of chemical interactions:
Machine learning in drug discovery relies on the assumption that similar molecules exhibit similar activity. This implies that the activity value changes continuously over chemical space and can be pictured as a smooth surface. Unfortunately, very similar compounds may in some cases possess extremely different activities leading to rugged canyons called "activity cliffs" in the activity landscape [76]. The detection and modeling of such activity cliffs is a problem of ongoing research.

On the one hand it is highly desirable to strive for a statistical model meeting all these challenges (and a lot of ongoing research addresses these problems). On the other hand one has to keep in mind that it is unrealistic to find such a model. Chemists and physicists are still discovering new chemical phenomena and modes of molecular interaction. Our current chemical knowledge is somehow incomplete and we can not expect a perfect model on the basis of incomplete information.

Thus, the question arises how to deal with the imperfection of data in chemoinformatics and the resulting inaccuracy of prediction models. In the following chapters this problem is addressed from an use-oriented point of view.

1.3.1 Thesis Scope and Contributions

In the first part of this thesis we analyze how one can incorporate the limited amount of data in a (kernel-based) prediction model such that it optimally fits the conditions and requirements of virtual screening and lead optimization.

Virtual screening aims to rank molecules in terms of their binding coefficients for the investigated drug target, such that the top-k molecules can be selected for further investigation. Thus, we derive a *new screening algorithm* StructRank which directly predicts a ranking for a given set of compounds and allows to focus on high binding affinities (Chapter 2). In contrast to other ligand-based screening algorithms the new approach is based on the relative binding affinity and makes better use of the information encoded in the training data if only few highly binding compounds are available.

In lead optimization and hit-to-lead optimization the exact prediction of chemical properties related to unfavorable side-effects is required. With every new batch of experiments more training data becomes available but the requested predictions commonly concern new molecules beyond this dataset. To estimate the prediction accuracy in such an application scenario we present a *clustered cross-validation* scheme (Section 3.4.1 and A.3) and compare it to standard cross-validation. Moreover, we show how the newly received data can be beneficially incorporate using *local bias correction* (Chapter 3). On the basis of hERG inhibition data this method is evaluated and compared to ensemble learning approaches.

The introduced methods significantly improve prediction models by extracting the most relevant information of a given (incomplete) dataset with respect to a certain application in drug discovery. Nevertheless, they can not reach perfect accuracy and a validity measure for single predictions is needed in order to to prioritize compounds correctly within the research process.

Thus the second part of this thesis is dedicated to the interpretability, knowledge of the do-

main of applicability[1], and estimation of confidence in machine learning based predictions. In Chapter 4 we develop and validate a *method for the interpretation* of kernel-based prediction models. The most influential training compounds are identified and visualized for individual predictions as foundation of interpretation. As a consequence of this interpretability, the method helps to assess regions of insufficient coverage or activity cliffs, to spot wrong labeling, and to determine relevant molecular features as illustrated on Ames mutagenicity data.

Subsequently, *local gradients* are introduced as a tool for interpretation in terms of chemical features (Chapter 5). They facilitate the understanding of prediction models by indicating the chemical input features with the greatest impact on the predicted value. Given a well-founded prediction model the local gradients allow to deduce global and local structure elements of the chemical space and thereby facilitate compound optimization. The framework presented and evaluated in Chapter 5 allows to calculate such local gradients for any classification model.

Finally, we illustrate the utility of the developed methods beyond drug discovery and design. Transition state theory is known as a semi-classical approach to assess the transition surface and reaction rate within the field of theoretical chemistry. In Chapter 6 support vector machines and local gradients are for the first time applied to enhance the sampling of the potential energy surface. The *new sampling approach* accelerates the assessment of the transition surface and improves the resulting reaction rate.

[1]The domain of applicability of a model refers to the part of the chemical space where the model provides reliable predictions.

Chapter 2

A Ranking Approach for Ligand-Based Virtual Screening

2.1 Introduction

Screening large libraries of chemical compounds against a biological target, typically a receptor or an enzyme, is a crucial step in the process of drug discovery.

Besides high-throughput screening, the physical screening of large libraries of chemicals, computational methods, known as virtual screening (VS), [144] gained attention within the last two decades and were applied successfully as an alternative and complementary screening tool [93, 102].

The task in VS, also known as "early recognition problem" [136, 84], can be characterized as follows: Given a library of molecules, the task is to output a ranking of these molecules in terms of their binding coefficient for the investigated drug target, such that the top k molecules can be selected for further investigations. As a standard, current quantitative structure-activity relationship (QSAR) regression models are applied to predict the level of activity: They learn a function $f : x \mapsto y,\ f : \mathbb{R}^d \to \mathbb{R}$ that predicts a label for any molecule given its features. To establish the subset of candidate molecules, predictions are made for all molecules in the database. In a second step an ordered list is generated based on these predictions. This two step approach is shown in Figure 2.1 (top). Finally the top k ranked compounds are selected to be investigated in more detail.

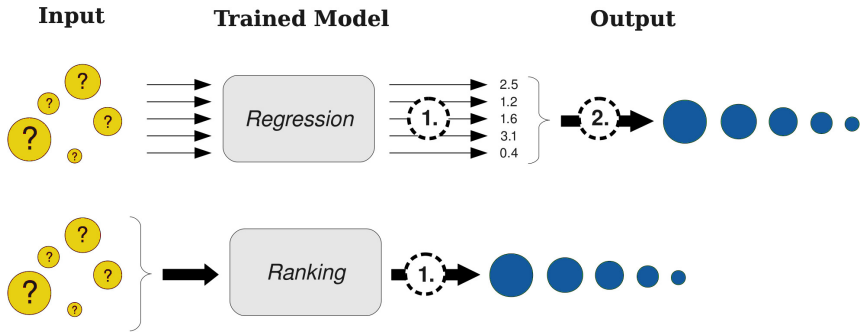


Figure 2.1: Two different ways to solve the ranking task of virtual screening: a) State-of-the-art approaches use a 2-step approach. In the first step a regression model is used to predict binding coefficients for all molecules in the library. In a second step the molecules are sorted according to their predictions. b) The new ranking approach directly predicts the ranking within a single step.

However, virtual screening approaches primarily aim to find molecules exhibiting high binding affinities with the target while the predictive accuracy with respect to the labels y is only of secondary interest. Although a perfect regression model would also imply a perfect ranking of the molecules of interest, the impact of suboptimal regressors on the ranking is not easily captured as equal models in terms of their mean squared error could give rise to completely different rankings. Thus, the question rises whether the detour via regression is necessary and whether the task can be addressed in a more natural way. In this chapter, a top k ranking algorithm, **StructRank**, that *directly solves the ranking problem* and that *focuses on the most promising molecules* (cf. 2.1, bottom) is derived and evaluated on three virtual screening datasets.

2.2 Methods

The formal problem setting of ranking for virtual screening is as follows: Let $\{(\mathbf{x}_i, y_i)_{i=1}^n\}$ be a given set of n molecules, where $\mathbf{x}_i \in \mathbb{R}^d$ denotes the feature vector of the i-th molecule containing the molecular descriptors, and $y_i \in \mathbb{R}$ is a scalar representing the biological/chemical property of that molecule, e.g. binding affinity.

Based on this set we aim at learning a function $f : \mathcal{X} \to \mathcal{P}$ that takes any set of molecules $\tilde{\mathbf{x}} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\} \in \mathcal{X}$ and returns a ranking $\mathbf{p} \in \mathcal{P}^1$ of these molecules according to the biological/chemical property of interest. Moreover, as the purpose of virtual screening methods is to rank actives *early* in an ordered list (see "early recognition problem" [136, 84]), we want the learning machine to focus on the top k molecules in the ranking.

In the following we derive a top k ranking SVM meeting these requirements for QSAR. The approach builds on work by Chapelle et al. [20] and Tsochantaridis [137].

2.2.1 Evaluate Rankings Using NDCG

The definition of an adequate quality measure for rankings of molecules is of crucial importance in the development of a ranking algorithm suitable for virtual screening. We propose to use a popular ranking measure that originates from the information retrieval community: Normalized Discounted Cumulative Gain (NDCG). Given the true ranking $\bar{\mathbf{p}}$, a predicted ranking $\hat{\mathbf{p}}$ and a cut-off k, NDCG is given by the DCG (Discounted Cumulative Gain) for the predicted ranking normalized by the DCG of the true ranking:

$$NDCG_k(\bar{\mathbf{p}}, \hat{\mathbf{p}}) = \frac{DCG_k(\hat{\mathbf{p}})}{DCG_k(\bar{\mathbf{p}})}, \qquad DCG_k(\mathbf{p}) = \sum_{r=1}^{k} \frac{2^{\mathbf{p}[\mathbf{y}]_r} - 1}{\log_2(1+r)} \tag{2.1}$$

where $\mathbf{p}[\mathbf{y}]_r$ is the binding coefficient y_i of the molecule $\mathbf{x}_i$ ranked at position r.

Originally, NDCG [56] was introduced to evaluate the results of web searches. It measures how similar a predicted ranking is compared to the true ranking. NDCG has several important properties:

- NDCG_k only evaluates the first k positions of predicted rankings, thus an error on positions below rank k is not punished.
- Furthermore the first k positions are weighted, which means that errors have different influence on the final score depending on which position of the ranking they occur.

[1] In the following a ranking is described as a permutation $\mathbf{p}$, i.e. for a given set of molecules $\tilde{\mathbf{x}}$ and a vector $\mathbf{y}$ of corresponding binding coefficients, $\mathbf{p}[\mathbf{y}]$ gives ideally the vector of binding coefficients in decreasing order.

Naturally position one is the most important, with lower positions discounted by the log of their rank r: $\log_2(1+r)$.

- Finally, NDCG is normalized, thus if the predicted ranking equals the true ranking the score is 1. To translate NDCG into a loss function we simply use $\Delta(\mathbf{p}, \hat{\mathbf{p}}) = 1 - NDCG(\mathbf{p}, \hat{\mathbf{p}})$.

In summary, NDCG aims at pushing the molecules with the highest binding affinity on top of the ranking.

2.2.2 Structured Support Vector Machines for QSAR

Let us reconsider the ultimate target of learning a function $f : \mathcal{X} \to \mathcal{P}$ that maps a set of molecules onto a ranking. In order to establish f, we utilize the basic concepts of Structured SVMs (see Tsochantaridis et al. [137]), a very flexible learning machine that has been applied to many different learning tasks in information retrieval [20, 152], natural language parsing [10], and protein sequence alignment [151]. Structured SVMs learn a discriminant function $F : \mathcal{X} \times \mathcal{P} \to \mathbb{R}$. F can be thought of as a *compatibility* function, that measures how well a certain ranking $\mathbf{p}$ fits the given set of molecules $\tilde{\mathbf{x}}$. The final prediction is given by the ranking $\mathbf{p}$ that achieves the maximal score $F(\tilde{\mathbf{x}}, \mathbf{p})$. Thus we have

$$f(\tilde{\mathbf{x}}) \quad = \quad \operatorname*{argmax}_{\mathbf{p} \in \mathcal{P}} F(\tilde{\mathbf{x}}, \mathbf{p}).$$

F is defined over a combined space of sets of molecules and corresponding rankings, a so called "joint feature space". To be able to learn F directly in that combined space, we define a function Ψ that maps each pair of a set of molecules $\tilde{\mathbf{x}}$ together with a ranking $\mathbf{p}$ (of $\tilde{\mathbf{x}}$) onto one corresponding data point in the joint feature space

$$\Psi(\tilde{\mathbf{x}}, \mathbf{p}) = \sum_{i=1}^{n} \phi(\tilde{\mathbf{x}}_i) A(\mathbf{p}_i) \tag{2.2}$$

where the function ϕ is a mapping into a Hilbert space corresponding to a kernel function $k(\mathbf{x_i}, \mathbf{x_j})$ and $A(r) = \max(0, k+1-r)$ weights the molecules according to their ranks as proposed by Chapelle [20]. Only molecules corresponding to the first k ranks are incorporated.

Given the joint feature map Ψ, F is defined as a linear function in the joint feature space:

$$F(\tilde{\mathbf{x}}, \mathbf{p}) \quad = \quad \mathbf{w}^T \Psi(\tilde{\mathbf{x}}, \mathbf{p}),$$

F is the scalar product of the corresponding joint feature map of $\tilde{\mathbf{x}}$ given a particular ranking $\mathbf{p}$ and the learned parameter vector $\mathbf{w}$.

Modeling F can be casted as follows: Given a set of molecules $\tilde{\mathbf{x}}$ we want the true ranking $\bar{\mathbf{p}}$ to score highest among all possible rankings $\mathbf{p} \in \mathcal{P}$ transforming into constraints

$$\mathbf{w}^T(\Psi(\tilde{\mathbf{x}}, \bar{\mathbf{p}}) - \Psi(\tilde{\mathbf{x}}, \mathbf{p})) \geq 0 \quad \forall\, \mathbf{p} \in \mathcal{P} \setminus \bar{\mathbf{p}}.$$

Alike classic SVMs for classification [138] this can be turned into a maximum-margin problem, where we want the difference between the true ranking $\bar{\mathbf{p}}$ and the closest runner-up $\operatorname{argmax}_{\mathbf{p} \neq \bar{\mathbf{p}}} \mathbf{w}^T \Psi(\tilde{\mathbf{x}}, \mathbf{p})$ to be maximal (cf Section A.2, support vector classification). Also we want different $\mathbf{p}$'s to get separated according to the degree of their falseness: A predicted ranking with only two ranks interchanged compared to the true ranking is much better than a predicted ranking with all ranks interchanged. We thus require the latter to get further separated with a larger margin from the true ranking than the first one. This is accomplished by

replacing the constant margin formulation with the loss-dependent margin (*margin scaling* [137, 127]):

$$\mathbf{w}^T(\Psi(\tilde{\mathbf{x}}, \bar{\mathbf{p}}) - \Psi(\tilde{\mathbf{x}}, \mathbf{p})) \geq \Delta(\mathbf{p}, \bar{\mathbf{p}}) \quad \forall\, \mathbf{p} \in \mathcal{P} \setminus \bar{\mathbf{p}} \tag{2.3}$$

where 1-NDCG$_k$ is used for $\Delta(\mathbf{p}, \bar{\mathbf{p}})$. Furthermore a *slack variable* ξ is introduced that reflects the maximal error made for the set of constraints (see eq.(2.3)). Finally, to improve performance, we employ a boosting approach: We randomly draw m different subsets $\tilde{\mathbf{x}}^j$ of molecules from the training set. Applying the methodology described so far to each subset j we obtain the final optimization problem

$$\begin{aligned} \min_{w,\xi} \quad & \frac{1}{2}\,\mathbf{w}^T\mathbf{w} + C\sum_{j=1}^{m}\xi^j \\ \text{subject to} \quad & \mathbf{w}^T(\Psi(\tilde{\mathbf{x}}^j, \bar{\mathbf{p}}^j) - \Psi(\tilde{\mathbf{x}}^j, \mathbf{p})) \geq \Delta(\bar{\mathbf{p}}^j, \mathbf{p}) - \xi^j \quad \forall j, \forall \mathbf{p} \neq \bar{\mathbf{p}}^j \\ & \xi^j \geq 0. \end{aligned} \tag{2.4}$$

The corresponding dual is given by

$$\begin{aligned} \max_{\alpha} \quad & -\frac{1}{2}\,\boldsymbol{\alpha}^T L\,\boldsymbol{\alpha} + \mathbf{b}^T\boldsymbol{\alpha} \\ \text{subject to} \quad & \sum_{\mathbf{p}\in\mathcal{P}} \alpha^j_{\mathbf{p}} \leq C, \quad \alpha^j_{\mathbf{p}} \geq 0 \quad \forall j, \forall \mathbf{p} \neq \bar{\mathbf{p}}^j \end{aligned} \tag{2.5}$$

where we have an α for each possible ranking $\mathbf{p}$ of subset $\tilde{\mathbf{x}}^j$. The matrix L consists of entries $(L)_{i\mathbf{p},j\mathbf{p}'} = (\Psi(\tilde{\mathbf{x}}^i, \bar{\mathbf{p}}^i) - \Psi(\tilde{\mathbf{x}}^i, \mathbf{p}))^T(\Psi(\tilde{\mathbf{x}}^j, \bar{\mathbf{p}}^j) - \Psi(\tilde{\mathbf{x}}^j, \mathbf{p}'))$ and $b_{i\mathbf{p}} = \Delta(\bar{\mathbf{p}}^i, \mathbf{p})$.

Note that there is a very high formal similarity to the original SVM formalization (compare eq. (2.4) and eq. (A.14) in the introduction) with the differences: (a) margin rescaling, (b) joint feature map and (c) very large quantity of constraints. A conceptual comparison of this StructRank and other SVM approaches is visualized in Figure 2.2.

For a set $\tilde{\mathbf{x}}$ with n molecules, there exist $n!$ possible ways of ranking these molecules. Imposing a constraint for each possible ranking would lead to problems becoming too big for being solved. Therefore, Tsochantaridis et al. [137] proposed a cutting plane approach that iteratively adds new constraints which violate the current solution. They show that there exists a polynomially sized subset of constraints whose solution fulfills all constraints of the full optimization problem. Astonishingly, the optimization problem can be solved efficiently, an example is the cutting-plane approach used in our implementation of StructRank.

2.2.3 Baseline Models

The novel ranking approach is compared to two algorithms both belonging to the family of support vector machines: support vector regression (SVR), a state-of-the-art regression method, often used for virtual screening and ranking SVM (RankSVM), another ranking approach.

Support Vector Regression (SVR)

Support vector regression [27] is an adaption of classic support vector classifiers for regression. Like their classification counterpart they follow the Structural Risk Minimization principle introduced by Vapnik [138], finding a trade-off between model complexity and training

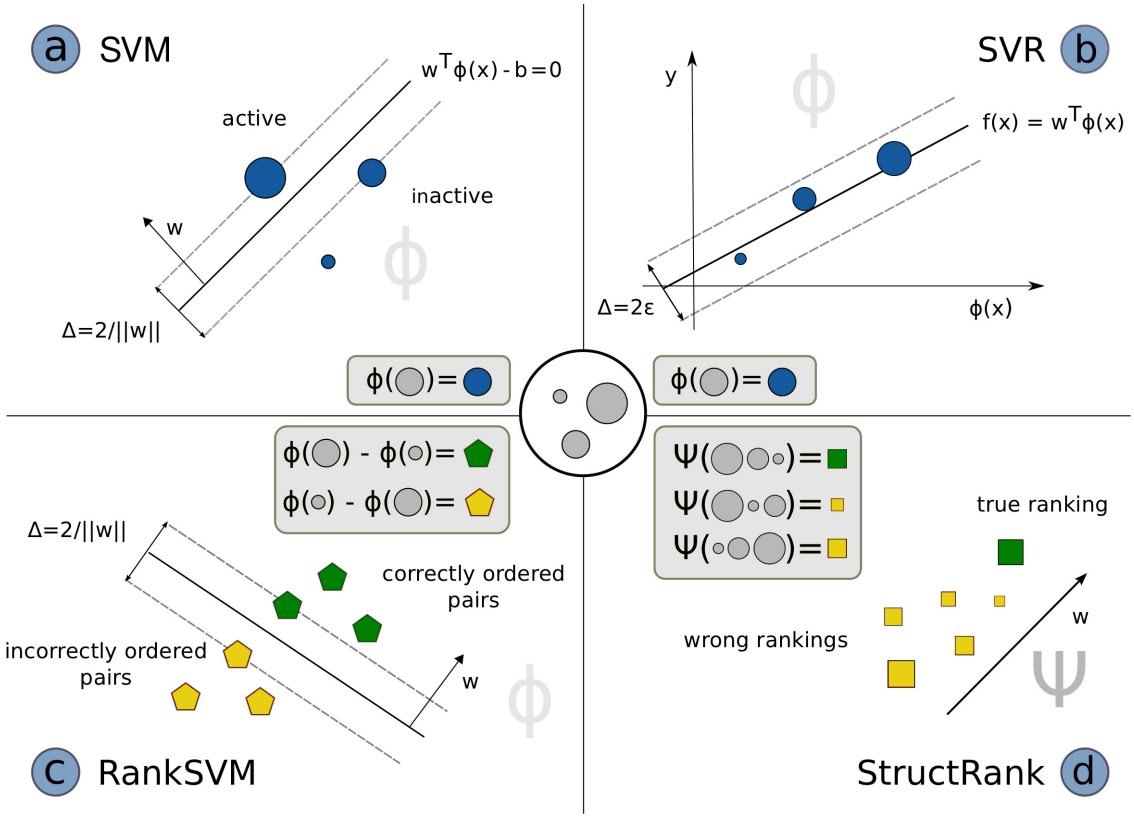


Figure 2.2: Comparison of different support vector machines: **a)** Support vector machines for classification learn a linear hyperplane $\mathbf{w}^T\phi(\mathbf{x}) = b$ with maximum margin Δ that optimally separates active from inactive molecules. **b)** Support vector regression learns a function $\mathbf{w}^T\phi(x)$ that predicts binding affinities for each molecule as correct as possible. **c)** Ranking SVM generates difference vectors of all possible pairs of molecules. Afterwards similar to a) a linear hyperplane is learned that separates correctly and incorrectly ordered pairs. **d)** Ψ takes a set of molecules $\tilde{\mathbf{x}}$ and a ranking $\mathbf{p}$ of this set and maps it onto a point in the joint feature space. StructRank learns a function $\mathbf{w}^T\Psi(\tilde{\mathbf{x}}, \mathbf{p})$ which assigns the highest score to the point representing the true ranking.

error. SVRs learn a linear function f in some chosen kernel feature space [108]. The final predictor is given by

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i k(\mathbf{x}_i, \mathbf{x}) + b. \tag{2.6}$$

The α's weight the influence of training points $\mathbf{x}_i$ on the prediction $f(\mathbf{x})$. An ϵ-sensitive loss function is minimized, penalizing only predictions $\hat{y} = f(\mathbf{x})$ that differ more than ϵ from the true label y:

$$\ell(y, \hat{y}) = |(y - \hat{y})|_\epsilon = \begin{cases} |(y - \hat{y})| & \text{for} \quad |(y - \hat{y})| > \epsilon \\ 0 & \text{else.} \end{cases} \tag{2.7}$$

See Figure 2.2b) for a visualization of SVR. Different studies [24, 74, 150, 67] showed that SVRs can outperform multiple linear regression and partial least squares and perform on par with neuronal networks. As implementation LIBSVM together with a Matlab interface available from `http://www.csie.ntu.edu.tw/~cjlin/libsvm/` (currently in version 3.0, 03.11.2010) is used.

Ranking SVM

As a second baseline we consider a second ranking approach: Ranking SVM [49, 59]. Falling into the category of *pairwise* ranking approaches, it maximizes the performance measure *Kendall's* τ. It measures the number of correctly ordered pairs within a ranking of length n, taking into account all possible $\frac{n(n-1)}{2}$ pairs. *Kendall's* τ has two crucial differences compared to NDCG: All positions of the ranking have an influence on the final performance unlike for NDCG, where only the top k positions matter. Additionally all positions have the same weight, unlike for NDCG, where higher positions are more important. The principle of Ranking SVM is visualized in Figure 2.2c). In this study the implementation of Chapelle (available from `http://olivier.chapelle.cc/primal/ranksvm.m`, accessed on the 03.11.2010) was extended for the use of kernels, according to [19].

2.2.4 Further Approaches and Alternatives

In virtual screening studies a model, generated based on a small number of labeled molecules, is used to screen large libraries of compounds. These libraries can be considered as unlabeled data which can be integrated into model generation using semi-supervised learning approaches. The framework of structured SVMs offers the possibility to integrate semi-supervised techniques like co-learning [9, 10]. Since this study is directed to the idea of ranking in virtual screening the aspect of semi-supervised techniques is not discussed and remains a promising direction of further investigations.

The structure-based ranking SVM with the NDCG loss presented in this work is only one possible approach to meet the requirements of virtual screening. An alternative approach to put more emphasis on compounds with a high binding affinity is based on the standard SVR model and simply re-weights the molecules within the SVR loss function according to their binding affinity. This leads to a stronger penalization of prediction errors for highly active molecules compared to molecules offering a low binding affinity. However, experiments based on this approach led to no performance gain.

2.3 Data

Sutherland et al. [125] tested spline-fitting together with a genetic algorithm to establish a good classifier on five virtual screening datasets. Out of these datasets a subset of three datasets most suitable for regression was selected: The benzodiazepine receptor (BZR), the enzymes cyclooxygenase-2 (COX-2) and dihydrofolate reductase (DHFR). All datasets were assembled from literature in order to mimic realistic HTS, i.e. possess high diversity and a low number of actives. Additionally almost all molecules can be considered drug-like satisfying Lipinski's rule of five [73]. Compounds with inexact measurements ($pIC50 < value$ instead of $pIC50 = value$) which are not suitable for regression approaches were removed from the original dataset. Table 2.1 summarizes the resulting datasets. A brief description of the biological function of each target is given below.

Table 2.1: Virtual Screening Datasets

Endpoint	Original Source	Exact Measurements[a]	pIC_{50} Range
BZR	405 molecules measured by Haefely et al. and Cook et al.	340	4.27 – 9.47
COX-2	467 molecules measured by Khanna et al.	414	4 – 9
DHFR	756 molecules measured by Queener et al.	682	3.03 – 10.45

[a] inexact measurements ($pIC50 < value$ instead of $pIC50 = value$) are excluded from the study

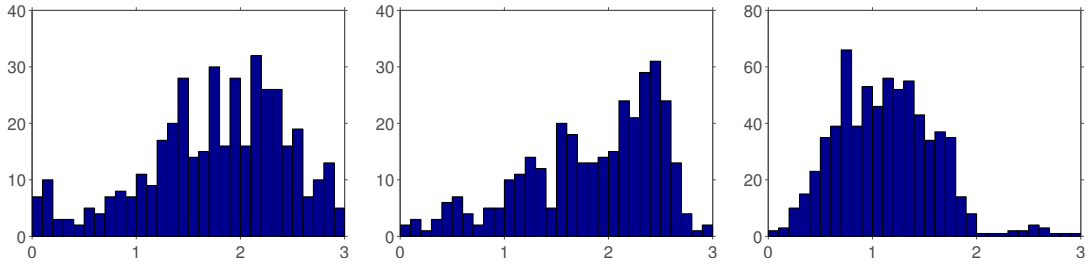
BZR The benzodiazepine receptor (BZR) or $GABA_A$ receptor is an ion channel located in the membrane of various neurons. The opening of BZR induced by its endogenous ligand GABA causes a membrane hyper polarization which increases the firing threshold. Drugs like benzodiazepine can bind in addition to GABA in their own allosteric binding site. They increase the frequency of channel opening thereby amplifying the inhibitory effect of GABA [118].

COX-2 The enzyme dyclooxygenase 2 (COX-2) together with it's isoform COX-1 [149] takes part in the synthesis of prostanoids. While COX-2 is an adaptive enzyme which is only produced in response to injury or inflammation, COX-1 is a constitutive enzyme which is produced constantly and provides for a physiological level of prostaglandins [25]. Unspecific COX inhibitors like aspirin produce gastrointestinal side-effects while specific COX-2 inhibitors were shown to reduce gastrointestinal side-effects at the price of increased cardiovascular risk [57].

DHFR The enzyme dihydrofolate reductase (DHFR) is involved in the syntheses of purins (adenine and guanine), pyrimidins (thymine) and some amino acids like glycine. As rapidly dividing cells like cancer cells need high amounts of thymine for DNA synthesis they are particularly vulnerable to the inhibition of DHFR. Methotrexat, for example, is a DHFR-inhibitor which is used in treatment amongst others of childhood leukemia and breast cancer [7].

2.3.1 Descriptor Generation and Data Preparation

As in previous studies [111, 112] a subset of Dragon blocks (1, 2, 6, 9, 12, 15, 16, 17, 18, and 20 generated by Dragon version 5.5.) is used to represent the molecules. This yielded 728–772



(a) COX-2 (“dense”) (b) BZR (“medium”) (c) DHFR (“sparse”)

Figure 2.3: The distribution of binding coefficients for the virtual screening datasets. The x-axis shows the binding coefficients (scaled into the range [0,3] for each dataset). The y-axis shows the number of molecules having that certain binding coefficient. Depending on the number of molecules with very high binding coefficients we refer to them as “dense” (COX-2), “medium” (BZR) and “sparse” (DHFR).

descriptors, depending on the dataset. The feature vectors are *normalized* to zero mean and unit variance on the training set. In order to keep the results between datasets comparable the binding coefficients are *scaled* for each dataset into the range $[0, 3]$ as this is a typical range when NDCG is used as scoring function for information retrieval datasets [56].

If we examine the distribution of binding coefficients for each dataset (see Figure 2.3), we can distinguish different types of distributions: For COX-2 we observe a high number of molecules with high binding coefficients, thus this dataset is called “dense” in the following. DHFR on the other hand has only a low number number of molecules with high binding coefficients, thus this dataset is denoted as “sparse”. BZR is in between with few molecules possessing very high binding coefficients (“medium”). We will make use of this distinction later in the result section.

2.3.2 Test Framework

A k-fold cross-validation is implemented to assess performance for the virtual screening datasets. In order to have similar training set sizes (about 225 molecules), the number of folds is varied for each dataset: BZR is split into three and COX-2 into two folds. Each of these folds is once used as test set, whereas the remaining two folds (fold) are used for training. Then an inner cross-validation with five folds is applied on the training set to determine the optimal hyperparameters.
For DHFR three folds are employed in the outer cross-validation loop but the single folds are used for training and the other two form the test set, thus also getting about 225 molecules in the training set. These cross-validations were performed seven times for DHFR and BZR, and ten times for COX-2.

As all three approaches share the same underlying SVM framework, they need to determine the same parameters within the inner cross-validation loop; for the RBF kernel

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{(\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j)}{2d\sigma^2}\right). \tag{2.8}$$

the parameters are $\sigma^2 \in \{0.1, 1, 10\}$ and d given by the number of descriptors. The SVM-parameter C controlling the model complexity is chosen from the set $\{0.01, 0.1, 1, 10, 100\}$. For the SVR the tube width is varied between $\{0.01, 0.1, 1\}$. For the StructRank approach 10, 10 and 30 ranks are considered during optimization.

2.3.3 Alternative Performance Measures

Besides NDCG two performance measures well known in the virtual screening community are evaluated: Enrichment Factor (EF) [92] and Robust Initial Enhancement (RIE) [116]. As shown by Truchon et. al. [136], the area under the ROC Curve is not suitable for the "early recognition problem" of virtual screening.

RIE and ER only distinguish between active and inactive molecules, contrary to NDCG, which takes precise binding affinities into account. To separate molecules into actives and inactives we chose 8.5 pIC_{50} (BZR), 8.0 pIC_{50} (COX-2) and 7.5 pIC_{50} (DHFR) resp as activity thresholds. The resulting datasets provide for challenging ranking problems with 60, 70 and 38 actives (BZR, COX-2 and DHFR).

The *Enrichment Factor* measures how many more actives are found in a defined fraction ζ of the ordered list, relative to a random distribution. Like NDCG it only takes the top k positions of the ranking into account, but weights each position equally. The Enrichment Factor is defined as

$$EF = \frac{\sum_{i=1}^{n} \delta_i}{\zeta \cdot n} \tag{2.9}$$

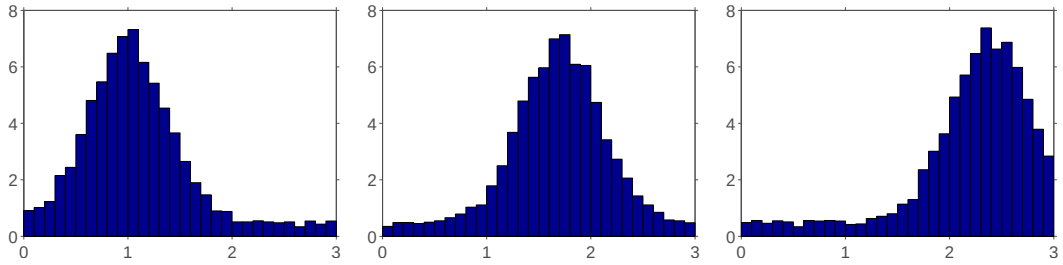
where n is the number of actives; δ_i is 1 if the active is ranked within the defined fraction of the list and 0 otherwise. *Robust Initial Enhancement* measures how much better a given ranking of actives is compared to their random distribution within the ranking. It considers the complete ranking, but like NDCG weights positions descending (depending on the parameter α, see eq. (2.10)). It is given by

$$RIE = \frac{\sum_{i=1}^{n} \mathrm{e}^{-\alpha r_i}}{\langle \sum_{i=1}^{n} \mathrm{e}^{-\alpha r_i} \rangle_r} \tag{2.10}$$

where r_i is the relative rank (i.e. the rank divided by the length of the ranking), and $1/\alpha$ is the fraction of the list that is most important for the final score, which has a similar meaning as the cutoff k of $NDCG$. The denominator is the mean score when the actives are distributed randomly across the ordered list.

2.3.4 Toy Example

Before analyzing real-world VS data we consider toy examples that reproduce a set of different label distributions typically found in virtual screening datasets: Datasets which possess only a low number of molecules with high binding affinities, and those which contain a medium or high number of those molecules. 300 training sets (100 of each type) with distribution of labels as outlined above were generated. Each training set consisted of 75 examples. Figure 2.4 shows the histograms, each averaged over all 100 sets. The aim is to compare the influence of the different label distributions on ranking performance. Thus validation and test sets were drawn with uniform label distributions for all three types of training sets (models were trained on the training set for different parameter combinations and the validation set is used to select the optimal parameter combination). Using the resulting model, ranking performance was measured out of sample on a left out test set. The function used to generate these datasets was randomly drawn from the space of 4-dimensional polynomials: $f(\mathbf{x}) = ax_1^4 - bx_2^3 - cx_3^2 - dx_4^4$. Inputs were sampled from the 4-dimensional unit cube $x \in \{[-1, 1]^4\}$ and the training sets were normalized. Labels again were scaled into the range [0,3].



(a) "sparse" (b) "medium" (c) "dense"

Figure 2.4: The histograms show the average label distribution for all three types of training sets (cf. text). The y-axis shows the number of elements having label given by the x-axis.

Table 2.2: Results for the virtual screening datasets for the two baseline models and the new StructRank approach (highlighted in gray). Bold numbers mark significant improvements with p-value ≤ 0.05 over approaches given as superscript: $^1 \hat{=}$ SVR and $^2 \hat{=}$ RankSVM. For all performance measures higher numbers indicate better results.

	Method	COX-2	BZR	DHFR
$NDCG_{10}$	SVR	0.920	0.877	**0.872**^{2}
	RankSVM	0.928	**0.901**^{1}	0.798
	StructRank	0.921	**0.919**^{1}	**0.905**^{1,2}
ER_{10}	SVR	5.452	3.955	**16.061**^{2}
	RankSVM	5.583	**4.310**^{1}	13.966
	StructRank	5.326	**4.527**^{1}	**17.168**^{1,2}
RIE	SVR	4.692	3.481	**11.939**^{1}
	RankSVM	4.736	3.575	11.010
	StructRank	4.595	3.698	**12.604**^{1,2}

2.4 Results

The comparative evaluation of support vector regression (SVR), Ranking SVM and the proposed StructRank approach on the virtual screening datasets, published by Sutherland et al. [125] is discussed in the next paragraphs. Afterwards the toy example will be evaluated to shed some light on the results obtained for the virtual screening datasets.

2.4.1 Virtual Screening Datasets

Table 2.2 summarizes the ranking performance measured in terms of NDCG, ER and RIE for both baseline models and the new ranking approach StructRank. Only the first 10 ranks are taken into account, which means cutoffs of 10 for $NDCG_{10}$ and ER_{10}, as well as a parameter α for RIE, which puts the most weight on the top 10 ranks. Note that the k-fold cross-validation described before was applied to optimize all three approaches were with respect to NDCG. The barplot in Figure 2.5 shows the results in terms of NDCG, where error bars indicate standard error.

Starting with the dense dataset *COX-2* we observe that all three approaches perform nearly equally well in terms of NDCG, with no approach gaining a significant advantage over the others. These results are confirmed by the two "virtual screening" performance measures ER and RIE. For *BZR*, which could be classified as "medium" in terms of the high

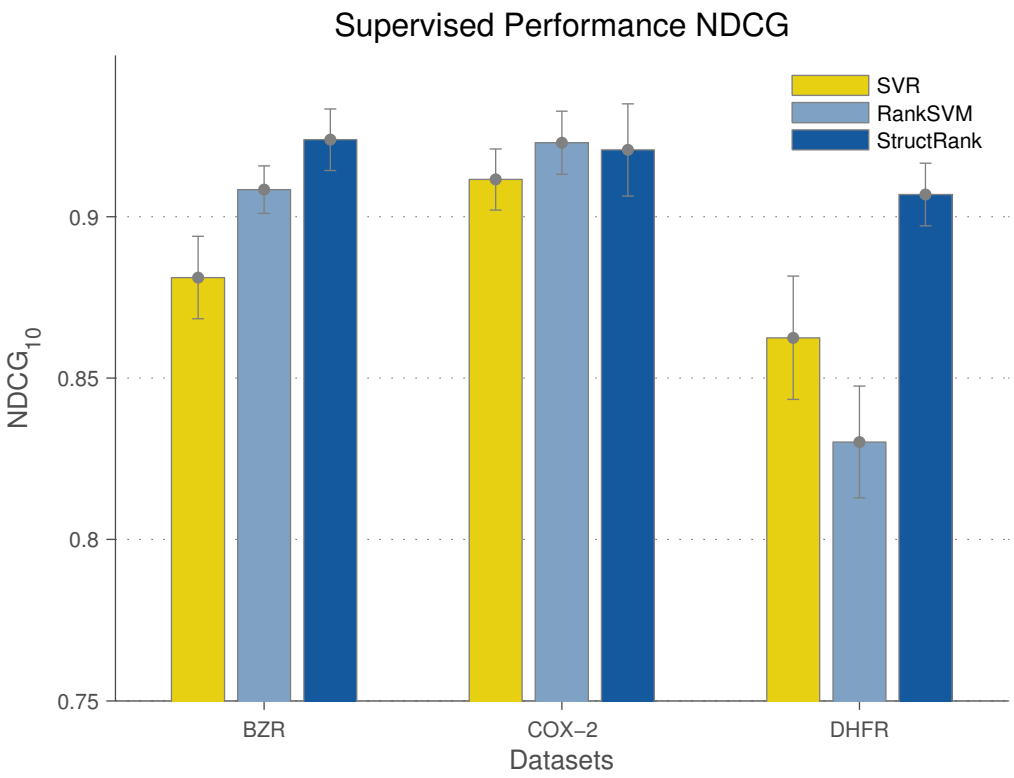


Figure 2.5: Averaged ranking performance measured in NDCG for the virtual screening datasets. Error bars indicate standard error.

labeled molecules, the new approach performs better than both baseline algorithms in terms of NDCG. Both ranking approaches RankSVM and StructRank improve significantly over SVR. These results are confirmed by ER but not by RIE. Finally, for the "sparse" dataset *DHFR*, the new approach exceeds both baseline methods in terms of all three performance measures. RankSVM is notably inferior to the others with a p-value below 0.001.

Subsuming our observations, we state that the new ranking approach outperforms both baselines on the BZR and the DHFR set while on the "dense" dataset COX-2, all approaches perform equally well. This dataset contains many molecules with high labels, thus the event that one of these molecules is ranked high by chance is very likely. For BZR we see (Figure 2.3) that the topmost bins, representing molecules with the highest labels, are sparsely populated. But subsequent bins, representing molecules with slightly lower labels, show a dense population like for COX-2. But these "sparse" bins seem to make it harder to obtain the perfect ranking, as performance drops in terms of NDCG for SVR and RankSVM. For the "sparse" dataset DHFR we observe another decline in terms of ranking performance. Containing only very few molecules with high labels, this dataset seems to be the hardest but also the most realistic virtual screening scenario. Thus we observed a continuous decline of performance of the baseline methods with decreasing number of highly labeled molecules.

2.4.2 Toy Example

The NDCG results obtained for SVR, RankSVM and StructRank on the three different artificial label distributions (described in 2.3.1) are illustrated in Figure 2.6. The results reveal nearly the same behavior as for the real world virtual screening datasets. The "dense"-type dataset has a big number of data points with large label and is therefore comparable to COX-

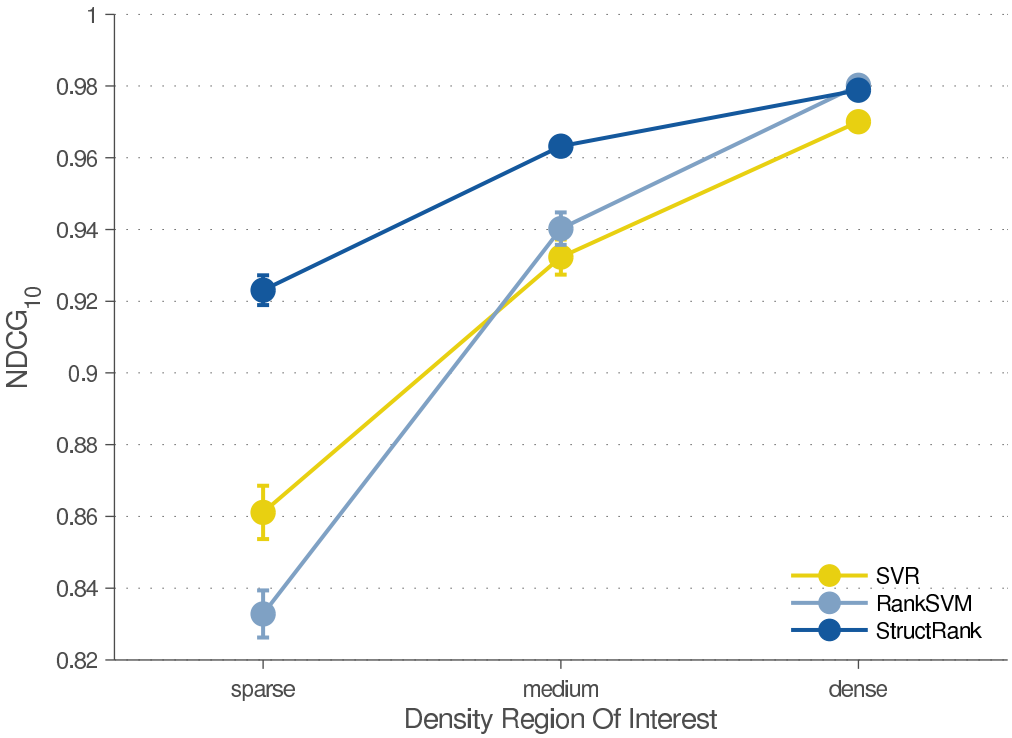


Figure 2.6: Ranking performance of support vector regression (SVR), ranking support vector machine (RankSVM) and structural ranking (StructRank) for three different types of training sets. The region with high labeled examples was covered either sparsely, medium or densely. Error bars indicate standard error.

2. Like for COX-2 all approaches perform equally well. The "medium"-type dataset has less data points with large labels and is comparable to BZR. Performance drops for both baselines, whereas StructRank's performance stays nearly the same. Also like for BZR RankSVM performs slightly better than SVR.

Finally the "sparse"-type dataset is comparable to DHFR, having the lowest number of data points with large labels. Being the most difficult dataset all approaches display a drop in ranking performance. Nevertheless for StructRank the drop is small compared to the other models, which are both clearly outperformed. Interestingly, SVR and RankSVM display the same behavior as for the virtual screening datasets: While RankSVM has the lead over SVR for the "medium" dataset, SVR has the lead over RankSVM for the "sparse" dataset.

2.4.3 Run Time Comparison

This section gives an overview of the CPU time needed by each approach for training and prediction. The given values represent average values received for the virtual screening datasets, i.e. a model is trained on about 225 molecules to obtain a prediction for the remaining test set. SVR requires the least CPU time to train a model since it needs to solve only one optimization problem. RankSVM has to solve a much more complex optimization problem which is reflected in the increased time needed. For StructRank the optimization problems become too big to be solved within one step. Thus an iterative branch-and-bound technique [137] is applied, where for each iteration a convex quadratic subproblem has to be solved. This repeated convex optimization step is the reason for the increase of CPU time by the factor of 25 compared to the SVR. For prediction time we have inverse results with the ranking approaches performing fastest.

Table 2.3: Average CPU time for training/prediction for the virtual screening datasets.

	SVR	**RankSVM**	**StructRank**
Training	0.18 s	1.71 s	2.32 s
Prediction	0.31 s	0.05 s	0.05 s

To investigate the dependency of time needed to train a model on the size of the training set the largest dataset, DHFR, is used. Increasing the training set in steps of 100 molecules revealed (cf. Table 2.4) that CPU time scales linearly with the size of the training set. This indicates, that StructRank can be applied to much larger virtual screening datasets with reasonable performance.

2.5 Discussion

This work investigated the use of ranking approaches when building QSAR-models for ligand-based virtual screening. Two ranking approaches, optimizing NDCG (StructRank) and Kendall's τ (RankSVM), were compared to one state-of-the-art approach for virtual screening: support vector regression. The performance was measured using NDCG as well as two established VS metrics: Enrichment Factor and Robust Initial Enhancement.

This was the first time a ranking approach similar to StructRank was used within the field of QSAR modeling. Regarding the mathematical concept, using a ranking approach like StructRank offers two advantages for virtual screening:

1. **Direct optimization of rankings**: StructRank directly optimizes a ranking measure, compared to the indirect optimization of regression approaches, which in the first place optimize a regression performance measure.
2. **Focus on highly binding compounds**: Because of its composition, NDCG focuses on molecules with high binding coefficients, whereas regression approaches like SVR or ranking approaches like RankSVM pay equal attention to each molecule owing to the structure of their loss functions. Thus necessary complexity for solving the problem may be wasted uniformly over the data instead of focusing the algorithms complexity on high rank entries.

Furthermore runtime seems to be no real obstacle, as it scales linearly with training set size. Thus even for much larger datasets a competitive performance is probable. To encourage future work on this assumption the source code of StructRank together with a documentation has been made publicly available[2].

The evaluation results demonstrate that for datasets which possess only a small or medium number of molecules with high binding coefficients (e.g. BZR and especially the "sparse" DHFR) the new approach performs significantly better than the baseline methods. For

[2]See `http://doc.ml.tu-berlin.de/structrank/`

Table 2.4: CPU time for training a model on DHFR for different training set sizes

	100	200	300	400	500	600	672
CPU Time	0.90s	0.99s	1.07s	1.16s	1.33s	1.54	1.62

datasets which show a high density for these molecules, ranking approaches deliver no real advantage (e.g. for COX-2). These findings are underlined by the toy example.

In summary, structural ranking represents a promising new approach in chemoinformatics that is very natural for virtual screening. From a machine learning point of view this study indicates that ranking approaches in general may outperform regression approaches on a ranking task if the underlying dataset shows a "sparse" data distribution with very few elements on top of the label range.

Chapter 3

Optimal Combination of Models

3.1 Introduction

A multitude of approaches to model drug absorption, distribution, metabolism, excretion, toxicity and other properties relevant in drug design have been published within the last two decades [34, 87, 130, 115, 86, 72]. The data analyzed in this study concerns a potassium channel which is of critical importance for the repolarization of cardiac muscle cells [32, 104]. Blocking the human ether-a-go-go[1] related gene-encoded potassium channel (hERG channel) results in an abnormal activity of the heart characterized by a prolonged QT interval in the electrocardiogram. QT prolongation enhances the risk of potentially fatal "torsades de pointes". A number of drugs such as terfenadine or cisapride were withdrawn from the market due to QT prolongation induced by an unwanted blockade of the hERG channel.

Therefore, it is highly desirable to identify compounds which exhibit hERG inhibition early in the discovery process [121], and many in silico methods have been developed and established to either cope with limited capacities for in-vitro testing or to assess virtual compounds (see [4, 5, 55, 52, 53] for a detailed review). Though the usage of different machine learning techniques and descriptor sets comprises different ideas and concepts, the resulting models for hERG inhibition almost all reached the same level of accuracy. So far less efforts have been made to investigate how the individual regression models can be fused to obtain more robust and/or accurate models. In machine learning so called *ensemble methods* like random forests or bagging approaches build on a similar idea: a prediction model is built by combining the strengths of a set of simpler base models [46]. For categorical models, ensemble or consensus approaches often outperform individual models [88]. In general the advantage lies in the accumulation of many weak learners which require very low computational costs. In contrast this chapter is concerned with different ways to fuse advanced regression models used in drug design.

The most straight forward approach to ensemble modeling calculates the average prediction of all models [96]. This way underestimates and overestimates may mutually compensate to yield a good average performance. Alternatively, one can select the model that will most likely exhibit the lowest prediction error for each individual compound. The model is selected according to the similarity to the closest member of a set of reference compounds with known experimental values. This approach was proposed by Kühne and coworkers and applied to water solubility [68]. The similarity to a set of reference compounds can be

[1] William D. Kaplan named a gene found in Drosophila fly and related to the later discovered hERG gene "Ether-a-go-go gene" inspired by the reaction flies with mutations in this gene show when anaesthetized with ether: their legs start to shake, like the dancing popular in the 1960s at the Whisky A Go-Go nightclub in West Hollywood, California [126].

exploited in an alternative approach to correct each individual prediction by a local bias estimate. This refers to the idea of associative or correction libraries that has been recently introduced and applied to different ADMET endpoints [128, 129, 99, 14].

In this chapter we extend the concept of model selection to the development of a biased model in which the expected prediction error of the selected model is used as a local correction term. Moreover, alternative ensemble predictions are compared to the corresponding single results and the impact of data- and method diversity on ensemble performance is discussed. Additionally, the estimate of the predictivity of our models is improved by performing clustered cross-validation with multiple random partitions in addition to standard cross-validation experiments.

3.2 Data

Two datasets entered the pre-processing process: The first one included 563 compounds with pIC_{50} in-house measurements of hERG inhibition performed by Boehringer Ingelheim Pharma GmbH (the experimental protocol is given in [43]). The second consists of 113 measurements of hERG inhibition for drug like compounds that were gathered from the literature by Kramer et al. [66]. In order to merge and further process these data all compounds are first represented as vectors of molecular descriptors and then computationally analyzed for overlaps and outliers.

3.2.1 Molecular Descriptors & Pre-Processing

To cover different aspects of chemical information we included descriptor sets derived from the 2D structure of the molecule as well as a 3D characterization of interaction between the molecule and its surroundings. Descriptors were generated as follows: All compounds were ionized at pH 7.4 according to ChemAxon's pKa predictor (JChem pKa plugin, ChemAxon Kft, Budapest, Hungary). Then a single 3D conformation was generated with Corina (Version 3.4, Molecular Networks GmbH, Erlangen, Germany) to calculate descriptors based on the 3D structure of a molecule. The conformational energy was minimized in the MMFF94x force field available in MOE (MOE 2007.09, Chemical Computing Group, Montreal, Canada). Chemical properties based on the 1D and 2D representation of the molecule such as size, weight, lipophilicity, atom and ring counts, and topological features were taken from the 2D subset of the QSAR descriptors available in MOE. The 2D topological arrangement of pharmacophoric interaction points was characterized by ChemAxon pharmacophoric fingerprints (ChemAxon Kft, Budapest, Hungary) and CATS descriptors [106]. To assess the interaction energy of the molecule with its environment, we employed the VolSurf package (vsplus 0.4.5a, Molecular Discovery Ltd, UK), using four standard chemical probes (water, hydrophobic probe, carbonyl oxygen, and amide nitrogen [23, 33]).

All descriptors were pre-processed as follows: Constant features, i.e. those that do not change over all compounds, were removed. Counts in MOE 2D descriptors and ChemAxon pharmacophoric fingerprints were log scaled, i.e. $x = \log(\mathrm{abs}(x) + 1) \cdot \mathrm{sgn}(x)$. Finally, all features were normalized in the following way: From each feature the median of this feature over all data was subtracted and the feature was normalized such that the largest absolute value was smaller than four (in preliminary studies this approach turned out to be more robust compared to common standardization when the distribution of the descriptors is skewed). All three descriptor sets, the ChemAxon pharmacophoric fingerprints, the CATS descriptors and the VolSurf descriptors were concatenated to one vector for each compound.

In cross-validation runs and for evaluation purposes the pre-processing was done on the training part of the data only. The results from the training set were then applied to the respective test data (i.e. remove those features that were constant in the training data and normalize using the parameters calculated on the training data).

3.2.2 Comparison of in-house and literature data

The literature and the in-house data span a similar range of pIC_{50} values. Moreover, on the ten reference compounds present in both sets the measurements show a correlation of $r^2 = 0.9$ (deviation below 0.5 log units for seven of them).

Though both sets are comparable with respect to the pIC_{50} labels, the PCA plot in 3.1 reveals that the two sets are centered in different regions of the chemical space. Thus the predictive power of separate machine learning models for the two sets and a single model built using data from both sets was evaluated. In a standard cross-validation setting, it was not possible to use the in-house data to predict literature data and vice versa. On the in-house data alone, i.e., train and test set from in-house data, the performance is slightly better than on the literature data alone (presumably since the in-house data are more consistent). The performance of models built using all data is very close to this single set performance on the corresponding datasets. Altogether these results indicate that literature and in-house data can safely be pooled in one dataset.

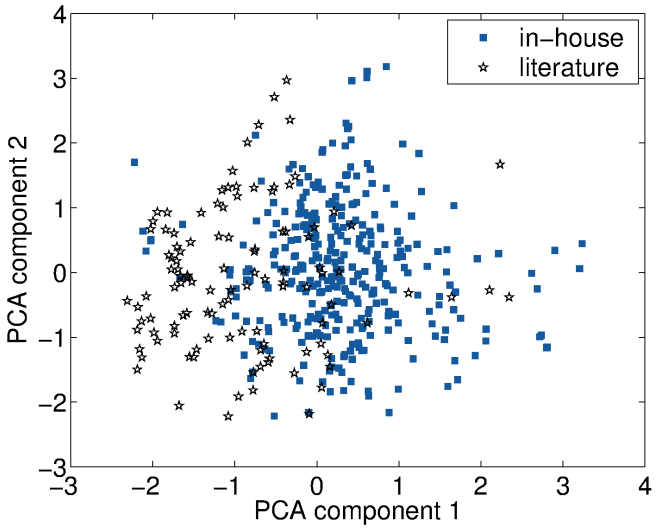


Figure 3.1: Projection of the dataset on the first and second component of a PCA model using all descriptors. The in-house and literature set are marked to illustrate the different distributions.

3.2.3 Analysis of Outliers

The outlier analysis is based on visual inspection of the raw descriptors, different PCA visualizations and the κ, γ and δ indices[2] introduced by Harmeling et al. [45]. All measures indicated that several percent of all compounds in the dataset might be outliers. The δ-index which quantifies the amount of extrapolation necessary to predict the corresponding data point was then applied to define a set of outliers. The distributions of δ-indices computed

[2] κ is the distance to the mth nearest neighbor; γ is the mean distance to m nearest neighbors; δ captures if a point is well embedded by its neighbors.

for each set of descriptors separately using $m = 7$ neighbors is shown in Figure 3.2. After visual inspection, the number of outliers was set to the top 50, i.e. by this working definition, a compound is an outlier if its δ-index is in the top 50 of δ-indices for any of the four sets of descriptors. The complete evaluation of single models (see Evaluation Strategy) was then performed twice, once including and once excluding the outliers.

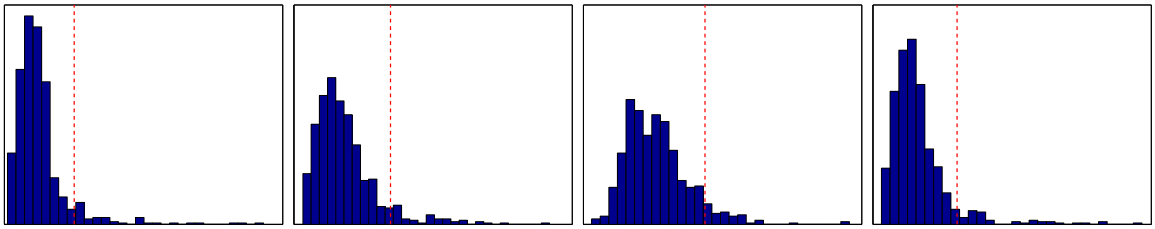


Figure 3.2: Histograms of δ outlier scores for each descriptor set (from left to right: ChemAxon, CATS, MOE, and Volsurf) using $m = 7$ neighbors. Dashed vertical lines indicate the respective cut-off corresponding to treating the 50 compounds with the highest scores as outliers.

3.3 Methods

The hERG pIC_{50} inhibition value was measured for the given set of chemical compounds. Based on these measurements we aim to predict the pIC_{50} inhibition value of new chemical compounds. More precisely, we look for a regression function f which can predict the pIC_{50} inhibition value for any compound represented as descriptor vector $\mathbf{x}$.

3.3.1 Single Modeling Approaches

In a first step four standard machine learning algorithms were used to build regression models:

- Ridge regression model,
- Gaussian process model,
- Support vector regression model and
- Random forest.

(See Chapter A of the appendix for a description of these methods.)

Kernels For the support vector regression model a radial basis function kernel, which showed good performance with SVR in previous QSAR studies (e.g. [115]), was selected. In the case of the Gaussian process we applied a combination of the radial basis function and the rational quadratic kernel function

$$k(\mathbf{x}, \mathbf{x}') = \eta \exp\left(\frac{-\|\mathbf{x} - \mathbf{x}'\|^2}{\sigma^2}\right) + (1 - \eta)\left(1 + \sum_{i=1}^{d} w_i (x_i - x_i')^2\right)^{-v},$$

with $\eta \in [0, 1]$.

Baseline Model All modeling approaches are compared to a baseline model. This model predicts identical values for each compound, namely the average target value seen in the training data.

3.3.2 Ensemble Modeling Approaches

The individual models listed above can be combined in different ways to obtain ensemble models. Following the work by Kühne et al. [68] we assume the following general setting: Several *single models* $f_i, \quad i = 1, \ldots, l$ have been generated on training data and evaluated on disjoint test sets. For all compounds in the test set, the true value is known. Now an unknown compound t is added to the test set.

The central idea is to derive a prediction $f^*(\mathbf{x}_t)$ for the unknown compound t by considering the performance of the models on the neighboring compounds within the test set. Thus, the test set can be considered as *correction set*. When validating such an approach we employ cross-validation, i.e. the test set of each cross-validation run is considered as correction set. Each compound of the correction set is left out once, and its predicted value is deduced from the neighboring compounds in the correction set.

To determine the neighborhood of each compound a measure of molecular similarity is required. In this work the distance of two chemical compounds a and b is defined as the Euclidean distance of the corresponding descriptor vectors taking a reduced set of normalized features into account

$$\| \mathbf{x}_a - \mathbf{x}_b \|_{red} \, . \tag{3.1}$$

Due to the curse of dimensionality, measuring the Euclidean distance in the whole descriptor space of more than 400 dimensions would result in unspecific distances. To diminish this effect, the descriptor vector is reduced to a small selection of descriptors which are most relevant in the context of hERG inhibition. Initial experiments showed that a set of 54 features with the highest weighting factors according to automatic relevance determination [97] in the GP model forms an appropriate set of descriptors (all names of the features are listed in supporting information of [43]).

Ensemble Models

The following ensemble modeling approaches were evaluated:

Selection by MAE (MAE Model) The single model with the *lowest mean absolute error on the neighboring compounds* is selected to predict the desired inhibition value [68]. The k nearest neighbors are selected based on the distance measure introduced in Equation 3.1, and the mean absolute error (MAE) is calculated as:

$$\mathrm{MAE}(f_i) = \frac{1}{k} \sum_{j=1}^{k} \mid f_i(\mathbf{x}_j) - y_j \mid \, . \tag{3.2}$$

Here f_i refers to one of the l trained single models. The predicted value $f^*(\mathbf{x}_t)$ of this ensemble model is given by

$$f^*(\mathbf{x}_t) = f_{\mathrm{minMAE}}(\mathbf{x}_t) \quad \text{with} \quad f_{\,\mathrm{minMAE}} = \underset{f_i \; i=1,..,l}{\operatorname{argmin}} (\mathrm{MAE}(f_i)). \tag{3.3}$$

If not stated otherwise, we set the number of nearest neighbors k that are considered in this or any other of the following ensemble approaches to $k = 10$.

Weighted Model This model is based on the idea that a *weighted sum of all predictions* of the different single models may result in a greater improvement than selecting the prediction of only one model. In the simplest way, all individual predictions can be combined with equal weighting, i.e. the average predicted value is calculated from all models. Here we determine the weight of each model, υ_{f_i}, according to the inverse mean absolute error a model achieves on the neighboring compounds:

$$\upsilon_{f_i} = \frac{1}{\mathrm{MAE}(f_i)} \left(\sum_{j=1}^{l} \frac{1}{\mathrm{MAE}(f_j)} \right)^{-1} \quad \text{with} \quad \sum_{i=1}^{l} \upsilon_{f_i} = 1. \tag{3.4}$$

The prediction of the weighted model is then given by

$$f^*(\mathbf{x}_t) = \sum_{i=1}^{l} \upsilon_{f_i} f_i(\mathbf{x}_t). \tag{3.5}$$

The higher the accuracy of a model f_i in the neighborhood of $\mathbf{x}_t$, the greater the impact of the model f_i on the predicted value.

Bias Corrected Model In this approach one single model is selected according to the minimum mean absolute error on the neighboring compounds—similarly to the MAE model. Then the prediction of the selected model is *corrected by the mean error* on the neighbors. To incorporate the distance between the unknown compound t and its neighbors we define a *distance weight* d_j for each of the k nearest neighbors as

$$d_j = \frac{1}{\parallel \mathbf{x}_t - \mathbf{x}_j \parallel_{red}} \left(\sum_{i=1}^{k} \frac{1}{\parallel \mathbf{x}_t - \mathbf{x}_i \parallel_{red}} \right)^{-1} \qquad j = 1, \ldots, k. \quad \text{with} \quad \sum_{j=1}^{k} d_j = 1. \tag{3.6}$$

This way close compounds receive high distance weights. The selected model is now given as

$$f_{\text{weightedDist}} = \underset{f_i \; i=1,..,l}{\operatorname{argmin}} \left(\frac{1}{k} \sum_{j=1}^{k} \frac{\parallel f_i(\mathbf{x}_j) - y_j \parallel_{red}}{d_j} \right). \tag{3.7}$$

The prediction is given as the prediction of $f_{\text{weightedDist}}$ reduced by the prediction error on the neighborhood

$$f^*(\mathbf{x}_t) = f_{\text{weightedDist}}(\mathbf{x}_t) - \frac{1}{k} \sum_{j=1}^{k} \frac{f_{\text{weightedDist}}(\mathbf{x}_j) - y_j}{d_j}. \tag{3.8}$$

This approach is closely related to the concept of correction libraries [99, 14].

Average KNN Model and Random Choice Model These reference models quantify the improvement achieved by applying ensemble models. In the Random Choice model the prediction of one single model is chosen *randomly* as the predicted value.
Unlike all other models, the Average KNN model predicts the value for the unknown compound without considering any single models f_i. The *average over the labels* of the neighboring compounds

$$f^*(\mathbf{x}_t) = \frac{1}{k} \sum_{j=1}^{k} y_j \tag{3.9}$$

serves as prediction.

3.4 Evaluation Strategy

3.4.1 Evaluation of Single Models

In order to test and compare the performance of all modeling approaches that have been introduced so far, two cross-validation schemes are applied: First all models are evaluated in *standard cross-validation* setting with three folds and 50 repetitions. Then the standard error and the variance of different performance measures over all 50 trials is calculated.

Additionally, we evaluate the models in a *clustered cross-validation* setting. The dataset is grouped into 15 equally sized clusters using the geo-clust algorithm [146] and the similarity measure introduced in Equation 3.1. Each cluster is then randomly allocated into three folds, each composed of five clusters and processed as in a standard three-fold cross-validation setting. This form of cross-validation helps to prevent too optimistic performance estimates by avoiding very similar compounds in both the training and the test set.

Hyperparameter For the ridge regression model and the support vector regression a three-times five-fold *nested cross-validation* is implemented to determine hyper-parameters that result in good generalization on unseen data. In case of ridge regression, the parameter λ is optimized, while for support vector regression, the hyper-parameters λ, σ and ε need to be determined (see Machine Learning Methodology). The implementation of the random forest is based on the algorithm introduced by Breiman [11]. Each tree is trained on the full training set and the parameters are kept constant. Contrary to the other learning techniques, all parameters in the Gaussian process are estimated on the fly using marginal likelihood maximization and not specified a priori or in an inner cross-validation loop.

3.4.2 Evaluation of Ensemble Models

In order to evaluate the *impact of model- and data-diversity* on performance improvements we consider two different settings: First a support vector regression, a random forest and a Gaussian process model are trained on the same set of compounds and then combined. In a second run we fuse several models that were obtained from the same learning algorithm but trained on different sets of compounds. This way we can distinguish between the performance improvement which results from the variety of machine learning methods and the improvement which originates from differences in the training sets. In the second setup the discussion is restricted to random forests; the evaluation of our single models shows that similar results can be expected for SVR of GP models (see below).

As in the evaluation of the single models, all ensemble models are evaluated in three-fold clustered cross-validation over 50 repetitions. In each cross-validation loop, the five left-out clusters represent the correction set. In the second ensemble setting we use the training set (composed of 10 clusters) to create 20 different subsets using a bagging approach, i.e. subsets are sampled with replacement. Each of the 20 different subsets is then used to train a regression model of the same type, e.g., a random forest. The ensemble of 20 bagged random forests is evaluated on the left out correction set as described in the first ensemble setup.

Performance Measures Models are evaluated with respect to the following performance criteria:

- **RMSE:** The root mean squared error defined as

$$\sqrt{\frac{1}{n}\sum_i (y_i - f(\mathbf{x}_i))^2}. \tag{3.10}$$

- **Correlation:** The r^2 value or squared correlation coefficient defined as

$$r^2 = \frac{(\sum_i ((y_i - \bar{y})(f(\mathbf{x}_i) - \bar{f}(\mathbf{x})))^2}{\sum_i (y_i - \bar{y})^2 \sum_i (f(\mathbf{x}_i) - \bar{f}(\mathbf{x}))^2.} \tag{3.11}$$

  where y_i are the labels, $f(\mathbf{x}_i)$ are the predictions and $\bar{x}$ denotes the respective mean values.

- **LOG-ϵ:** The fraction of predictions within a specific interval ϵ around the true value.

3.5 Results and Discussion

3.5.1 Single Models

Table 3.1 compares the average performance (measured via RMSE, r^2, LOG05 and LOG1) of the five single models in standard cross-validation and in clustered cross-validation. The distribution of RMSE values is illustrated in Figure 3.3, the relation between prediction and measured pIC_{50} value is visualized in Figure 3.4.

standard cross-validation

Method	RMSE	r^2	LOG05	LOG1
Baseline Model	0.86	-	0.50	0.77
ridge regression	0.91	0.25	0.51	0.79
Gaussian process	0.62	0.49	0.66	0.92
support vector regression	0.62	0.48	0.66	0.91
random forest	0.63	0.48	0.64	0.91

clustered cross-validation

Method	RMSE	r^2	LOG05	LOG1
Baseline Model	0.87	-	0.49	0.77
ridge regression	1.15	0.11	0.38	0.65
Gaussian process	0.73	0.30	0.54	0.85
support vector regression	0.73	0.29	0.55	0.84
random forest	0.73	0.31	0.55	0.85

Table 3.1: Evaluation of Single Model Approaches: Different error measures applied in the standard cross-validation setting and the clustered cross-validation setting. RMSE denotes the root mean squared error, r^2 the correlation coefficient and LOG05 and LOG1 the fraction of predictions falling within 0.5 and 1 (log) units of the true value, respectively. The corresponding standard errors across all 50 repetitions are all below 0.01. See text for details.

Complexity of models and data The ridge regression model does not perform very well, moreover, it is outperformed by the baseline model. In contrast, all other models yield results significantly improving upon the baseline prediction. Hence, a linear model seems to be too "simple", i.e. it has not enough flexibility to capture the complex molecular mechanisms which determine the inhibition of the hERG channel. Although the dataset is relatively small, and taking the complexity of the biological mechanism into account, the nonlinear

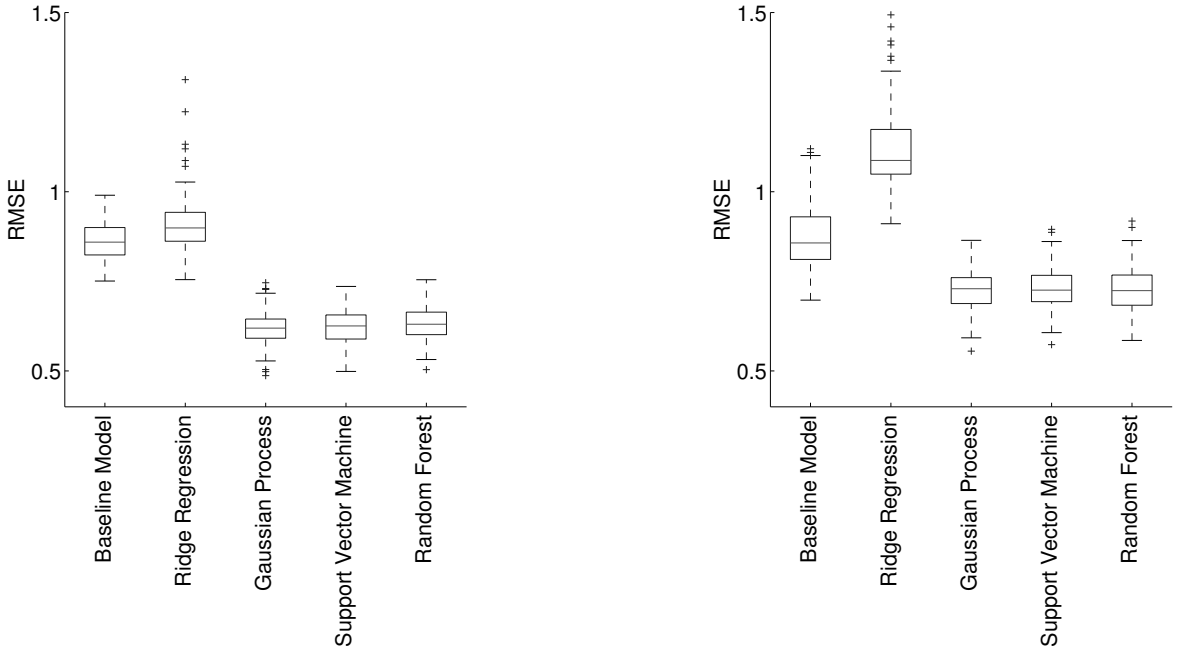


Figure 3.3: Box-plot depiction of the root mean squared error (RMSE) in the *standard* (left) and *clustered* (right) cross-validation setting over 50 repetitions. The box covers 50% of the actual data, the box height being the interquartile range, the horizontal line denotes the median. The whiskers are at most 1.5 times the interquartile range. Points outside this range are marked as outliers.

methods yield models where up to two thirds of all predictions are within 0.5 log units of the experimental value. The results across all 50 repetitions of our cross-validation experiments are very consistent, showing only a small "within-method" and "between-method" variance. From this we conclude that all models are close to the performance which is achievable on this dataset. Notably, this holds true when comparing the kernel-based learners SVR and GP and the density-based random forest.

Outliers The model performances when using training sets with and without the outliers identified by the δ-indices did not differ significantly for GP, SVR and RF models. The learning algorithms seem to be robust enough to deal with the outliers included in the present set of data. Hence, results obtained on the limited dataset are not discussed any further.

Clustered cross-validation The *clustered cross-validation* and the *standard cross-validation* show the same tendencies but the latter one yields more optimistic performance estimates—especially for ridge regression. As shown in Figures 3.4 and 3.5 the spread of the predicted y-values is significantly smaller in the clustered cross-validation leading to larger errors on the tails of the pIC_{50} value distribution. When leaving out whole clusters of compounds from the training set the most informative neighboring points of each test compound are taken away, a higher amount of extrapolation is necessary and predictions drift towards the mean value. A similar effect has been observed in many practical applications of QSAR models: During application, contrary to the validation results, a model tends to miss out on the tails, i.e. the high and low values, of the target distribution. This underpins that a clustered cross-validation yields more realistic performance estimates for a real world application. In fact, in a realistic application scenario in drug research, prediction models are often applied to new chemical series which might be significantly dissimilar to the compounds that have entered the model training process.

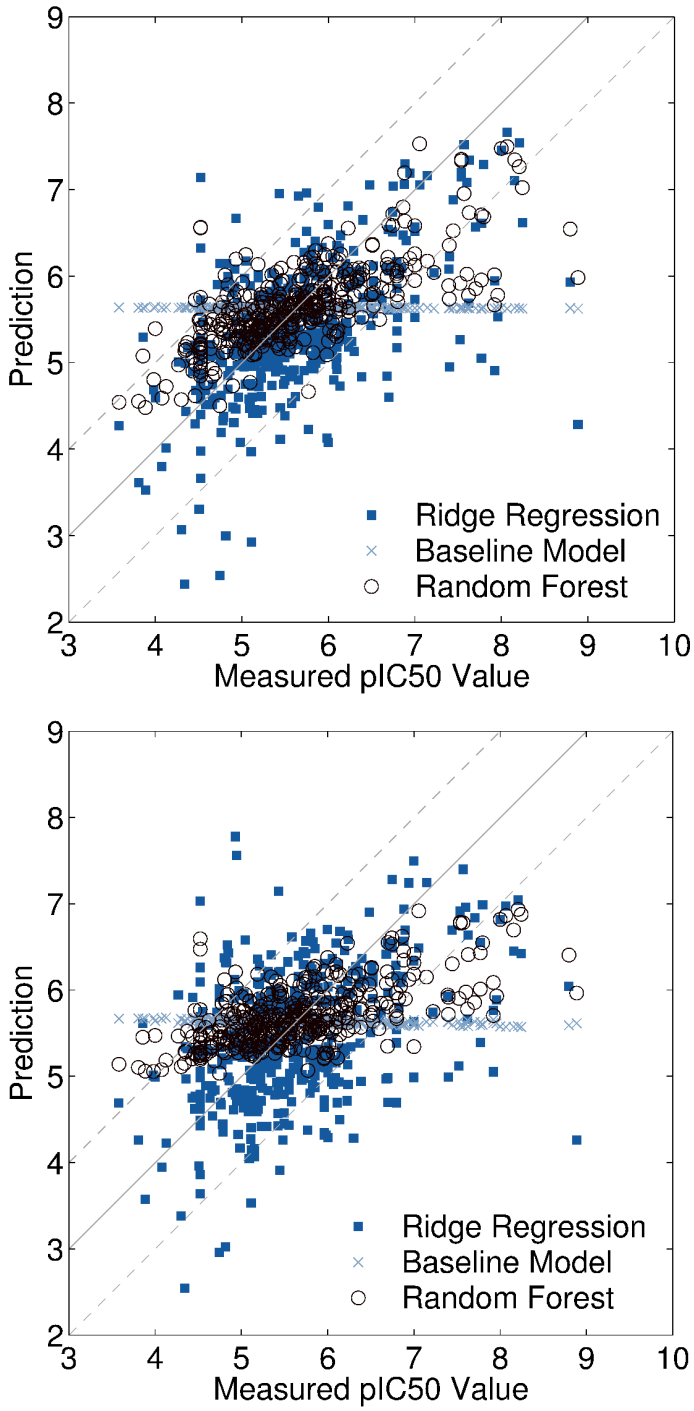


Figure 3.4: Relation between predictions and true values for baseline, ridge regression and random forest model evaluated in the standard cross-validation setting (top) and the clustered cross-validation setting (bottom) over 50 repetitions. Note that the random forest model outperforms the other two approaches and the clustered cross-validation yields less optimistic results. The plots for the Gaussian process and the support vector regression model (not shown) nearly equal the one of the random forest model.

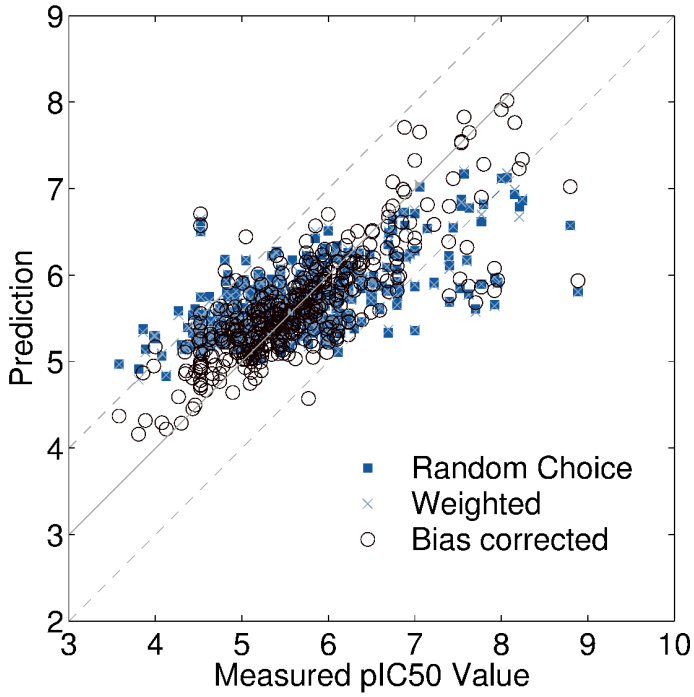


Figure 3.5: Relation between predictions and true values for the Random Choice model, the Bias corrected model and the Weighted model evaluated in the clustered cross-validation setting over 50 repetitions: The Bias corrected model improves the prediction especially in less frequent regions of extreme pIC_{50} values.

The performance of a prediction model applied on a cluster not represented in the training set differs highly between the individual clusters. Figure 3.6 shows the squared correlation coefficient between predictions and measured pIC_{50} values calculated on each cluster separately. There is a large spread in the performance depending on the cluster. Clusters 7 and 11 can be predicted very well, whereas the model fails on other clusters (e.g. 4, 8, 13).

Assuming that the clusters group structurally similar compounds, this finding resembles some of our experiences when applying QSAR models of this type: the interaction between members of certain structural classes and the hERG channel is in some cases only partially covered by the molecular descriptors and our models, whereas the structure-activity relationships revealed by other structural classes are much better reproduced. It is interesting to note that there is no direct correlation between the cluster composition into proprietary and public domain data and the corresponding model performance (data not shown). However, a more thorough analysis of the structural classes represented by each cluster and their putative mode of interaction with the hERG channel is beyond the scope of this paper.

3.5.2 Ensemble Models

Benchmarks for Ensemble Models To allow for more insights into the performance gain achieved by different ensemble strategies we compare them not only to both baseline models (Average KNN and Random Choice) but also to the following two quantities:

- The RMSE of a *single random forest model* is taken as an upper benchmark: The ensembles are expected to achieve a RMSE that is smaller than this upper bound.
- The RMSE of a *leave-one-out cross-validated random forest model* is taken as a lower benchmark. This model is trained on all compounds in the training set and all (but one) com-

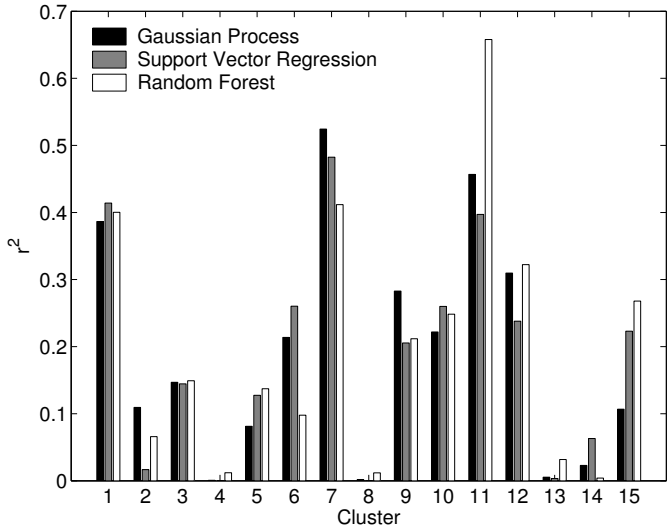


Figure 3.6: Calculation of the squared correlation coefficient on the clustered cross-validation results for each cluster separately.

pounds in the correction set. Since only one test compound is left out in each iteration the model has almost full knowledge (contrary to the ensembles which are validated in clustered cross-validation).

Combination of Different Single Models Trained on Equal Training Sets

Figure 3.7 visualizes the distribution of the RMSE over 50 repetitions for different ensemble approaches when training three single models on identical datasets. The *Random Choice* model does not improve over any single model since all single models (GP, SVR and random forest) perform about equally well (see Table 3.2).

Also the *Weighted* as well as the *Selection by MAE* approach do not perform significantly better than a random forest (dashed line) or any other single model. The reason for this behavior is illustrated in Figure 3.8: The prediction errors of the individual models are highly correlated, i.e. if one model yields an inaccurate prediction, the other single models show similar prediction errors. Hence, a mutual compensation of prediction errors by combining single model predictions is not possible.

In contrast, the *Average KNN* and the *Bias corrected* model significantly improve over the single model approaches—the *Bias corrected* model even outperforms the random forest model evaluated in leave-one-out cross-validation (lower dashed line). Considering the fact that the *Bias corrected* consensus model is based on about 30% less data points than the leave-one-out random forest model makes this result remarkable. We conclude that the way in which data enters the model can be more important than the number of data points. Here the separate retrospective inspection of only 10 nearest neighbors using bias correction works best—even better than considering all data points from the beginning.

So far the evaluation was focused on ensemble models which incorporate a neighborhood of ten compounds. To determine the influence of the neighborhood size on the quality of the model the number of neighbors is varied and the evaluation repeated. The results are summarized in Figure 3.9: The *Average KNN* and the *Bias corrected* model are strongly dependent on the number of neighbors, where an optimal number of neighbors seems to be 5. However, the RMSE does not significantly decrease when more neighbors are taken into account. The

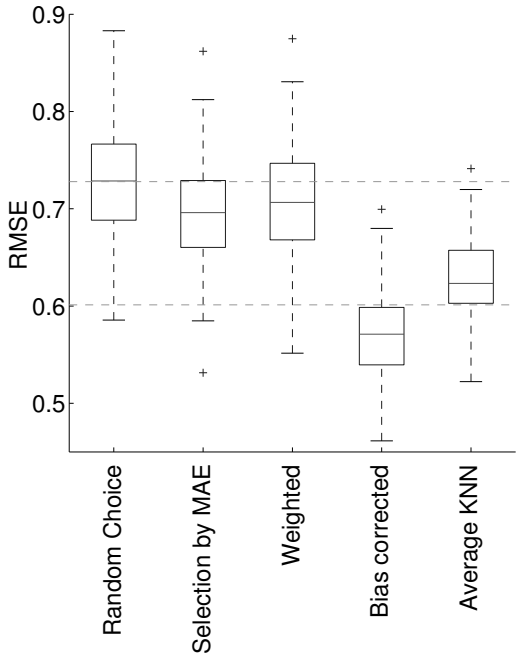


Figure 3.7: Combination of random forest, Gaussian process and SVR model trained on equal sets: Box-plot depiction of the root mean squared error (RMSE) of the different ensemble methods in the clustered cross-validation setting over 50 repetitions. The dashed lines refer to the RMSE of the underlying single random forest model (upper line) and the RMSE of a random forest model trained in leave-one-out cross-validation (lower line). For more details, see also Figure 3.3.

MAE and the Weighted model do not improve with the size of the neighborhood. This may again be caused by the highly correlated prediction errors (c.f. 3.8).

Combination of Single RF Models Trained on Different Training Sets

In this section we evaluate the performance of ensemble models which combine the predictions of 20 random forests trained on different parts of the training set (see bagging approach in Section 3.4). The main results of this evaluation are summarized in Table 3.2 and Figure 3.10. Due to the different training sets, stronger deviations between each pair of single models occur. Some errors of the single models are now compensated in the ensemble model. However, the distribution of RMSE values shows similar tendencies as in the previous setting: For the *Random Choice* model we observe a worse performance than for a single model (upper dashed line). The *Weighted* model again only achieves a small improvement. In contrast to the previous observation, the *Selection by MAE* model now shows a somewhat larger improvement with respect to the single model. The *Bias corrected* model again reveals the best performance of all ensemble methods.

3.6 Conclusions

In this study the performance of several machine learning algorithms in single and ensemble model settings was investigated to address hERG inhibition. Single Gaussian process, support vector regression, and random forest models which were trained on the combined dataset of literature and in-house data gave RMSE values of roughly 0.6 in standard cross-

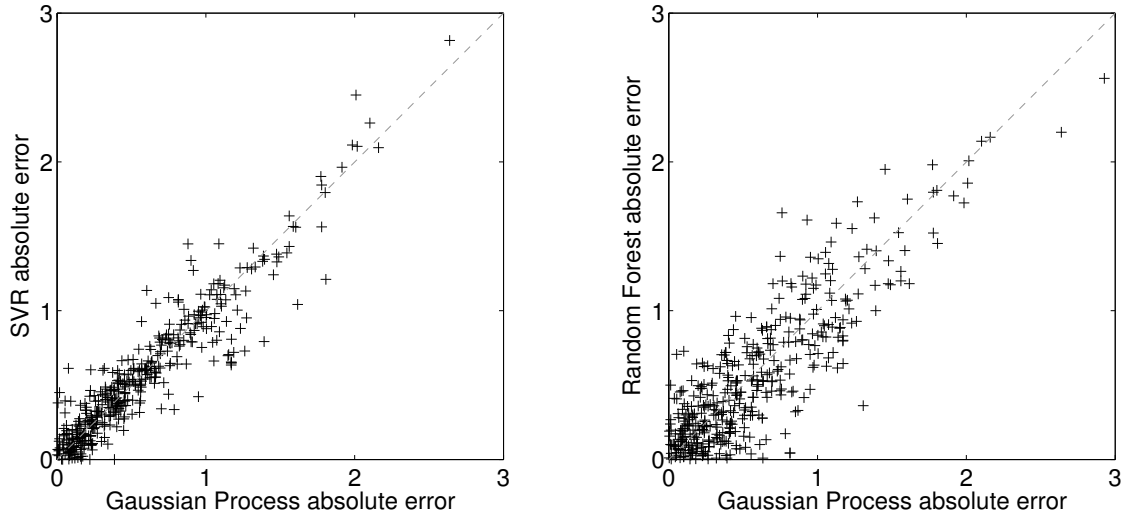


Figure 3.8: Visualization of the strong correlation between the absolute error of the GP and SVR model (left) and the GP and the random forest model (right). The corresponding correlation coefficients amount to 0.96 (GP versus SVR), 0.86 (GP versus random forest) and 0.82 (SVR versus random forest).

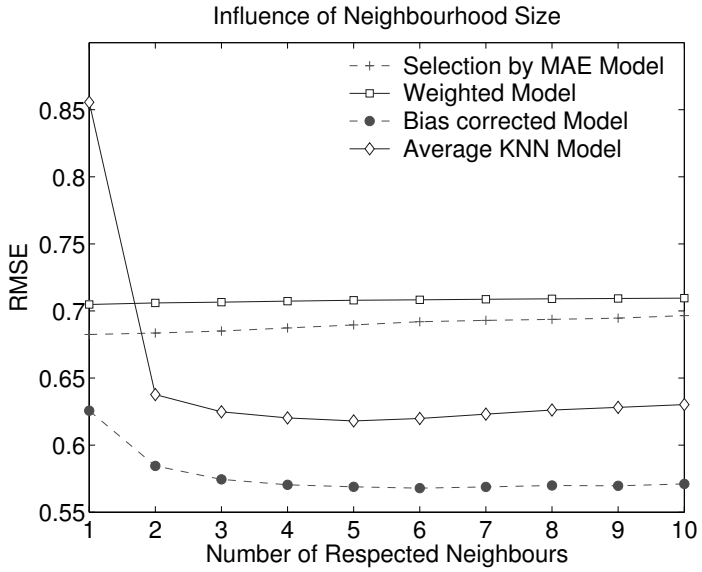


Figure 3.9: Influence of the number of considered neighbors on the ensemble model performance. The Average KNN and Bias corrected model are more sensitive to the neighborhood size than the other models.

combination of GP, SVR and random forest

Method	RMSE	r^2	LOG05	LOG1
Random Choice	0.73	0.30	0.55	0.85
Selection by MAE	0.7	0.35	0.58	0.87
Weighted	0.71	0.33	0.56	0.86
Bias corrected	0.57	0.54	0.71	0.93
Average KNN	0.63	0.44	0.62	0.9
single random forest	0.73	0.31	0.55	0.85
leave-one-out random forest	0.6	0.726	0.66	0.92

combination of random forest models trained on different sets

Method	RMSE	r^2	LOG05	LOG1
Random Choice	0.76	0.26	0.52	0.84
Selection by MAE	0.7	0.35	0.56	0.86
Weighted	0.74	0.31	0.53	0.85
Bias corrected	0.57	0.55	0.69	0.93
Average KNN	0.63	0.46	0.63	0.91
(bagging) single random forest	0.76	0.26	0.52	0.83
leave-one-out random forest	0.6	0.726	0.66	0.92

Table 3.2: Evaluation of Ensemble Model Approaches: RMSE denotes the root mean squared error, r^2 the correlation coefficient and LOG05 and LOG1 the fraction of predictions falling within 0.5 and 1 (log) units of the true value, respectively. The corresponding standard errors across all 50 repetitions are all below 0.02. See text for details.

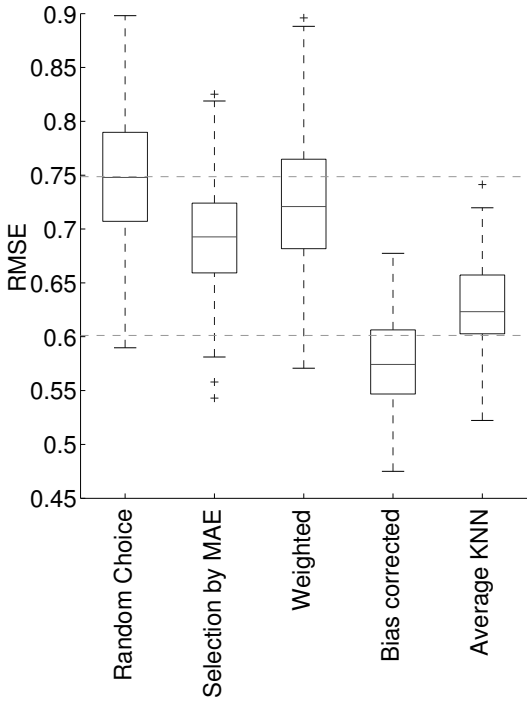


Figure 3.10: Combination of 20 bagging random forest models trained on different sets: Boxplot depiction of the root mean squared error (RMSE) of the different ensemble methods in the clustered cross-validation setting over 50 repetitions. For details, see also Figure 3.3.

validation and about 0.7 in clustered cross-validation, whereas the linear ridge regression model was not able to discover a relationship between the molecular descriptors and hERG inhibition.

Although all three nonlinear models are based on different ideas and algorithms, their prediction errors are highly correlated. Thus the performance of a consensus model based on these three single models only slightly exceeds the single model performance, especially if standard cross-validation is applied. In the more realistic clustered cross-validation setting, combining different models improves the performance of the final model. This can be observed for an ensemble whose individual models are trained with the same dataset as well as for an ensemble based on different training subsets. In both cases, a local bias correction yields the best results.

From a machine learning point of view the results reveal the impact of the test set and single models on ensemble modeling in the presence of correction sets. Even if the single models are highly correlated ensemble modeling can improve the prediction, especially if the test compound is not optimally covered by training data. In this case the utilization of additional data in a local bias correction method is more powerful than a model retrained on the complete dataset including the additional information. Further investigations are necessary to evaluate in which cases the bias corrected approach is adequate and in which cases retraining should be preferred.

For chemical researchers these findings are encouraging in two aspects: First, they indicate that local bias correction may be a good way to cope with the influence of subtle structural modifications on the interaction with the hERG channel while global trends such as the overall influence of compound lipophilicity [145, 38, 55] are still covered. Second, the calculation of a simple local bias correction from new measurements can substitute time-consuming retraining of a model using the expanded dataset, as proposed also in earlier studies [99].

All QSAR models that have been discussed in this study only hardly give practical hints which molecular features should be altered during compound optimization to overcome hERG interaction. They are rather intended to provide a fast and reliable method for assessing large compound sets which originate from HTS and virtual screening campaigns or combinatorial libraries. Of course, a variety of experimental high-throughput methods such as competitive binding, rubidium efflux, or high-throughput automated patch clamp assays are available for these tasks [120, 12]. However, well-tuned in silico models which were trained on high-quality experimental data can be of use especially in an early stage of a drug discovery project. They can be applied for virtual compounds before they are synthesized or purchased from external vendor catalogs. Moreover, in silico methods are faster and cheaper to run. Since the accuracy of the experimental values may suffer from sample impurities, poor solubility, poor chemical stability, a tendency to stick to surfaces or other properties, a predictive in silico model may be a valuable alternative to support decisions such as the prioritization of HTS clusters, selection of compounds from vendor data bases, or even to assist medicinal chemists in prioritizing synthesis plans.

Chapter 4

Structure-based Explanation of Nonlinear Classifiers

4.1 Interpretability of Predictions in Chemoinformatics

In the previous chapters we explored new algorithms and enhancements to improve the prediction performance of machine learning methods in chemical applications. The next two chapters are dedicated to the interpretation of such machine learning predictions. In chemoinformatics the *interpretability* of predictions is a subject of growing attention since it enables one to assess model applicability, reliability of single predictions, and relevant characteristics of the compound in question.

The *applicability domain* [135, 124, 134] of a model refers to the chemical structure space in which a model makes predictions with a given reliability. In drug discovery the investigated compounds commonly lie beyond the region covered by previously examined series of compounds. Thus, prediction results become inaccurate due to missing information in the chemical space of interest. Furthermore, the variable constitution of "activity landscapes" [76] reduces the reliability of predictions. Most modeling techniques capture major trends ("rolling hills") and fail to recognize "activity cliffs" (Figure 4.1).

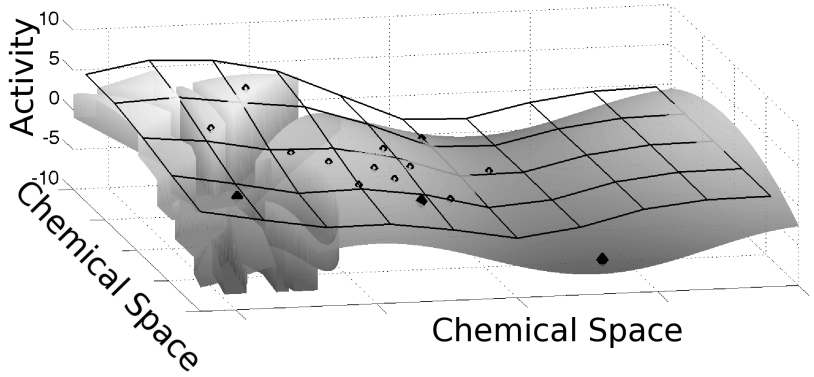


Figure 4.1: Sketch of predictions on changing activity landscape; gray surface: true activity landscape; black grid: prediction model based on training set (small markers); black filled diamonds: three potential test points. The left and right test point are not accurately predicted due to activity cliffs and missing training data.

Given an accurate prediction the question of *feature importance* arises in the context of compound optimization [143]. The chemical characteristics most relevant for the predicted prop-

erty are analyzed to deduce new chemical hypotheses and to design compounds with improved activity. Acceptance of computational models thus depends not only on rigorous verification (determining whether an implementation correctly computes the underlying mathematical model) and statistical validation (determining whether the model adequately represents the modeled phenomena), but also on the ability of the user to understand its predictions and evaluate their quality. The latter is strongly affected by the model interpretability. In brief, interpretable models allow for appropriate utilization and a better understanding of the algorithms as well as general acceptance of in silico predictions.

However, not all prediction models are easy to interpret. Especially kernel-based models, like support vector machines, are often treated as black boxes [29]. In contrast to linear models, they can capture nonlinear relations between hundreds of chemical features and take local activity trends into account. This flexibility and non-linearity makes them both more powerful in terms of solving a larger class of problems and more challenging in terms of interpretation. Namely interpretation of a nonlinear prediction model needs to be local, a straight forward global explanation in terms of features - similarly to a linear prediction model - is systematically unfeasible.

The following two chapters address this challenge in different ways: The method presented in this chapter quantifies the *influence of each training sample* on a single prediction in order to visualize the most relevant training compounds along with a predicted value. This approach primarily evaluates the reliability of single predictions and identifies weaknesses of models due to missing training data, high complexity of the underlying chemical effect or mislabeled training data points.

The next chapter emphasizes the interpretation and understanding of prediction models in order to gain chemical insights on the modeled property. Local gradients are used to determine *feature importance* for kernel-based predictors. In contrast to other feature importance measures this method allows for the detection of local trends, i.e. the chemical features that severely affect complex biochemical properties, like binding affinity, of a certain class of compounds yet have no significant influence with respect to the entire chemical space.

4.2 The Idea of Structure-Based Explanations

In supervised machine learning a training set is used to fit a prediction model. Thus any prediction depends on these training samples.
The idea of the following study is to (i) identify the training compounds most relevant to a single prediction by taking the influence measures introduced in Section 4.3 into account, and then to (ii) visualize these compounds along with the corresponding predictions (see Schroeter [110]).
This approach allows to explain nonlinear classifiers to users in terms of chemical compounds—elements of their domain of expertise. Though the visual inspection of relevant compounds may allow for insights into the reasoning of prediction methods the beneficial effect of this approach is hard to quantify. The questionnaire study discussed in Sections 4.4 and 4.6 quantifies the impact of visual explanations in the context of Ames mutagenicity prediction. There, the visual information significantly improved the participants ability to identify unreliable predictions.

4.3 Measuring Influence in Predictions of Kernel-Based Models

In this section the influence of training compounds on kernel-based predictions is analyzed and exemplarily calculated for Gaussian process models.

The mean function (or prediction function) $\mu(\mathbf{x})$ of a Gaussian process model (Equation A.18) can be expressed as a linear combination of kernel function values[1]

$$\mu(\mathbf{x}) = \mathbf{k}_*^T \underbrace{(\mathbf{K} + \lambda \mathbf{I})^{-1}\mathbf{y}}_{\alpha} = \mathbf{k}_*^T \alpha = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}, \mathbf{x}_i). \tag{4.1}$$

In this representation we can associate each summand with the contribution of a training sample to the predicted value: The factor α_i comprises information about noise, labels and input features derived from the training data and the kernel function value $k(\mathbf{x}, \mathbf{x}_i)$ measures the similarity between the new input and the i-th training compound. The product of both reflects the desired contribution of the i-th training compound to the prediction.

For a large class of learning methods, i.e. SVMs and related methods[2], such sums of contributions can be calculated. The representer theorem[3] ensures for these methods that a representation of the prediction as a linear combination of training data exists.

Alternatively, only certain parts of the summands in Equation 4.1 are considered as contribution weights: The factors α_i for example are independent of the test data and represent a kind of general importance weights of the training samples. In sparse models, these contribution weights are set to zero for most of the training samples.
Another option is to reinterpret the mean function as a linear combination of labels

$$\mu(\mathbf{x}) = \underbrace{\mathbf{k}_*^T (\mathbf{K} + \lambda \mathbf{I})^{-1}}_{\beta(\mathbf{x})^T}\mathbf{y} = \sum_{i=1}^{n} \beta(\mathbf{x})_i \, y_i. \tag{4.2}$$

in order to define a weight function $\beta(\mathbf{x})$ which measures the contribution of each label y_i. This weight function[4] can be viewed as a nonlinear analogon of the leverage term in linear regression. It is of special interest in defective regression tasks where extreme label values of outliers may dominate the summands.

In this study we only consider classification tasks, hence the absolute values of the first option (whole summand) and the last option ($\beta(\mathbf{x})_i$) coincide. The weight vector for the normalized contribution of the training compounds is defined as

Definition 1

$$\hat{\beta} := \frac{\mathrm{abs}(\beta(\mathbf{x}))}{\sum_{i=1}^{n} \mid \beta(\mathbf{x})_i \mid}, \tag{4.3}$$

where $\mathrm{abs}(\beta(\mathbf{x}))$ denotes the component-wise absolute value of the vector $\beta(\mathbf{x})$. In the following, this normalized contribution is used to identify the most relevant compounds for single predictions.

[1] In order to calculate the most relevant compounds of the training set, we focus on the predictive mean μ used in classical GPs and disregard the subsequent transformation of the latent predictor applied in the case of GP classification (GPC), cf. Section A.2.

[2] In general these methods display a regularizer which is a nondecreasing function of the L_2 norm [3], see also methods in Table A.1.

[3] A representer theorem has first been stated by Kimeldorf and Wahba [63] in 1971. It has been extended by O'Sullivan et al. [90] and generalized by Schölkopf et al. [109]. For the definition used here and a discussion in the context of Gaussian processes see Rasmussen and Williams [97, Chapter 6].

[4] The weight function is sometimes also called an *equivalent kernel*, although it is not a kernel function in the strict sense of being positive definite, see also Rasmussen and Williams [97, Chapters 2.6 and 7.1].

4.4 Evaluation of Compound Relevance Weights

Recall that the main idea of this study is to visualize the most influential training compounds in order to explain predictions of kernel-based models in terms of chemical structures. The questionnaire study described in the following section was conducted to assess the impact of such visualizations.

Questionnaire Study on Mutagenicity Predictions

The questionnaire study evaluates the participants' ability to identify reliable predictions using our visual explanations. Participants are asked to judge the reliability of a prediction on Ames mutagenicity [2] generated by a Gaussian process model for classification (GPC model).

This task implies two potential sources of bias: First, each person holds an individual level of general trust in computer-based predictions, e.g., due to personal experiences. Second, showing additional information besides the predicted value might let the user judge the prediction differently. Therefore, participants are asked to decide between two contradicting predictions, i.e., one model predicts the compound in question to be "mutagenic", and the other one to be "non-mutagenic". In this forced design, there is no option to vote for or against machine predictions as such, eliminating the first source of bias. The prediction values ("mutagenic" / "non-mutagenic") are presented together with several explaining compounds. To measure the effect of these visual explanations' on participants' decision making, placebos are used: In 20 of the 40 test cases the visual explanations' are informative, while the remaining 20 cases are presented with non-informative "placebo explanations". Figure 4.2 presents an annotated screen shot of the questionnaire as presented to participants. All types of test samples are equally represented in the test set of 40 compounds (ten mutagenic compounds with informative explanations, ten mutagenic compounds with non-informative explanations, and two corresponding sets of ten non-mutagenic compounds).

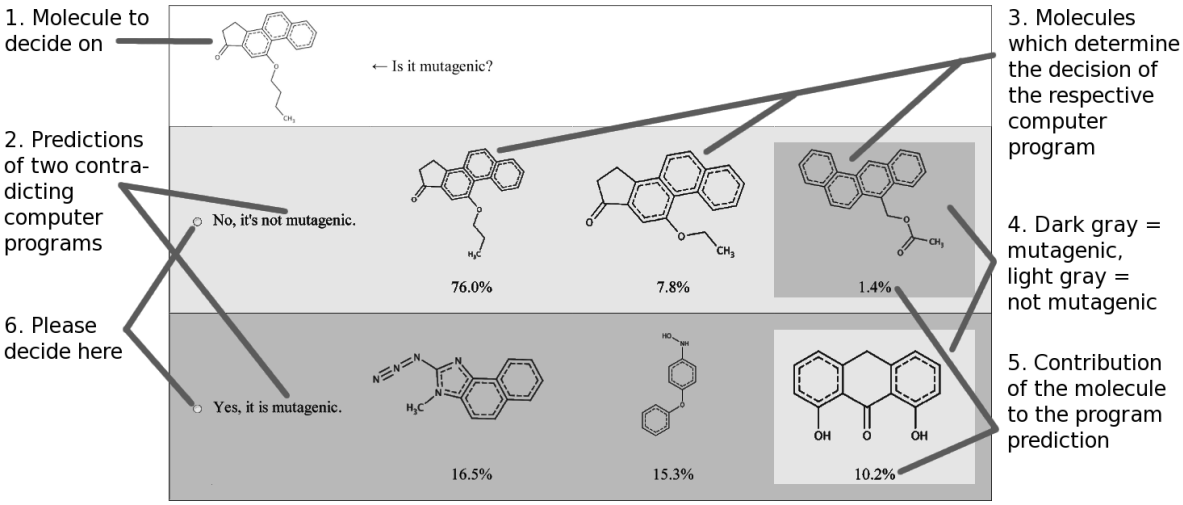


Figure 4.2: The elements of information in the questionnaire study for deciding on Ames mutagenicity: The compound in question and two contradicting GPC model predictions are shown. For each prediction, the three most relevant training compounds and their respective importance in percent is given. The background color indicates the classification: dark gray for mutagenic, and light gray for non-mutagenic. For the 20 non-informative "placebo explanations", the true explaining compounds are replaced by randomly selected training compounds of the same mutagenicity class.

Each test compound is presented in turn to the human participant. The presentation is permuted at random for every new user. Moreover, no feedback on correctness is given to participants[5]. For more details on study design, machine learning models, and datasets see Hansen et al. [44] .

To address different levels of chemical knowledge, participants were grouped according to profession into different levels of expertise: "pharmaceutical science" (included one of "toxicology"), "chemistry", "medicine", and "layman" (remaining disciplines). Since the decisions on the test compounds are assumed independent from each other and the order of presentation of the compounds is randomized for each new participant, every single decision of all humans is taken into account.

To assess the questionnaire study a multinomial logistic regression model [78] is fitted to the outcomes of the human decisions,

$$\mathbb{E}[g(y)|x] = \vartheta_0 + \sum_{i=1}^{k} \vartheta_i x_i + \epsilon. \tag{4.4}$$

The dependent variable y falls into one of the categories "hit" or "miss" for each participant on an individual test compound. There are five binary predictor variables $x_i, i \in \{1, \ldots, 5\}$ involved, namely "layman", "chemistry", "medicine", "pharmaceutical science", and "explanation". The first four specify the expertise level of the participant. The last one equals 1 for the decision on test compounds with the relevant training compounds shown (i.e., the true explanations), and 0 in the cases of random compounds (the "placebo explanations"). The parameters $\vartheta_i, i \in \{0, \ldots, 5\}$ are fitted to the experimental results by logistic regression using the logit function

$$y = \frac{1}{1 + e^{-(\vartheta_0 + \sum_1^k \vartheta_i x_i)}}. \tag{4.5}$$

For each level of expertise, the odds-ratios of making a correct decision are calculated in the presence of the true explanations as well as in the presence of random"placebo explanations". The odds-ratios are obtained from the logistic regression model parameters for each predictor variable $\frac{\mathbb{P}(\text{hit})}{\mathbb{P}(\text{miss})} = \exp(\vartheta_i), i \in \{1, \ldots, 5\}$. As a baseline, 10 unbiased random guesses of the label for each test compound are generated automatically. The odds-ratio gives the respective odds of making the correct decision in comparison to random guessing (the *null hypothesis*). The results are considered statistically significant if the probability to generate them under the null hypothesis is smaller than 5% (p-value < 0.05 using a likelihood ratio test).

Results

The decisions of 71 human participants were evaluated. Most of them were chemical scientists working in industry or PhD students at German universities. Table 4.1 summarizes the numbers of participants, grouped by their level of expertise.

[5]The questionnaire study is available online under `http://doc.ml.tu-berlin.de/toxpoll/` (in German only)

expertise	participants	finished	decisions
Layman	35	21	906
Chemistry	17	12	498
Medicine	3	2	83
Pharm. Sc.	16	15	606
Total	71	50	2093

Table 4.1: Participants grouped by expertise

The results of the human decision performance with informative explanations and with "placebo explanations" are listed in Table 4.2. The achieved odds-ratios of making a correct decision in comparison to the baseline of random guessing and the levels of significance are shown.

	Layman	Chemistry	Medicine	Pharm. Sc.	Explanation	Hits	Misses	Accuracy
	0	0	0	0	0	195	205	49%
	1	0	0	0	0	273	180	60%
	1	0	0	0	1	346	107	76%
	0	1	0	0	0	133	114	54%
	0	1	0	0	1	182	69	73%
	0	0	1	0	0	26	15	63%
	0	0	1	0	1	34	8	81%
	0	0	0	1	0	209	94	69%
	0	0	0	1	1	228	75	75%
odds-ratio	1.67	1.32	2.02	2.01	1.93			
p-value	0.056	0.23	0.14	**0.03**	**0.01**			

Table 4.2: Contingency of observed decision performance. First row refers to the random guessing baseline, the following to the questionnaire results of the participants grouped by expertise. Access to true explanations of the model predictions results in a 93% improvement of decision performance (red, significant p-value 0.01). Pharmacists showed a doubled chance to identify Ames mutagenicity of the test compounds correctly (blue, significant p-value 0.03).

There are two significant improvements in decision performance: Being a pharmacist and access to the explaining visualizations have the strongest and most significant impact. Being a pharmacist roughly doubled the chance to correctly identify Ames mutagenicity of the test compounds (p-value of 0.03). Seeing the most relevant training compounds as determined by our algorithm increases performance by 93% (p-value of 0.01).

Pharmacists, who start at the highest level of accuracy, still benefit from the explaining compounds (increase in accuracy from 69% to 75%). In summary, explaining the model classifications by visualizing the most relevant training compounds significantly improved decision performance. Even users with profound prior knowledge profited from the explanations.

4.5 Related Work

The visualization of training compounds most relevant for prediction is connected to various techniques and problems in chemoinformatics. We briefly discuss relations to techniques for interpretation of kernel-based models, sensitivity analysis, training set visualization, and confidence estimation.

Model interpretability is often discussed in conjunction with feature importance and selection [40]. In nonlinear modeling, feature selection methods are used to identify features which improve the overall prediction performance (e.g., Byvatov and Schneider [16]) or to extract the most relevant features for single predictions [17]. In Chapter 5 feature gradients are applied to estimate the local importance of single features. In contrast, the visual explanations of this chapter do not refer to individual features, but visualize complete compounds from the training set. The approach provides information on the objects that form the basis of the nonlinear classifier predictions. Thus, human experts are able to judge the prediction with their expertise about these objects holistically, possibly identifying and recruiting more or different features than those available to the model.

The rating of the elements in the training set is the conjoint necessity of the explanation approach and sensitivity analysis: In outlier sensitivity [41], the effects of removing single data points on estimated parameters are evaluated by an influence function. In regression problems, leverage analysis detects leverage points which have the potential to give a large impact on the estimate of the regression function. In contrast to outliers, removing a leverage sample may not actually change the predictor, if its response is very close to the predicted value; e.g., for linear regression the samples whose inputs are far from the mean are the leverage points.
These sensitivity analysis techniques examine the impact of data points on the model and do not detect the local trends covered by the visual explaining approach. However, the notion of relevance defined in Equation (4.2) can be interpreted as a kernelized version of leverage analysis: If the compound relevance vectors β are computed for the whole training set and arranged side by side, a hat matrix is generated and may be used for a nonlinear leverage analysis:

$$\hat{\mathbf{y}} = \mu(\mathbf{X}) = \underbrace{\mathbf{K}^T(\mathbf{K} + \lambda\mathbf{I})^{-1}}_{\mathbf{H}}\mathbf{y}. \tag{4.6}$$

To assess the reliability of model predictions, distance measures between training and test examples are commonly used [117]. Sushko et al. [124] evaluate several distance-based measures to estimate the domain of applicability of QSAR models, as well as more sophisticated approaches. The applicability of Gaussian process models can be estimated using their built-in estimate of predictive variance [114]. All of these approaches quantify the confidence in single predictions as a numeric value. In contrast, our approach does not provide a single number as confidence measure. It visualizes the specific molecules that determine a prediction on a per-compound basis and allows practitioners to take personal knowledge into account to assess the reliability of predictions.

In this work we address interpretability of kernel-based predictions and exemplarily illustrate our approach on Ames mutagenicity data. The point of our study is not to predict Ames mutagenicity with the best possible performance. There are more powerful tools available to estimate Ames mutagenicity [42] than the predictors used in our evaluation. Rather, we focus on and recreate a situation where not enough training data are available (e.g., predicting new chemical compound series). For this, we trained the two GPC models on a relatively small set of 1000 compounds to allow for contradicting predictions.

Principled visualization algorithms have proven useful to visualize multi-dimensional training sets [64, 77]. Additionally, there are interactive tools that allow to display and decompose chemical compounds in a set, as well as to graphically analyze their varying properties [65]. Our algorithm ranks the training compounds according to their relevance for prediction. The simple structural visualization used in the questionnaire might be improved by interactive tools for more detailed chemical analysis.

4.6 Discussion

Locality of explanation The width parameter σ of the used squared exponential kernel (also called radial basis function kernel Eq. A.11) determines the number of training compounds the model uses primarily to infer predictions for new test compounds. Cross-validation experiments were performed on the training set to determine an appropriate trade-off between model locality and model accuracy.

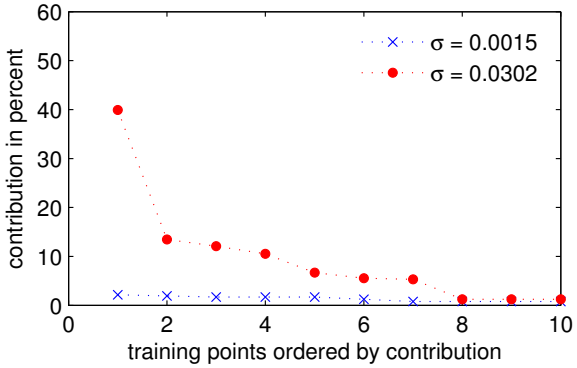


Figure 4.3: Contribution of the 10 most relevant training compounds to the prediction for a given test compound. For a width parameter of $\sigma = 0.0302$ the first seven compounds dominate the prediction. If the width parameter is too small ($\sigma = 0.0015$) all training compounds almost uniformly influence the predicted value by a small amount. (illustrated for GPC model 1, GPC model 2 behaves similarly).

Figure 4.3 shows how the width parameter σ influences the 10 training compounds most relevant for a given test compound. The chosen kernel width of $\sigma = 0.0302$ ensures that sufficiently local models are generated. This procedure is appropriate in this context because mutagenicity is an inherently local property.[6] When transferring the methodology to different datasets, it may be worth checking whether such vicinity dependent behavior dominates the problem in question.

Individual test cases The results of the questionnaire study indicate that a visual model explanation improves the judgment on model predictions significantly over the baseline. We discuss this observation on the basis of individual examples.

In the test setting, a participant is asked to judge the Ames mutagenicity of a compound based on his personal knowledge and the conflicting predictions of two models. Note that we consider only the most difficult cases here. The two GPC models agree on most evaluation compounds and predict their mutagenicity correctly. The test set used in the questionnaire study is a selection of compounds on which the classifiers disagree and yield less

[6] Local in the sense that the chemical features which determine mutagenicity vary locally in the chemical space, e.g. containing epoxides generally tends to make non-steroids mutagenic, but we do not observe this effect for steroids [6].

confident predictions[7]. These compounds may lie outside the domain of applicability of at least one classifier.

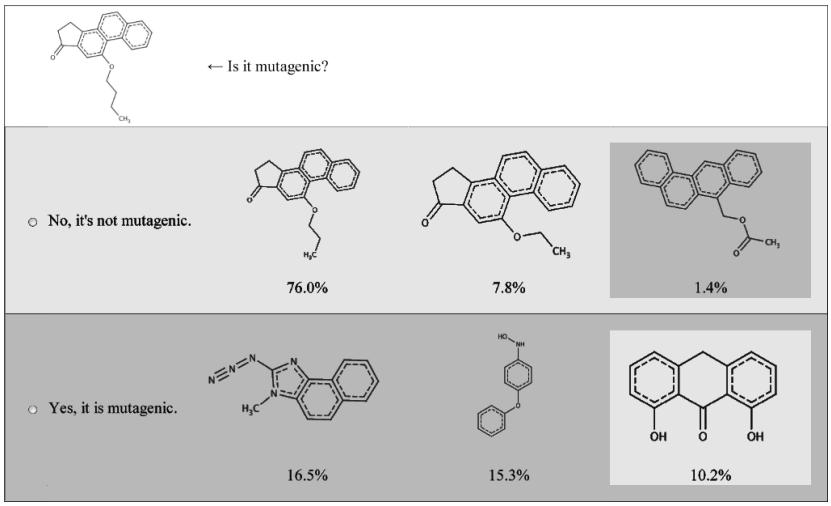


Figure 4.4: The non-mutagenic test compound 15,16-dihydro-11-n-butoxy-cyclopenta(a)phenanthren-17-one from the questionnaire study.

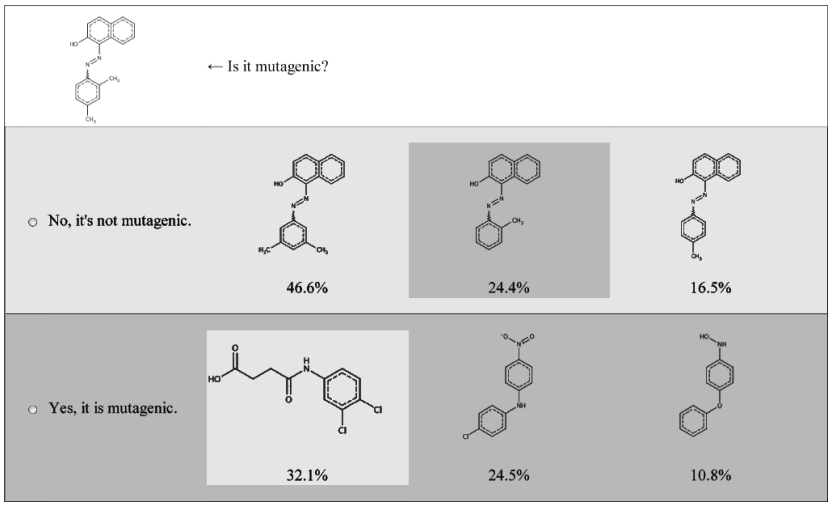


Figure 4.5: Mutagenic azo dye sudan red from the questionnaire study.

Figures 4.4 and 4.5 illustrate the behavior of the explaining compounds in such situations: In Figure 4.4, the second classifier displays a peculiar mixture of compounds barely related to the compound in question. Moreover, the percental impact of these compounds (given below the structures) is small, indicating a low relation of the test compound to most training compounds. The explaining visualization thus indicates a misclassification of the second classifier due to insufficient coverage of the training data. In Figure 4.5, the second classifier also operates out of its domain of applicability.

However, the assessment of the first classifier is more difficult: On the one hand, the first explaining compound equals the compound in question except for the 2,4 (meta/meta) methyl-substitution instead of a 1,3 (ortho/para) methyl-substitution at the lower ring, and is non-mutagenic. On the other hand, the second explaining compound is also very similar (meta-

[7]The Gaussian process classifier outputs for each compound the probability of being in the positive class. Outputs around 0.5 are considered as less confident.

but not para-substituted), but shows a different mutagenicity. In this case the explaining compounds indicate a chemically complex situation, such as activity cliffs. Therefore, the given explanation can serve as a red flag and further laboratory experiments or literature research are advisable to determine mutagenicity. While the prediction itself is unreliable, the explaining compounds reveal valuable information and ideas for optimization, e.g. they indicate a strong effect of the meta-methyl group position at the lower ring on mutagenicity.

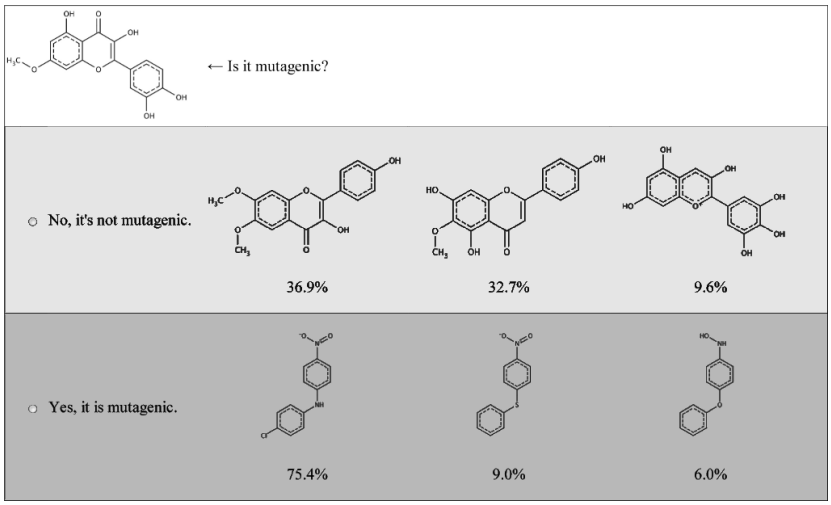


Figure 4.6: The mutagenic test compound rhamnetin from the questionnaire study.

After demonstrating the advantages of visual explanations we investigate possible shortcomings of our approach. Since the information captured in the training set is not complete, it is possible that the classifier makes a prediction that is reasonable from its limited perspective, but wrong. We found only one example for this situation, the mutagenic rhamnetin [21] (Figure 4.6). The second classifier refers to a different class of compounds, and is therefore neglected. The other classifier displays reasonable examples: The first two compounds (hortensin and hispidulin), both flavonoids like the compound in question, show a high similarity, and are consistently marked as “non-mutagenic”. The classification as “non-mutagenic” seems plausible (all pharmacists in the poll voted for “non-mutagenic”), but the explaining compounds are rather misleading. In this case the complex behavior of flavonoids is not reflected in the training set. In such a situation, we recommend to verify the relevant training data in order to exclude errors, and to update the model afterwards.

Finally, the users expertise influences the impact of the visual approach. In Figure 4.7, both sets of explaining compounds look equally plausible at first sight. However, the pharmacists more often correctly assessed the high influence of the aromatic nitrogen substitution on the mutagenicity (see upper row of explaining compounds), and voted correctly for the lower predictor. The laymen in contrast performed poorly. This example shows that visual explanations can not turn the layman into an expert, but can support the expert with valuable information. In summary, the new visual explanations enhance interpretation of nonlinear prediction models. We have used GPs as one possible example of nonlinear models; the techniques can be transferred to other nonlinear models in a straight forward manner. Visual explanations may support practitioners in their work with machine learning models by detecting and resolving weaknesses of the models, as well as discovering important characteristics of the training data.

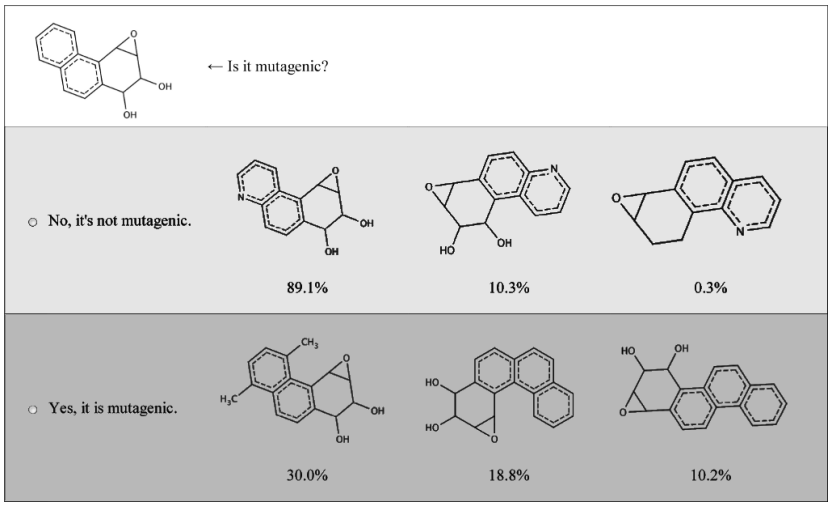


Figure 4.7: The mutagenic test compound anti-phenanthrene-1,2-diol-3,4-epoxide from the questionnaire study.

4.7 Conclusions

In this chapter a new approach to interpret kernel-based predictors was investigated. The method measures the influence of individual training compounds (explaining components) on single model predictions. The impact of finding and showing these explaining components was confirmed by a questionnaire study of human decision performance on Ames mutagenicity classification. By applying the general representer theorem [109], such explanations can be given for a large class of nonlinear machine learning methods, including e.g. Gaussian processes, support vector machines, kernel partial least squares and kernel ridge regression. The questionnaire study employed Gaussian processes with the squared exponential kernel. The analysis of other kernel methods and kernels are candidates for further research. Access to the molecular evidence behind a prediction facilitates understanding of the predictions validity in a given context.

The examination of individual test cases suggests that the provided explanation enables human experts to make an informed decision on difficult instances, where automatic confidence estimation might fail. Looking into the explaining components of individual predictions may prove useful to spot wrong labeling and insufficient coverage of specific regions in large complex datasets. Thus visualizing explaining components could be applied as a generic tool of quality assurance to validate nonlinear in silico model predictions for specific application areas. Understanding nonlinear predictors as such furthermore allows not only to assess the domain of applicability, it may also provide general ideas on enhancements of machine learning approaches, e.g., in the field of kernel design and descriptor generation.

Chapter 5

Interpretation in Terms of Local Feature Importance

In the previous chapter (Section 4.1) we already discussed the importance and different facets of interpretability in chemoinformatics. In this chapter single predictions are not interpreted in terms of chemical compounds but in terms of physicochemical features. We analyze which features of the compound in question influence the prediction applying local gradients. Again Gaussian processes are used to illustrate the approach.

5.1 Methods

Definition of Explanation Vectors

We first consider a *regression* model $f(\mathbf{x})$ learned from examples $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\} \in \mathbb{R}^d \times \mathbb{R}$ with f first-order differentiable w.r.t. $\mathbf{x}$ [1]. This situation is pictured for two-dimensional inputs in Figure 5.1: The x_1- and x_2-axes describe the chemical space of compounds, the vertical axis represents the experimentally measured property of interest y (label), and the learned prediction function $f(x)$ is shown as a wavy surface.

In order to investigate the importance of input features we consider the local gradients as *explanation vectors*. The local gradient of a data point $\mathbf{x}_0$ is defined to be the derivative of f with respect to $\mathbf{x}$ at $\mathbf{x} = \mathbf{x}_0$, or formally,

$$\eta_f(x_0) := \nabla f(x)|_{x=x_0}. \tag{5.1}$$

Note that $\eta(\mathbf{x}_0)$ is a d-dimensional vector just like $\mathbf{x}_0$ and points towards the steepest ascent of the prediction function f. Thus the sign of each of its individual entries indicates whether the prediction would increase or decrease when the corresponding feature of $\mathbf{x}_0$ is increased locally and each entry's absolute value gives the amount of influence in the change in prediction. Figure 5.1 illustrates how the importance of features may change for different inputs.

The vector $\eta(\mathbf{x}_0)$ defines a vector field over the chemical input space that characterizes the flow towards the function's maximum in each point. This maximum is not necessarily the global maximum since the gradient denotes the strongest improvement around $\mathbf{x}_0$. As long as the model f is not wiggly due to over-fitting, the explanation vector captures trends within small regions of the chemical space, e.g., compound classes.

[1] Most kernel-methods are first-order differentiable; an approximation techniques for kernels that are not first-order differentiable is discussed in Section 5.1)

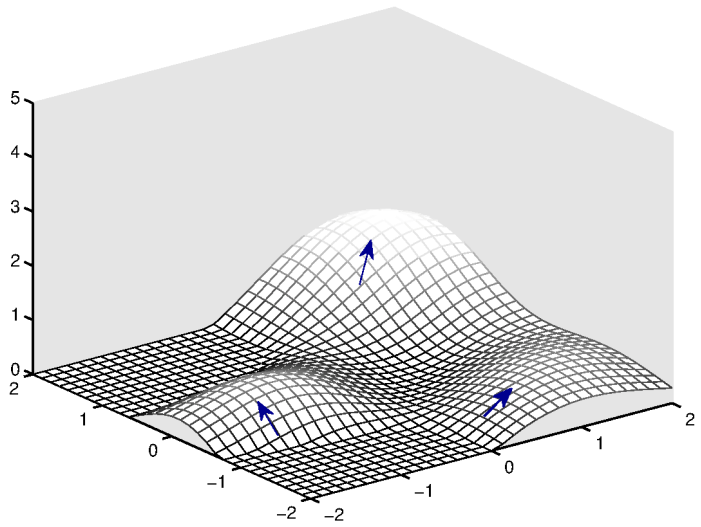


Figure 5.1: Sketch of local gradients for feature importance: The x_1- and x_2-axes describe the chemical space of compounds, the vertical axis represents the experimentally measured property of interest y (label), and the learned prediction function $f(x)$ is sketched as a wavy surface. Following the local gradients (direction of blue arrows) in order to increase the predicted value requires either a modification of only a single property (x_1 or x_2) or a joint adjustment of both.

In case of binary *classification* we define local explanation vectors in an analog manner as local gradients of the probability function $p(\mathbf{x})$ of the learned model for the positive class:

$$\eta_p(\mathbf{x}_0) := \nabla p(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_0} \qquad \text{with} \qquad p(\mathbf{x}) = P(Y = 1 \mid X = \mathbf{x}). \tag{5.2}$$

Here $p : \mathbb{R}^d \to [0, 1]$ refers to the probability function of a classification model learned from examples $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\} \in \mathbb{R}^d \times \{-1, +1\}$. In this section we discuss alternatives that are applicable if this function is not available for a given classification method. By Definition (5.2) the explanation vector η is a d-dimensional vector that gives the direction of the steepest ascent from the test point to higher probabilities for the positive class. The negative version $-\eta_p(\mathbf{x}_0)$ indicates the changes in features needed to increase the probability for the negative class which may be especially useful for $\mathbf{x}_0$ predicted to be in the positive class.

We remark that $\eta(\mathbf{x}_0)$ becomes a zero vector, for example, when p is equal to a constant in some neighborhood of $\mathbf{x}_0$. The explanation vector fits well to classifiers where the probability function $P(Y = 1 \mid X = \mathbf{x})$ is usually not completely flat in some regions. In the case of deterministic classifiers, despite of this issue, Parzen window estimators with appropriate widths (Section 5.1) can provide meaningful explanation vectors for many samples in practice.

For an example we apply the Definition 5.2 to model predictions learned by Gaussian process classification (GPC). GPC does model the class probability function p directly. For other classification methods, such as support vector machines, that do not provide a probability function as its output, we give an example for an estimation method later in this section.
The local gradients of the GPC probability function can be calculated analytically for differentiable kernels as we discuss next.

In the case of the probit likelihood term defined by the error function, the probability for being of the positive class $p(\mathbf{x})$ is defined for GPC in equation A.19. The derivation of this

function with respect to $\mathbf{x}$ at $\mathbf{x} = \mathbf{x}_0$ yields the local gradient

$$\nabla p(x)|_{\mathbf{x}=\mathbf{x}_0} = \frac{\exp\left(\frac{-\mu(\mathbf{x}_0)^2}{2(1+\mathrm{var}(\mathbf{x}_0))}\right)}{\sqrt{2\pi}} \left(\frac{\nabla\mu(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_0}}{\sqrt{1+\mathrm{var}_f(\mathbf{x}_0)}} - \frac{1}{2}\frac{\mu(\mathbf{x}_0)}{(1+\mathrm{var}(\mathbf{x}_0))^{\frac{3}{2}}} \nabla\mathrm{var}(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_0} \right). \quad (5.3)$$

As a kernel function choose, for example, the radial basis function (RBF) kernel $k(\mathbf{x}_0, \mathbf{x}_1) = e^{-w\,\|\mathbf{x}_0-\mathbf{x}_1\|^2}$, which has the derivative

$$\frac{\partial}{\partial \mathbf{x}_{0,j}} k(\mathbf{x}_0, \mathbf{x}_1) = -2w \;\; e^{-w\,\|\mathbf{x}_0-\mathbf{x}_1\|^2} \;\; (\mathbf{x}_{0,j} - \mathbf{x}_{1,j}) \qquad \text{for } \; j \in \{1, \ldots, d\}.$$

Then the elements of the local gradient $\nabla\mu(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_0}$ are

$$\frac{\partial \mu}{\partial \mathbf{x}_{0,j}} = -2w \sum_{i=1}^{n} \alpha_i e^{-w(\mathbf{x}_0-\mathbf{x}_i)^2} \;\; (\mathbf{x}_{0,j} - \mathbf{x}_{i,j}) \qquad \text{for} \;\; j \in \{1, \ldots, d\}.$$

Figure 5.2 illustrates the local gradients of this GPC model on a toy example.

In summary explanation vectors let us *locally* understand the the prediction function of classification and regression problems.

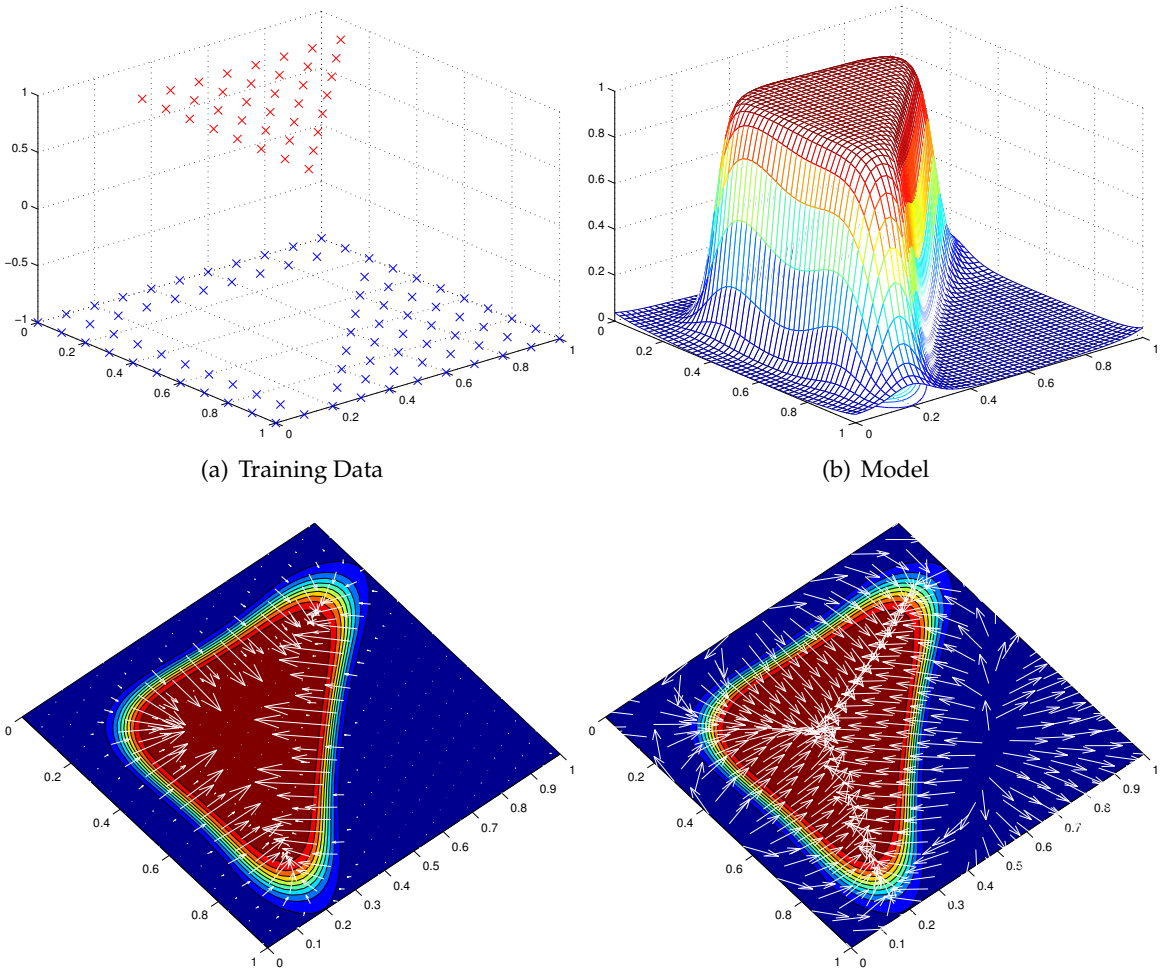
Estimating Explanation Vectors

Several *classification* methods directly estimate the decision rule, which often has no interpretation as a probability function. For example decision trees provide the class label directly and support vector machines [138, 107, 82] estimate the distances to a high-dimensional hyperplane. In the following, we will explain how explanations can be obtained for such classifiers.

Suppose we learned a classification function g that assigns a class label $c \in -1, 1$ to each compound vector $\mathbf{x}$. For test data points $\mathbf{z}_1, \ldots, \mathbf{z}_m \in \mathbb{R}^d$ which are assumed to be sampled from the same unknown distribution as the training data, the function g estimates labels $g(\mathbf{z}_1), \ldots, g(\mathbf{z}_m)$. Now, instead of trying to explain the classifier g, we suggest to approximate g by another classifier $\hat{g}$, the actual form of which resembles a Bayes classifier with a probability function.

There are several choices for $\hat{g}$, for example, GPC, logistic regression, and Parzen windows.[2]

[2]For the special case of support vector machines Platt [95] a sigmoid function is fitted to map the outputs on probabilities. The approximation of g by methods like Parzen windows, as introduced in this study, defines a more general approach for estimating explanation vectors [6].



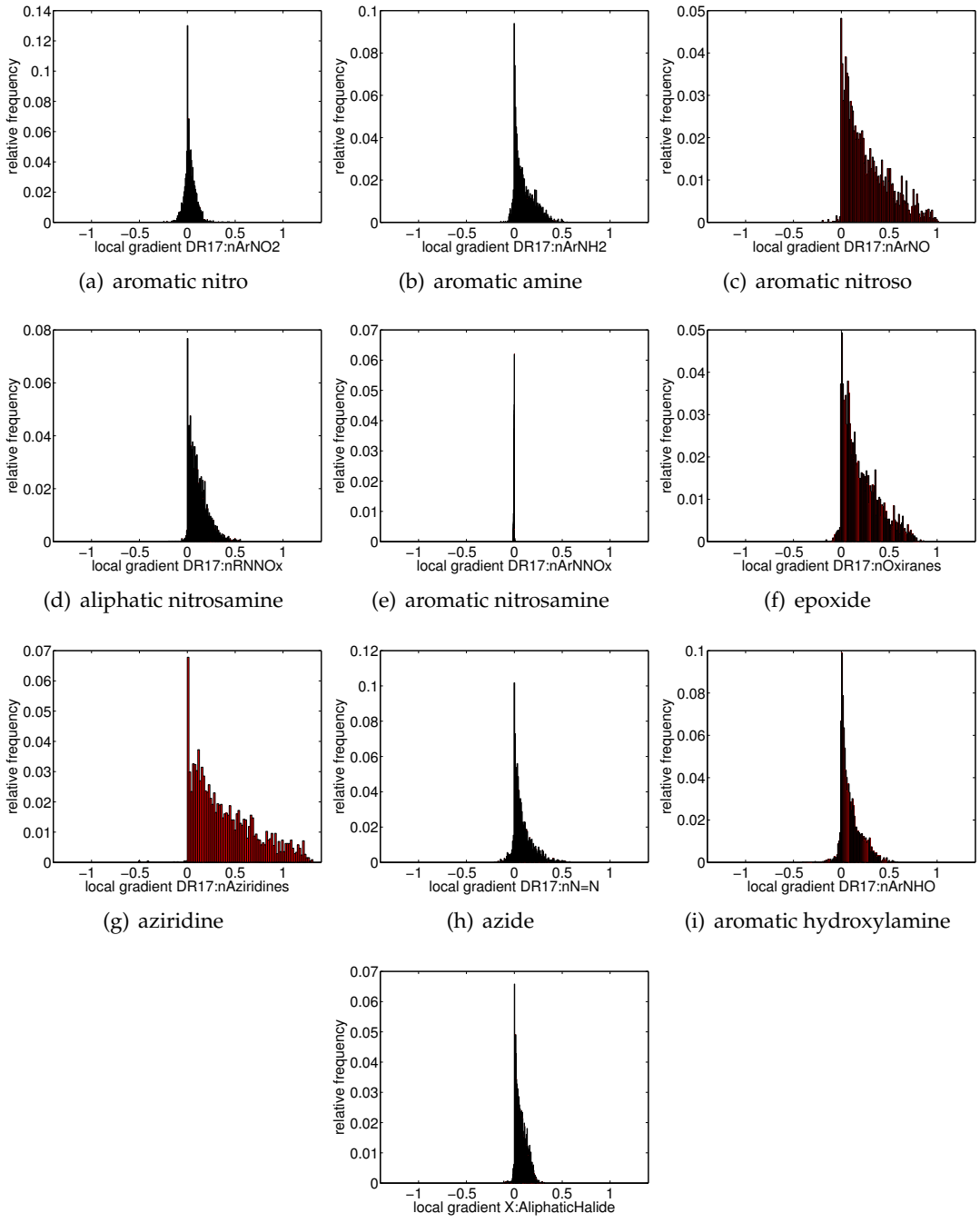
(a) Training Data (b) Model

(c) Local explanation vectors (d) Direction of explanation vectors

Figure 5.2: Explaining object classification with local gradients: (a) training data of a simple object classification task, blue points are labeled -1 and red points $+1$; (b) probability function for the positive class learned using GPC; (c,d) local gradient explanation vectors of the model together with the elevation profile of the probability function; in panel (d) vectors are normalized. (c) Along the hypotenuse and at the corners of the triangle explanations from both features interact towards the triangle class while along the edges the importance of one of the two feature dimensions dominates. At the transition from the negative to the positive class the length of the local gradient vectors represents the increased importance of the relevant features. In panel (d) we see that explanations close to the edges of the plot (especially in the right hand side corner) point away from the positive class. However, panel (c) shows that their magnitude is very small.

The approach has the advantage that we can use our estimated classifier g to generate any amount of labeled data for constructing $\hat{g}$. Even in high dimensions the classifier g can be exactly resembled. For the classifier $\hat{g}$ the explanation vectors are calculated as described previously and interpreted as explanations for the original classifier g.

In an analog manner a non-differentiable *regression* function f can be locally imitated by a first-order differentiable regression function $\hat{f}$ to estimate the explanation vectors.



(a) aromatic nitro (b) aromatic amine (c) aromatic nitroso

(d) aliphatic nitrosamine (e) aromatic nitrosamine (f) epoxide

(g) aziridine (h) azide (i) aromatic hydroxylamine

(j) aliphatic halide

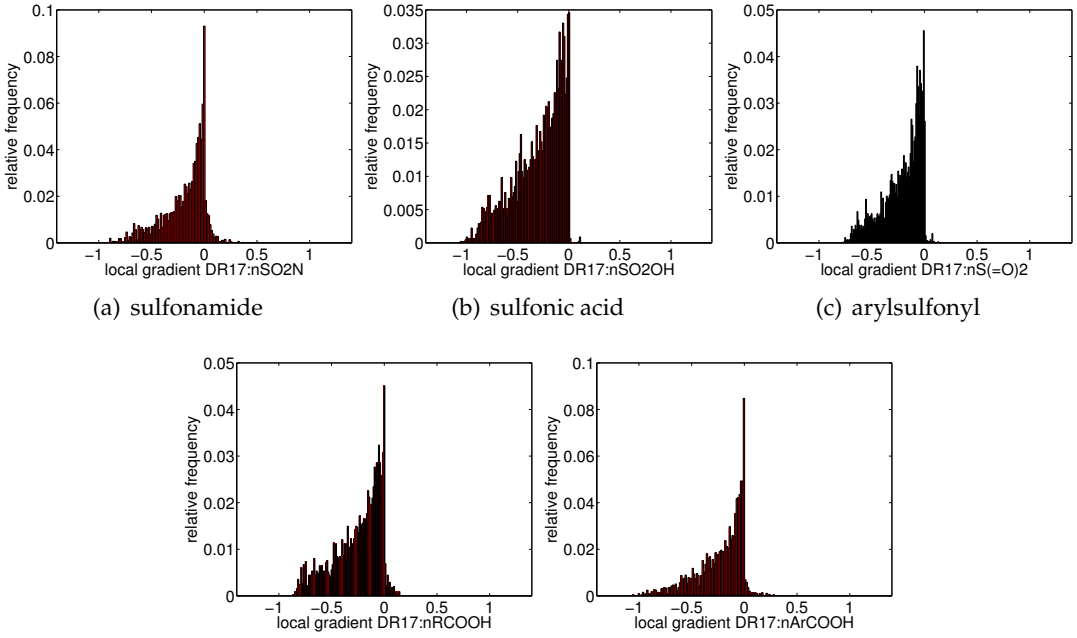
Figure 5.3: Distribution of local importance of selected features across the test set of 4512 compounds. Nine out of ten known toxicophores [61] exhibit positive local gradients. This indicates a mutagenic effect of toxicophores learned in the GPC model.

5.2 Evaluation of Explanation Vectors

In the following we investigate the task of predicting Ames mutagenic activity using the local gradient explanation methodology. The aim of this analysis is to find structure specific to the problem domain that has *not* been fed into training explicitly but is captured implicitly by the GPC model.

The GPC model is generated as follows: Each compound of the Ames dataset is represented by a vector of 142 molecular descriptors (counts of molecular substructures calculated using the DRAGON software [132]). The 6512 data points are randomly split into 2000 training and 4512 test examples such that the two classes (mutagenic and non-mutagenic) are balanced within the training set (stratified cross-validation). Additionally, the balance of steroid compounds in the train and test set is enforced. Ten additional random data splits are investigated separately to confirm the results presented below. Training and test set are normalized using the mean and variance of the training set before a GPC model with RBF kernel is trained. Finally, the explanation vector for each test point is calculated together with the prediction. The performance of the GPC models on the test points averaged out at 84 % area under curve (which equals the performance level reached in previous studies of GPC models for Ames mutagenicity [42]). The remainder of this section is an evaluation of the calculated local explanations.

The distribution of the local importance of single features is analyzed in Figures 5.3 and 5.4: For each input feature a histogram of local importance values is generated taking the corresponding entry in the explanation vector of each of the 4512 test compounds into account. The features examined in Figure 5.3 are counts of substructures known to cause mutagenicity. The figure shows all approved "specific toxicophores" introduced by Kazius et al. [61] that are also present in the DRAGON set of features. With the exception of 5.3(e) all histograms picture a high frequency of positive importance values and almost no negatives. Thus these toxicophores also have a toxifying influence according to the GPC prediction model. Feature 5.3(e) seems to be mostly irrelevant for the prediction of the GPC model on the test points[3].



(a) sulfonamide (b) sulfonic acid (c) arylsulfonyl

(d) aliphatic carboxylic acid (e) aromatic carboxylic acid

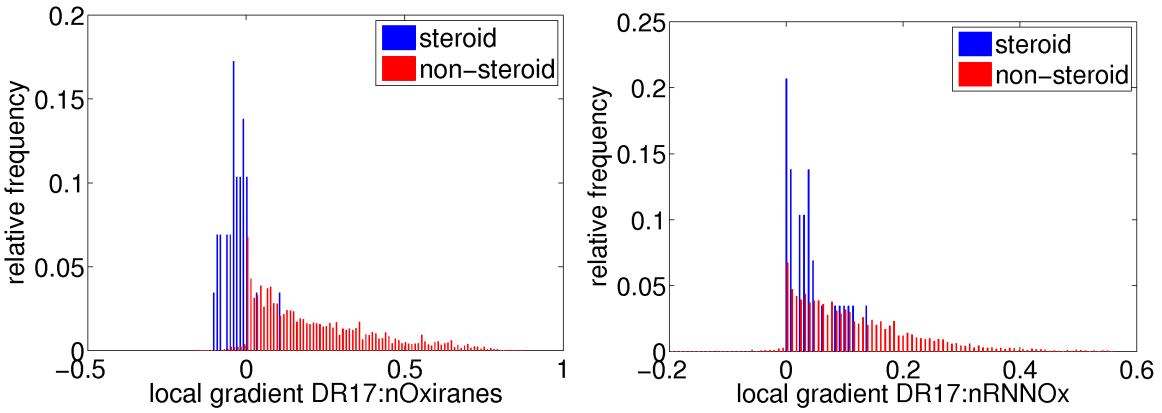
Figure 5.4: Distribution of local importance of selected features across the test set of 4512 compounds. All five known detoxicophores exhibit negative local gradients which corresponds to an non-mutagenic effect within the GPC model.

[3]We found that only very few compounds with this feature are present in the dataset. Consequently, detection of this feature is only possible if enough of these few compounds are included in the training data. This was not the case in the random split used to produce the results presented above.

In contrast the substructures known to detoxify certain toxicophores (detoxicophores [61]) have a negative influence on the prediction outcome of the GPC model (see Figure 5.4).

Compared to the remaining features the detoxicophores and toxicophores are among the features with the strongest effect on the GPC model. Hence the knowledge about toxicophores and detoxicophores is not only confirmed but also *(re)discovered* exclusively from the analysis of explanation vectors.

So far the conclusions drawn from the explanation vectors refer to global trends in the dataset. In the following paragraph we discuss steroids[4] as an example of an important compound class for which the meaning of features differs from this global trend, so that local explanation vectors are needed to correctly identify relevant features.



(a) epoxide feature: steroid vs. non-steroid

(b) aliphatic nitrosamine feature: steroid vs. non-steroid

Figure 5.5: The local distribution of feature importance differs significantly between steroids and non-steroid compounds for two known toxicophores (epoxide and aliphatic nitrosamine): The small local gradients found for the steroids (shown in blue) indicate that the presence of each toxicophore is irrelevant to the molecules toxicity. For non-steroids (shown in red) the known toxicophores exhibit larger positive local gradients.

Figure 5.5 displays the difference in relevance of epoxide (a) and aliphatic nitrosamine (b) substructures for the predicted mutagenicity of steroids and non-steroid compounds. In contrast to the non-steroid compounds, almost all epoxide containing steroids do not follow the global distribution (see Figure 5.3(f)) and exhibit gradients just below zero. The difference between these two distributions is confirmed by a p-value below 0.005 of the corresponding Kolmogorov-Smirnoff (KS) test[5]. This "immunity" of steroids to the epoxide toxicophore is an established fact and has first been discussed by Glatt et al. [37]. For aliphatic nitrosamine, the situation in the GPC model is less clear but still the toxifying influence seems to be less in steroids than in many other compounds (p-value of KS test below 0.005). To our knowledge, this phenomenon has not yet been discussed in the pharmaceutical literature.

In conclusion, we can learn from the explanation vectors that:

[4]Steroids are natural products and occur in humans, animals, and plants. They have a characteristic backbone containing four fused carbon-rings. Many hormones important to the development of the human body are steroids, including androgens, estrogens, progestagens, cholesterol and natural anabolics. These have been used as starting points for the development of many different drugs, including the most reliable contraceptives currently on the market.

[5]The Kolmogorov-Smirnoff test gives the probability of error when rejecting the null hypothesis that both relative frequencies are drawn from the same underlying distribution.

- Toxicophores tend to make compounds mutagenic.
- Detoxicophores tend to make compounds non-mutagenic.
- Steroids are immune to the presence of some toxicophores (epoxide, possibly also aliphatic nitrosamine).

From a methodological point of view the analysis exhibits that:

- Local explanations can reveal the global importance of input features.
- Local explanations can capture local trends within the dataset.

5.3 Related Work

Assigning potentially different explanations to individual data points distinguishes our approach from conventional feature extraction methods. Most feature extraction methods extract global features that are relevant for all data points, that is, those features that allow to achieve a small overall prediction error [40]. In this work the notion of explanation is not related to the prediction error, but only to the label provided by the prediction algorithm. Even if the error is large, the new framework is able to answer the question *why* the algorithm has decided on a data point the way it did.

In recent decades, explanation of results by expert systems has been an important topic in the artificial intelligence community. Especially for expert systems based on Bayesian belief networks, such explanation is crucial in practical use. In this context sensitivity analysis has also been used as a guiding principle [51]. There the influence is evaluated by removing a set of variables (features) from the evidence and the explanation is constructed from those variables that affect inference (relevant variables). For example, Suermondt [122] measures the cost of omitting a single feature E_i by the cross-entropy

$$H^{-}(E_i) = H(p(D|E); P(D|E\backslash E_i)\,) = \sum_{j=1}^{N} P(d_j|E) \log \frac{P(d_j|E)}{p(d_j|E\backslash E_i)},$$

where E denotes the evidence and $D = (d_1, \ldots, d_N)^T$ is the target variable. The cost of a subset $F \subset E$ can be defined similarly. This line of research is more connected to our work, because explanation can depend on the assigned values of the evidence E, and is thus local.

Similarly Robnik-Šikonja and Kononenko [98] and Štrumbelj and Kononenko [141] try to explain the decision of trained k-nearest neighbor models, SVMs, and artificial neural networks for individual instances by measuring the difference in their prediction with sets of features omitted. The cost of omitting features is evaluated as the information difference, the log-odds ratio, or the difference of probabilities between the model with knowledge about all features and with omissions, respectively. To know what the prediction would be without the knowledge of a certain feature the model is retrained for every choice of features whose influence is to be explained. To save the time of combinatorial training Robnik-Šikonja and Kononenko [98] propose to use neutral values which have to be estimated by a known prior distribution of all possible parameter values. As a theoretical framework for considering feature interactions, Štrumbelj and Kononenko [141] propose to calculate the differences between model predictions for every choice of feature subset.

Byvatov and Schneider [16] improved SVM performance by selecting features based solely on the set of support vectors and their gradients. For multi-layer perceptrons Féraud and Clérot [31] measure the importance of individual input variables on clusters of test points.

Therefore the change in the model output is evaluated for the change of a single input variable in a chosen interval while all other input variables are fixed. Lemaire and Féraud [71] use a similar approach on an instance by instance basis. By considering each input variable in turn there is no way to measure input feature interactions on the model output (see LeCun et al. [70]).

The principal differences between our approach and these frameworks are: (i) We consider continuous features and no structure among them is required, while some other frameworks start from binary features and may require discretization steps with the need to estimate parameters for it. (ii) We allow changes in any direction, that is, any weighted combination of variables, while other approaches only consider one feature at a time or the omission of a set of variables.

5.4 Discussion

We observed that explanation vectors are useful in a variety of situations. In the following we discuss their limitations.

Dealing with the Zero Derivatives

Where multiple clusters of compounds from the positive and negative class interfere, the decision function exhibits local minima and maxima. In contrast, broad clusters of a single class lead to flat regions in the decision function. Both local minima as well as flat regions lead to zero gradients. In the first case the Hessian matrix (the second derivative of the decision function) can be used to find "interesting" directions. The eigenvector corresponding to the largest eigenvalue of the Hessian matrix points towards the direction of the largest curvature and causes strong changes in the decision function. In flat regions gradient and Hessian will be zero and no meaningful explanation can be obtained by the gradient-based approach. Practically, by using Parzen window estimators with larger widths, the explanation vector can capture coarse structures of the classifier and give meaningful gradients in these regions.

Implicit Limitations of Analytical Gradients

Far from the training data, GPC models always predict a probability of 0.5 for the positive class. When one approaches the boundaries of the space populated with training data in an area of negative prediction values, explanation vectors will point away from any training data and therefore also away from areas of positive prediction. This behavior can be observed in Figure 5.2(d), where unit length vectors indicate the direction of explanation vectors. In the right hand side corner, arrows point away from the triangle. However, we can see that the length of these vectors is so small that they are not even visible in Figure 5.2(c). Moreover, the predictive variance of the GPC model is high at the boundaries of the space populated with training data and can be used to detect these regions. Consequently, this property of GPC models does not pose a restriction for identifying the locally most influential features using local gradients.

5.5 Conclusions

In this chapter a method to explain local decisions taken by arbitrary (possibly) nonlinear prediction algorithms was introduced. The estimated explanations are local gradients that characterize how a data point has to be moved to change its predicted label. For models where such gradient information cannot be calculated explicitly, we employ a probabilistic approximate mimic of the learning machine to be explained.

To validate the methodology we applied the new method to a challenging drug discovery problem. The results on that data fully agree with existing domain knowledge, which was not available to the method. Even local peculiarities in chemical space (the extraordinary behavior of steroids) were discovered using the local explanations given by our approach.

Chapter 6

Learning Transition States

6.1 Introduction

The various methods in computational chemistry operate on different length scales starting from atom vibrations to macromolecular interactions [58]. To overcome the gaps between these distinct levels of modeling remains one of the great challenges in computational chemistry.

Transition State Theory (TST) is a semi-classical approach which addresses the gap between classical molecular dynamics (MD) simulations and the rare events of chemical reactions observable on a microscopic level [30, 147, 62]. Assuming that nuclei are infinitely heavier than electrons the Born-Oppenheimer approximation allows to view a chemical reaction as nuclei moving on a potential energy surface [58]. Based on this approximation we can associate each configuration of a chemical system in phase space (position and momentum of each nucleus) with a potential energy value. Neglecting the momentum, as most often done in TST, the potential energy of a N-atom system can be viewed as a $3N$-dimensional surface. Figure 6.1(a) illustrates the relevant elements of a *potential energy surface (PES)* for a 2-dimensional system space: A product or reactant corresponds to a minimum on the PES and reactive trajectories refer to paths which connect the product with the reactant basin by following the surface gradient in each point. The easiest and most likely path from one minimum two another is along the reaction path or minimum energy path. The highest point along a minimum energy path is a saddle point and almost all reactive trajectory have to pass through the so called *bottle neck regions* very close around these saddle points. In literature, especially older textbooks, the system configuration in a saddle point is often called transition state. More precisely, the *transition state (TS)* of a system with N atoms is defined as the 3N-1 dimensional surface separating reactants and products—even though the surface sometimes appears lower dimensional if it is perpendicular to some degrees of freedom.

One intention of transition state theory is to estimate *reaction rates* using statistical mechanics. For a given transition state hypersurface the rate of reaction is approximated by the equilibrium flux out of this TS as

$$k_{\mathrm{TST}} = \frac{1}{2} \left\langle \delta(x - x^*) |\bar{v}| \right\rangle_R , \tag{6.1}$$

where $\langle ... \rangle_R$ is a Boltzmann average over the reactant region, $x = x^*$ is the location of the TS surface and $\bar{v}$ is the average velocity through it. The k_{TST} calculated according to 6.1 only equals the true reaction rate if

- every reactive trajectory passes the TS only once and
- every point on the TS dividing surface leads to a single reactive trajectory.

Violation of any of these assumptions leads to an overestimation of the true rate. In any complex system, however, finding the exact TS is a difficult problem. Even for systems in which the reaction mechanisms are known, an analytic expression of the TS surface can be intractable. Given a suboptimal surface not all crossing points will lead to reactive trajectories, and reactive trajectories may also re-cross the surface especially in high dimensional or high friction systems. Given that there are at least as many crossing points as reactive trajectories, the TST rate is always an overestimation of (or equal to) the true rate:

$$k_{\mathrm{True}} = \kappa \, k_{\mathrm{TST}} \qquad \kappa \in [0,1], \tag{6.2}$$

where the transmission coefficient, $\kappa \in [0,1]$, represents the fraction of trajectories that originate in the reactant basin and arrive in a product basin. The relation 6.2 is helpful for improving TS surfaces, for any dividing surface can be variationally optimized to minimize the TST rate and approach the true rate.

The choice of a dividing surface can be nontrivial even for simple systems and there is no common approach to this problem. In general, *prior information* on the reaction mechanism is used to approximate the TS surface for each specific chemical system. If, for example, an atom is being transferred from a donor to an acceptor, the TS would then be defined by a particular value of the distances from the transferring atom to the donor and acceptor. In this case the reaction mechanism is simple enough that a geometric quantity can be determined which quantifies the reaction progress. If motion of other atoms in the molecule were important for the reaction, this TST rate would be a poor approximation of the true rate. Hence, a major challenge in TST is to find good dividing surfaces in high-dimensional systems [60].

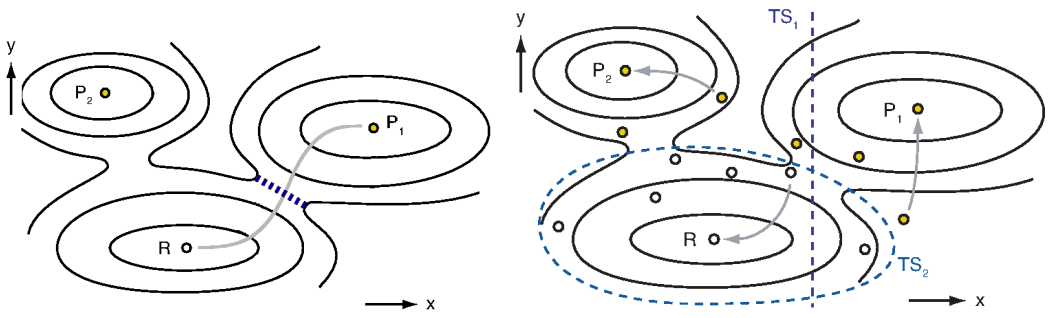


Figure 6.1: The contour lines indicate a potential energy surface in a 2-dimensional system. The gray line in (a) represents the run of a reactive trajectory from the reactant R to a product basin P_1 passing through the bottleneck region at the crest. In (b) the transition state TS_1, placed along an assumed reaction coordinate x, separates reactant R and product P_1 but fails to describe the transition to P_2. TS_2 is a surface which can be determined by training a SVM to distinguish a set of points as reactant or product.

In this chapter we introduce *machine learning techniques* to transition state theory in order to identify the TS surface in a fast and accurate way, without requiring detailed knowledge about the underlying reaction mechanisms.

6.2 Transition State Surface Estimation via Binary Classification

Let us describe the different states of a system consisting of n atoms by the $3n$-dimensional vectors $\mathbf{x}_1, \dots \mathbf{x}_N \in \mathbb{R}^{3n}$, where each $\mathbf{x}_i$ denotes a row vector of the Cartesian coordinates of all atoms or nuclei in the system. For a given set of such system states, each system state $\mathbf{x}_i$ is assigned to one of the classes products or reactants ($y_i \in \{-1, 1\}$) by following the steepest descent paths to the local minimum. Then a SVM is trained based on this dataset to separate the product from the reactant area. This way the *decision boundary of the SVM* is placed along the desired transition surface if an appropriate training set is given. In contrast to other approaches that are based on prior knowledge or planar TS surfaces, the SVM decision boundary is nonlinear and may capture small cavities in the TS surface (Figure 6.1 illustrates this idea).

An informative training set which covers all bottle neck regions is not obtained easily. One of the principal tools in the theoretical study of molecular systems is the method of *molecular dynamics (MD) simulations*. This technique aims at generating a representative sampling of a system at finite temperature (see Jensen [58] for details). Starting from an initial system configuration defined by coordinates and velocities of each atom MD methods simulate the development of the system according to Newton's second equation at a series of finite time steps. The resulting series of time-correlated points in phase space (trajectories) may be used to determine macroscopic quantities of the system based on the ergodic hypothesis[1].

A MD simulation at low temperature tends to only sample the regions in phase space close to the starting conditions and it is very unlikely to observe the rare event of crossing the transitions state surface. Whereas high temperature MD results in various reactive trajectories but doesn't provide information about the low temperature bottleneck regions and reaction rates we are interested in.

In order to reduce sampling time of low temperature MD and focus on informative system states a biased sampling approach is applied and combined with an iterative SVM update: An initial training set is generated by sampling the potential energy surface (PES) with high-temperature MD. The NVT[2] canonical ensemble is sampled in all cases with the Bussi-Donadio-Parrinello thermostat [15]. Points $\mathbf{x}_i$ are collected regularly from independent MD trajectories and minimized in order to determine y_i. These points are used to generate an initial SVM hypersurface.

From this initial sampling of the PES, the SVM is refined by sampling along the decision boundary (or approximated dividing surface). The system is initially placed on the surface and an MD trajectory is then initiated with a total force given by

$$F(\mathbf{x}) = -\nabla U(\mathbf{x}) + F_{surf}, \tag{6.3}$$

where $U(\mathbf{x})$ refers to the potential energy surface and F_{surf} describes an additional force that pulls the system towards the SVM decision boundary. The force $-\nabla U(\mathbf{x})$ in contrast influences the system in the direction of lower potential energy - like a marble placed on a surface is pulled downhill by gravity. While the latter force grows the steeper the PES, the force F_{surf} is independent of the PES and increases with growing distance to the SVM decision boundary.

To calculate the direction of F_{surf} we calculate the local gradient of the SVM model as defined in the previous chapter. Coming from the negative class the gradient of the classification model $\nabla f(\mathbf{x})$ points towards the positive class, i.e. in the direction of the decision

[1] The ergodic hypothesis implies that time average over a single particle (as done in MD) is equivalent to an average of a large number of particles at any given time snapshot.

[2] NVT refers to the constant number of molecules (N), volume (V) and temperature (T).

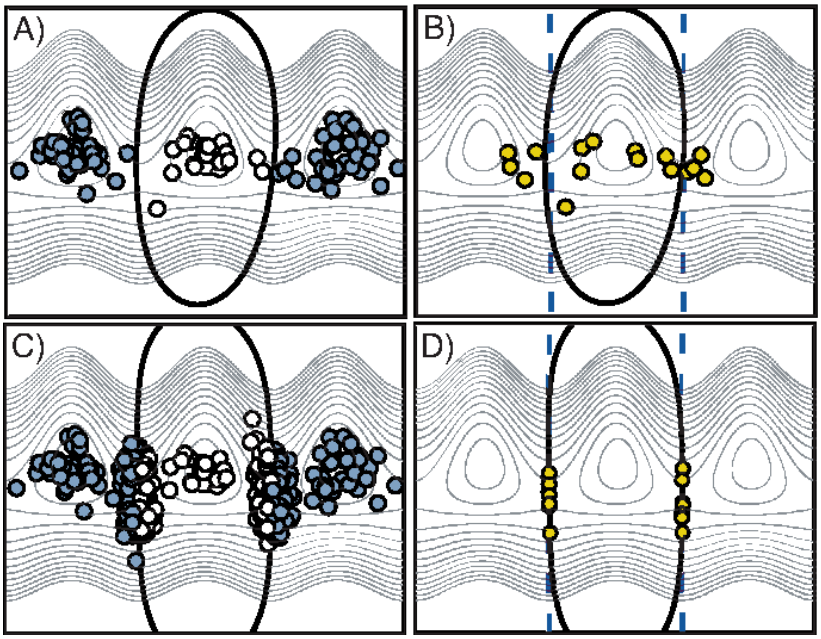


Figure 6.2: The process of sampling/re-learning a hypersurface is demonstrated graphically for the Voter97 potential. An initial surface (A), defined by a set of support vectors (B), is generated from high temperature dynamics. The final surface (C) is defined by a small set of support vectors (D) along the reaction bottlenecks.

boundary; coming from the positive class the direction of F_{surf} is given by the negative gradient $-\nabla f(\mathbf{x})$. The magnitude of F_{surf} is proportional to the distance between $\mathbf{x}$ and the transition state $\mathbf{x}_0$. Since $f(\mathbf{x}_0) = 0$, we can approximate f like in Newton's method by its tangent line through $\mathbf{x}$ and estimate the distance $d(\mathbf{x}, \mathbf{x}_0)$ as $\frac{|f(\mathbf{x})|}{\|\nabla f(\mathbf{x})\|}$. The spring force F_{surf} from $\mathbf{x}$ towards the decision surface is accordingly defined as

$$F_{surf} = -k_s \frac{f(\mathbf{x})\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|^2} \tag{6.4}$$

where k_s is a spring constant for the restraint.

In this manner, the sampling of additional training points is restricted to the region near the SVM hypersurface. After each sampling cycle the SVM is retrained on the full dataset and a new surface is generated. Note that the sampling is inherently parallelizable and that multiple independent trajectories at different temperature can be simulated at the same time. Therefore, we may apply a method called *parallel tempering* (or replica exchange) to make configurations at high temperatures available to the simulations at low temperatures and to sample the low energy configurations efficiently [123]. This approach is compared to the standard MD sampling in the result section.

By iterating through the multiple re-learning cycles, the problem of identifying a dividing surface is transformed from one of parametrization to one of sampling, which is a significantly more tractable problem. If the low-energy regions of the surface have been fully sampled, the SVM-based surface will contain all of the relevant bottleneck regions.

6.3 Experiments and Results

The new method for TS localization is evaluated on three potential energy surfaces. For theses experiments a SVM implementation based on the scikits.learn python package [1]

Table 6.1: κ Values for Dividing Surfaces

Frozen Al(100) Surface		
	100 K	300 K
SVM Dividing Surface	0.97	0.98
Spherical Dividing Surface	0.95	0.95

Relaxed Al(100) Surface		
	100 K	300 K
SVM Dividing Surface	0.55	0.40
Spherical Dividing Surface	0.04	0.09

and libsvm [18] is used with an RBF kernel. The parameters C and γ are determined with a grid search in 5-fold cross-validation [82].

At first, the method is applied to a simple 2-dimensional model potential originally used by Voter in 1997 [140]. The potential contains two saddle points and is periodic in the x-direction and harmonic in the y-direction. If the center basin is chosen as the reactant, then the best dividing surface consists of two 1-dimensional planes that contain the saddle points (Fig. 6.2). The initial high-temperature MD sampling and the resulting SVM surface is shown in Fig. 6.2A-B. Sampling the surface with a force given by Eq. (6.3) and re-learning a new SVM surface in an iterative fashion results in the converged surface shown in Fig. 6.2C-D. We observe that the initial SVM already well identifies the two main saddle points of the Voter potential. Further sampling, using the local gradient approach, leads to aggregation of samples along the dividing surface within the bottleneck regions. The additional training examples allow the refined SVM to capture the linear character of the best dividing surface.

In the case of a single Al adatom[3] on a frozen Al(100) surface, as sketched in Fig. 6.3A, the TST dividing surface is a 2-dimensional diamond shaped hypersurface with four saddle points. In contrast to the Voter97 potential the initial sampling in this case does not provide for a SVM-based identification of all four bottleneck regions. By iterating through many learning cycles and running high temperature dynamics, the training set of samples completely encloses the reactants basin of attraction on the PES leading to a perfect alignment of the support vectors along the ridges between basins. Alternatively, many cycles of parallel tempering may be applied to refine the SVM hypersurface in the low-free energy regions. This parallel approach does not yield the perfect transition surface but ensures that the surface is highly optimized in the critically important bottleneck regions without the computational cost of collecting enough samples to enclose the whole basin of attraction (Fig. 6.3E-F).

To measure the convergence and accuracy of the refined SVM surfaces the transmission coefficient, κ (see Eq. 6.2) is calculated as described by Lu et al. [75]. For an optimal approximation of the rate of reaction the κ value is maximized and we consider a SVM-generated surface as converged when a maximized κ may be extracted.

Since there is no common practice to calculate TS surfaces, we compare our method to a distance-based spherical dividing surface that is defined by the adatom's displacement (Table 6.1). The displacement radius of the sphere was chosen to contain the lowest-energy saddle point. In the case of a frozen Al(100) surface, the escape pathways from the minimum are defined by a single adatom displacement distance. Both the spherical dividing surface as well as the learned SVM dividing surface capture the four escape routes well enough to fully describe the reactive trajectories.

In a second simulation the N atoms of the Al(100) surface are allowed to relax. The dimensionality of the system grows by a factor of $3N$; however, the number of reaction bottlenecks grows much more slowly. In contrast to the frozen surface conditions the spherical approach

[3] Adatoms are atoms adsorbed on a crystal surface.

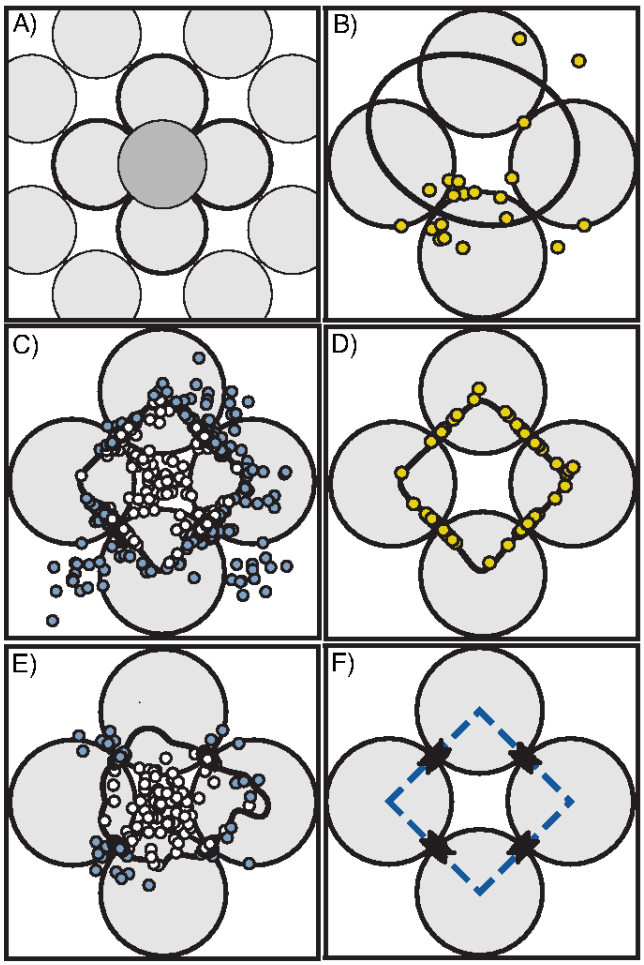


Figure 6.3: For an Al adatom on a frozen Al(100) surface (A), an initial high-temperature MD surface (B) may be refined through high temperature sampling (C) to produce a set of support vectors (D) that align surrounding the basin of attraction for the reactant state. Parallel tempering sampling for the initial surface will produce a dividing surface (E) that is refined at the saddle points such that a forward flux-weighted set of crossing points (F) aligns along the true dividing ridges, shown in blue.

completely fails in this scenario. The different transition mechanisms sketched in Figure 6.4 may cause this effect. Either the attached atom hops over the bridge site or it replaces a surface atom in a two-atom concerted mechanism. Though the first mechanism appears to be more simple the second one is more likely to take place.

Problematic for any distance-based surface like the spherical one is the disparity in the distances by which atoms are displaced during reaction. The SVM dividing surface in contrast is not restricted in terms of distances and captures bottlenecks at all atom displacements. The success of the SVM dividing surface from Table 6.1 is a result of the information density of the surface. The SVM hypersurface captures escape channels at different distances from the minimum as well as the positions of adjacent atoms that are displaced during successful reactive events.

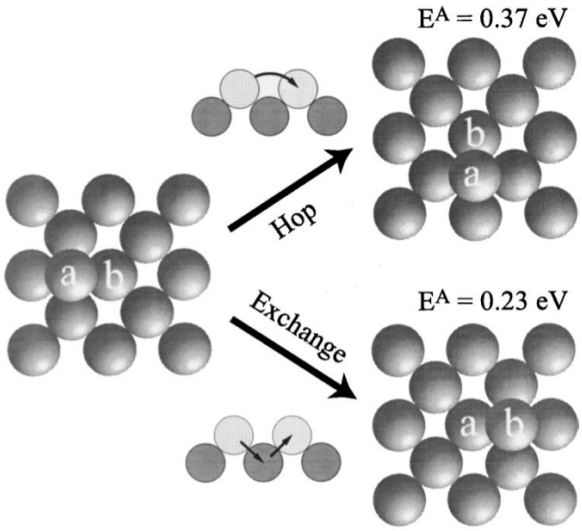


Figure 6.4: This figure illustrates the two different diffusion mechanisms for an Al adatom on an Al(100) surface, the hop and two-atom concerted displacement (after Jóhannesson and Jónsson [60]). The concerted displacement "exchange" process has lower activation energy. Note that the final state is quite different in the two processes and the hop mechanism displaces fewer surface atoms but requires the adatom to travel farther. The SVM method in contrast to the spherical dividing surface enables to model both of these diverse mechanisms together.

6.4 Discussion and Conclusions

In this chapter, SVMs and the local gradient approach were applied to derive a novel method for optimizing transition state dividing surfaces. We define the TS surface as a decision boundary which separates the class of system configurations resulting in reactants from configurations resulting in products. By combining MD simulation techniques with the local gradient approach valuable system configurations around the bottle neck regions are generated to refine the SVM model. This way the number of time-consuming MD evaluations is reduced and we can characterize the TS surface much faster than standard MD simulations.

Our methodology is capable of maximizing the transmission coefficient for systems containing many degrees of freedom without parametrization. In contrast to other approaches our SVM-based method does not require any prior knowledge about the system and is not restricted to hyperplanar dividing surfaces. Moreover, the learned dividing surfaces contain all relevant low-energy reaction bottlenecks when the algorithm is fully converged.

This study exemplifies how the techniques developed within this thesis in the context of drug discovery may have an impact on other fields of computational chemistry such as transition state theory. To discover more parallels and to transfer the machine learning experience accumulated in drug discovery to other fields in computational chemistry is an important issue of further research.

For transition state theory, our results reveal that machine learning techniques may significantly improve computation of reaction rates in high-dimensional systems.

Conclusions

In the present work we have analyzed the requirements of computational methods arising in the drug discovery process and derived novel methods to generate, evaluate and interpret machine learning models.

Summary of Results

The first part of this work focused on enhancing prediction models. We proposed the screening algorithm, *StructRank*, which directly optimizes rankings and is focused on highly binding compounds in order to meet the requirements of virtual screening. The retrospective evaluation on virtual screening and toy data sets revealed a clear advantage of StructRank over related approaches in the relevant scenario where only a small or medium number of molecules with high binding coefficients is available. These results encourage further evaluations of StructRank in prospective screening studies.

The study on hERG inhibition illustrated the limits of ensemble learning and the potential of local bias correction in screening and lead optimization applications. The outstanding performance and simplicity of the introduced *local bias correction* approach makes it a viable alternative to model-retraining in the presence of new measurements. Since local bias correction is a supplementary adjustment it may be applied to improve any kind of predictions beyond computational chemistry.

In the same study we established a *clustered cross-validation* framework. In contrast to standard cross-validation it reduces similarities between training and test splits in order to assess the ability of the model to generalize to new molecules. This model selection method provides a more realistic estimate of the prediction error expected in chemical research and other dependent data applications.

The studies in the first part of this work overall documented the need for an understanding of the chemical task behind the data. Only if both, chemical as well as computational concerns, are combined, excellent prediction models can be obtained.

In the second part of this thesis the focus was turned to the interpretability and transparency of machine learning predictions. We proposed a *visual approach on interpretability* where the most relevant training compounds of single predictions are visualized along with the predicted value itself. To this end a compact method to calculate the impact of training compounds on single predictions was derived for kernel-based machine-learning algorithms. The conducted questionnaire study revealed a beneficial effect of the visualization approach on the participants ability to judge the quality of predictions and illustrated a visual assessment of the domain of applicability. In contrast to numerical measures of reliability that have been published previously, the new approach allows for an intuitive interpretation of predictions by chemical researchers. Thus, this approach represents a new tool to examine predictions where numerical measures indicate a reduced reliability and to understand the strengths and weaknesses of current models.

A *local gradient* approach was established as a tool for interpretation of nonlinear models in terms of local feature importance. The gradients measure the local influence of chemical features on a predicted property. In contrast to existing measures of feature importance this approach is inherently local, suitable for continuous features and considers feature interactions. Due to its locality the approach can identify local as well as global trends in the training data and reveal unique characteristics of compound classes as illustrated for the Ames mutagenicity of steroids. The information provided by local gradients enhances the understanding of the underlying chemical space and may direct compound optimization.

In the last chapter we introduced local gradients and kernel-based machine learning methods to a new field of application. SVMs in combination with local gradients were employed to improve the *sampling of the potential energy surface* in transition state theory. The results of this proof-of-concept study already indicate the potential of machine learning methods in transition state theory, and material science in general.

The present work contributes to closing the gap between machine learning technologies and applied chemical research.

Future Directions

The present work clearly demonstrates the benefits of specializing machine learning algorithms to the specific requirements of chemical research. Further improvements along these lines seem possible: The abundant amount of unlabeled data in virtual screening could be incorporated by an extension of StructRank to semi-supervised learning. Alternatively, labeled data from related targets or similar assays and additional chemical prior knowledge could be included. First attempts to include protein relationships into kernel-based models have been made in the area Chemogenomics for kinases and GPCRs [85, 54]. However, the most urgent and most challenging extension aims at improving the representation of molecules. The method for interpretation introduced in the second part may guide this endeavor and help to characterize the shortcomings of current representations and methods.

Further applications and development of the two explaining methods or alternative approaches to interpret single predictions are necessary. The interpretability of the models represents the key to increasing their acceptance and utility. The proposed methods may help both computer scientists as well as chemists to understand chemical specifics limiting the predictive accuracy and to reveal hints for compound- as well as method-optimization. Furthermore, a standard measure to assess the quality of the different approach to interpretation and prediction quality is needed. There is no gold standard to compare the existing approaches to the domain of applicability.

Besides our study on the application of SVMs in theoretical chemistry, Rupp et al. [103] recently published an innovative approach to predict molecular atomization energies using kernel ridge regression models and Miller et al. [80] applied Gaussian processes to combine calculations of high and low fidelity in order to fit potential energy surfaces. We are confident that machine learning will play a decisive role in theoretical chemistry in the future.

Appendix A

Overview of Machine Learning Methods & Concepts

We will first take a frequentist approach and derive ridge regression and support vector methods based on the concept of empirical risk minimization. Then, Bayesian Gaussian processes are introduced and similarities to previous methods are discussed.

A.1 Empirical Risk Minimization

Recall that in supervised learning we are looking for a function f which can predict the label value y for any new sample $\mathbf{x}$ based on a given dataset $\mathbf{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots (\mathbf{x}_n, y_n)\}$. When measuring the quality of a candidate function f, contradictory aspects have to be considered: On the one hand, the complexity of the function f must be sufficient to express the relation between the given labels $(y_1, y_2, \dots, y_n)$ and the corresponding sample vectors $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$ accurately. On the other hand, f should not be too complex (e.g. too closely adapted to the training data) to allow for reliable predictions of new samples. This trade-off is captured mathematically in the minimization of the *regularized empirical loss function* [119]:

$$\min R_{emp}^{reg}(f) = \underbrace{\frac{1}{n}\sum_{i=1}^{n} \ell(f(\mathbf{x}_i), y_i)}_{\textit{quality of fit}} + \underbrace{\lambda \cdot r(f)}_{\textit{regularizer}} \quad . \tag{A.1}$$

where ℓ refers to a loss function, r to a regularization function and λ to a positive balance parameter. The first term in Equation A.1 measures the quality of the fit of the model on the training data, and the second term penalizes the complexity of the function f to prevent over-fitting. The parameter λ is used to adjust the influence of the regularization function r. The regularization function r not only prevents over-fitting. Moreover, it is often used to ensure that the problem in Equation A.1 is not ill-posed which is required by various optimization methods.
The loss function ℓ determines the loss resulting from the inaccuracy of the predictions given by f. Most predictive machine learning methods minimize the empirical risk function with respect to different model types f, regularization terms r and loss functions ℓ (cf. Table A.1).

Method	Prediction Model	Optimization Problem
Linear Regression	$f(\mathbf{x}) = \mathbf{x}'\mathbf{w} + \mathsf{b}$	$\min_{\mathbf{w}} = \frac{1}{n}\sum_{i=1}^{n} \ell_{\mathsf{se}}(f(\mathbf{x}_i), y_i)$ (A.2)
Ridge Regression	$f(\mathbf{x}) = \mathbf{x}'\mathbf{w}(+\mathsf{b})$	$\min_{\mathbf{w}} = \frac{1}{n}\sum_{i=1}^{n} \ell_{\mathsf{se}}(f(\mathbf{x}_i), y_i) + \lambda \cdot \|\mathbf{w}\|^2$ (A.3)
Kernel Ridge Regression	$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \mathbf{x}) + \mathsf{b}$	$\min_{\alpha} = \frac{1}{n}\sum_{i=1}^{n} \ell_{\mathsf{se}}(f(\mathbf{x}_i), y_i) + \lambda \cdot \sum_{i,j} \alpha_i k(\mathbf{x}_i, \mathbf{x}_j)\alpha_j$ (A.4)
Support Vector Regression	$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \mathbf{x}) + \mathsf{b}$	$\min_{\alpha} = \frac{1}{n}\sum_{i=1}^{n} \ell_{\varepsilon}(f(\mathbf{x}_i), y_i) + \lambda \cdot \sum_{i,j} \alpha_i k(\mathbf{x}_i, \mathbf{x}_j)\alpha_j$ (A.5)
Support Vector Classification	$c(\mathbf{x}) = \mathsf{sgn}(f(\mathbf{x}))$ with $f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \mathbf{x}) + \mathsf{b}$	$\min_{\alpha} = \frac{1}{n}\sum_{i=1}^{n} \ell_{+}(f(\mathbf{x}_i), y_i) + \lambda \cdot \sum_{i,j} \alpha_i k(\mathbf{x}_i, \mathbf{x}_j)\alpha_j$ (A.6)

Table A.1: The machine learning methods listed above are defined by the parametric model function (second column) and the optimization problem (third column) solved to determine the parameters. All optimization problems correspond to the minimization of an empirical risk function (see Eq.A.1).

Regularization Functions	
squared error loss	$\ell_{\text{se}}(f(\mathbf{x}_i), y_i) = (y_i - f(\mathbf{x}_i))^2$ (A.7)
ε-intensive loss	$\ell_\varepsilon(f(\mathbf{x}_i), y_i) = \lvert f(\mathbf{x}_i) - y_i \rvert_\varepsilon = \begin{cases} 0 & \textit{if} \quad \lvert f(\mathbf{x}_i) - y_i \rvert \leqslant \varepsilon \\ \lvert f(\mathbf{x}_i) - y_i \rvert - \varepsilon & \textit{else} \end{cases}$ (A.8)
hinge loss	$\ell_+(f(\mathbf{x}_i), y_i) = \lvert 1 - y_i f(\mathbf{x}_i) \rvert_+ = \begin{cases} 0 & \textit{if} \quad 1 - y_i f(\mathbf{x}_i) \leqslant 0 \\ 1 - y_i f(\mathbf{x}_i) & \textit{else} \end{cases}$ (A.9)

Kernel Functions	
linear function	$\mathbf{k}(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{d} x_i x_i'$ (A.10)
radial basis function (RBF)	$k(\mathbf{x}, \mathbf{x}') = \exp\left(\frac{-\lVert \mathbf{x} - \mathbf{x}' \rVert^2}{2\sigma^2}\right)$ (A.11)
rational quadratic function	$k(\mathbf{x}, \mathbf{x}') = \left(1 + \sum_{i=1}^{d} w_i (x_i - x_i')^2\right)^{-v}$ (A.12)

Table A.2: Common loss and kernel functions used in machine learning, cf. Table A.1.

A.2 Standard Machine Learning Approaches

Ridge Regression

Ridge regression [46] extents the standard linear regression by a regularization function (cf. Table A.1). In both methods the data is estimated using a hyperplane $f(\mathbf{x}) = \mathbf{x}'\mathbf{w} + \mathrm{b}$ with the incline vector or regression weights $\mathbf{w}$ and the offset b. In ridge regression an additional regularization function $r(f) = \|\mathbf{w}\|^2$ imposes a penalty on the sum of squares of regression weights. This regularization is especially important when dealing with correlated inputs: In a typical linear model, a widely large positive weight can be canceled by a similarly negative weight in its correlated counterpart. Effectively, the regularizer shrinks the weights towards zero and towards each other in the optimization process in order to find a unique and robust solution. The problem can be solved analytically; the weight vector $\mathbf{w} = (\mathbf{X}\mathbf{X}' + \lambda\mathbf{I})^{-1}\mathbf{X}\mathbf{y}$ minimizes Equation A.3.

Kernel Ridge Regression and the Kernel Trick

Kernel ridge regression (KRR) is a nonlinear extension of ridge regression (see [22]). The samples are mapped from the input space into a high dimensional space called feature space before a linear ridge regression is applied. However, this mapping function is never specified explicitly. Thanks to the "kernel trick", all calculations can be done in the input space and only the inner product between two samples in the feature space needs to be defined explicitly (see Müller et al. [82]). This inner product is called *kernel function* and the associated feature space is referred to as Reproducing Kernel Hilbert Space [107]. A kernel function $\mathbf{k}(\mathbf{x}, \mathbf{x}')$ can be thought of as a similarity function which drops down to zero for unrelated samples $\mathbf{x}, \mathbf{x}'$. Many machine learning methods are based on kernel functions since they allow to consider geometric quantities in a complex infinite dimensional feature space. Table A.2 comprises kernel functions employed in this work.

KRR is described in terms of empirical risk minimization in Equation A.4. This problem can be solved analytically and the resulting predictor is given as

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \mathbf{x}) + \mathrm{b} \qquad \text{with} \qquad \alpha = (\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{y} \tag{A.13}$$

Note that the model prediction is represented as as a weighted sum of kernel evaluations (cf representer theorem 4.3). The number of optimized parameters (α_i) now corresponds to the number of samples and not to the number of input dimensions like in linear ridge regression. Thus kernel methods of this form can process high-dimensional input data very efficiently.

Support Vector Regression

Support vector regression (SVR) [139, 22, 107] is a kernel-based regression method which can as well be depicted as a linear regression in feature space. Contrary to kernel ridge regression SVR is based on an ε-intensive loss function (Eq A.8) where absolute deviations up to ε are tolerated, and larger differences are penalized linearly.

The optimization problem of SVR in terms of empirical risk minimization is given in Equation A.5. It is convex problem and has a unique solution which can not be written in a closed form but determined efficiently using numerical methods for quadratic programming (see Schölkopf and Smola [107, Chap10], Platt [94]).

One key feature of support vector regression is sparsity. Most of the optimized coefficients (α_i) are zero; only those data points which deviate more (or equal) than ε from the hyperplane have a non-zero coefficient α_i and contribute to the solution. These data points are called *support vectors*.

Support Vector Classification

The idea of support vectors was originally formulated by Vapnik and coworkers [138] in the context of support vector classification (often called support vector machine, SVM) and can be summarized as follows: Given a set of data points in feature space $\phi(\mathbf{x}_i)$, belonging to either class +1 or -1 Vapnik aimed to separate these classes with a hyperplane and additionally maximized the margin around the hyperplane such that $y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq 1$ for all $\mathbf{x}_i$. The primal optimization problem is given by

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{n}\xi_i \\ \text{subject to} \quad & y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0,\ i = 1, \ldots, n. \end{aligned} \tag{A.14}$$

ξ_i are called slack variables and are nonzero for the support vectors which violate

$$y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq 1, \tag{A.15}$$

i.e., for those points that are either misclassified or within the margin ± 1 around the hyperplane $w^T x - b = 0$. In Chapter 2 we build on the idea of slack variables and margin maximization to derive a new ranking algorithm. In Equation A.6 the optimization problem of Equation A.14 is reformulated in terms of empirical risk minimization using the hinge loss (Eq. A.9).

Random Forests

A random forest is essentially a collection of tree predictors where each tree depends on the value of a randomly sampled parameter vector [11]. Random forests can be applied to regression as well as classification tasks. In the case of regression each tree predictor recursively splits the training data into subsets and fits a constant model, e.g., the mean or predominant label value of samples, on each subset. In each step, one subset of compounds X is divided into two subsets X_L, X_R guided by a least squares error criterion. The loss of a set X with n_X compounds is defined as

$$\ell_{tree}(X) = \frac{1}{n_X}\sum_{x_i \in X}(y_i - \bar{y}_X)^2 \tag{A.16}$$

where $\bar{y}_X$ refers to the mean inhibition value of the compounds in X [133]. The best split of a subset X is the split that maximizes

$$\ell_{tree}(X) - \ell_{tree}(X_L) - \ell_{tree}(X_R) \qquad \text{with} \qquad X_L \cup X_R = X. \tag{A.17}$$

In the case of classification the loss function A.16 is modified such that it counts misclassification within a subset. Since a different optimization problem is solved in each step, this approach can not be considered as a global risk minimization problem (Equation A.1) and results in a discontinuous prediction function.

Gaussian Process

In contrast to all previous methods Gaussian processes (GPs) are a kernel-based techniques from the field of Bayesian statistics (see [97] for a detailed introduction to GPs). The idea

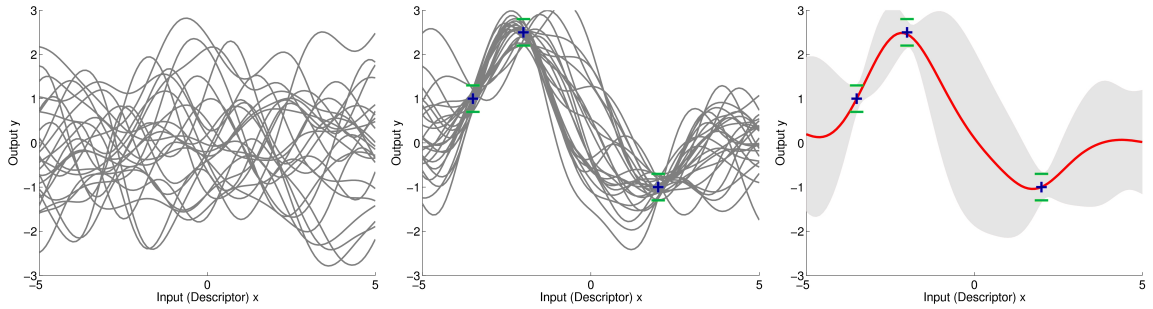


Figure A.1: Idea of Gaussian processes: Specify a distribution of possible functions using a prior (left); eliminate those that do not agree with the data by calculating the posterior (center); average over the remaining functions to generate the predictive distribution of new data points (right).

of GP modeling is to assume a prior probability distribution for the model underlying the data and to update this probability distribution in the light of the observed data to finally obtain a posterior distribution [97]; cf. Figure A.1. A GP prediction is not a single number but a distribution of labels y where the mean can be interpreted as the predicted value and the variance as a confidence estimate or uncertainty:

$$p(y|\mathbf{x}, \mathbf{X}, \mathbf{y}) = \mathcal{N}\left(\mathbf{k}_*^T(\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{y},\ \mathbf{k}(\mathbf{x},\mathbf{x}) - \mathbf{k}_*^T(\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{k}_*\right) \tag{A.18}$$

Note that the predicted mean of a classical GP model with a Gaussian prior equals the prediction of a kernel ridge regression model (see Equation A.13).

Though Gaussian processes are inherently well suited for regression, enhancements for classification exist. In binary *Gaussian process classification* (GPC) a latent Gaussian process is "squashed" through a transfer function to give an output in the range [0,1] which resembles the probability of $\mathbf{x}_0$ being in the positive class. In this case the posterior can not be calculated analytically anymore, and needs to be approximated, e.g., by expectation propagation[1]. In this thesis the Gaussian cumulative distribution function is employed as

$$p(\mathbf{x}) = \frac{1}{2}\mathrm{erfc}\left(\frac{-\mu(\mathbf{x})}{\sqrt{2}\cdot\sqrt{1+\mathrm{var}(\mathbf{x})}}\right). \tag{A.19}$$

Here erfc denotes the complementary error function, and $\mathrm{var}(\mathbf{x})$ the predictive variance (see Eq. 6 in Schwaighofer et al. 2008[115]).

k-Nearest Neighbors

The k-nearest neighbor (KNN) methods are common baseline predictors and require no model to be fit. For each new sample $\mathbf{x}$ the k nearest training samples are selected and the predominant label among those neighbors determines the classification; for regression tasks the mean label value is taken into account. Despite it's simplicity the KNN approach is capable of complex prediction tasks, like short term climate change simulations, but fails to detect global trends. As in all memory-based methods all training samples need to be stored an reconsidered for prediction.

[1] See Kuss and Ramussen [69] for details.

A.3 Model Evaluation

In Chemoinformatics it is highly important to assess the prediction capability of a learning method on new independent test samples. Otherwise it is impossible to include in silico predictions adequately in the crucial process of compound prioritization within screening or lead optimization.

The main performance criteria considered in this thesis are defined in Section 3.4. Since using only one single separate test set can be heavily biased, we employ n-fold cross-validation with multiple repetitions with a reasonable n: The dataset is partitioned into n subsets of equal size. For each subset, the machine learning model is trained on the remaining $n-1$ subsets and then evaluated on the left out subset. This way each sample is used for validation exactly once. The n performance results from the folds are combined to produce a single estimation of the generalization error.

The selection of folds is not trivial. If we choose n too small the results maybe biased and the small-sized training set may limit the model performance. Switching to the other extreme and using as many folds as samples (also called leave-one-out validation) may as well be misleading[47] and underestimate the true error in a physiochemical application scenario: The available experimental dataset only cover a very small subspace of the vast chemical space. Moreover, the included samples are highly dependent, representing series of experiments and compounds with similar structure or properties. In leave-one-out validation it is very likely that for any compound in the validation set there is a similar compound in the training set. During model application however, the compound to be predicted often originates from a new series and is structurally different from the training data and thus produces a higher prediction error. This mis-estimation dilemma can only be partially resolved by using for example, only moderately large n's or by using the clustered cross-validation.

In clustered cross-validation the whole dataset is arranged into equally sized clusters of structurally similar compounds. These clusters are then randomly distributed into n folds and processed as in standard cross-validation. This way too optimistic performance estimates resulting from strong similarities within the dataset may be avoided. Clustered and standard cross-validation are further discussed and comparatively analyzed in Chapter 3.

Appendix B

Abbreviations

ADMET	absorption, distribution, metabolism, excretion, and toxicity
BZR	benzodiazepine receptor
COX-2	dyclooxygenase 2
DHFR	dihydrofolate reductase
GP	Gaussian process
GPC	Gaussian process classification
HTS	high-throughput screening
ML	machine learning
NDCG	normalized discounted cumulative gain
QSAR	quantitive structure-activity relationship
RankSVM	ranking support vector machine
StructRank	structural ranking
SVM	support vector machine
SVR	support vector regression
VS	virtual screening

Bibliography

[1] scikits.learn python module, 2007. available at `http://scikit-learn.sourceforge.net` (accessed June, 2011).

[2] B. N. Ames, E. G. Gurney, J. A. Miller, and H. Bartsch. Carcinogens as Frameshift Mutagens: Metabolites and Derivatives of 2-Acetylaminofluorene and Other Aromatic Amine Carcinogens. *Proceedings of the National Academy of Sciences of the United States of America*, 69(11):3128–3132, 1972.

[3] A. Argyriou, C. A. Micchelli, and M. Pontil. When is there a representer theorem? vector versus matrix regularizers. *Journal of Machine Learning Research*, 10:2507–2529, Dec. 2009. ISSN 1532-4435.

[4] A. Aronov. Predictive in silico modeling for herg channel blockers. *Drug Discovery Today*, 10:149–155, 2005.

[5] A. Aronov. Tuning out of herg. *Current opinion in Drug Discovery & Development*, 11: 128–140, 2008.

[6] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11:1803–1831, June 2010.

[7] J. R. Bertino. Karnofsky memorial lecture. ode to methotrexate. *Journal of Clinical Oncology*, 11(1):5–14, Jan 1993.

[8] C. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[9] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. *Annual Workshop on Computational Learning Theory*, pages 92–100, 1998.

[10] U. Brefeld and T. Scheffer. Semi-supervised learning for structured output variables. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 145–152. ACM, 2006.

[11] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[12] M. Bridgland-Taylor, A. Hargreaves, A. Easter, A. Orme, D. Henthorn, M. Ding, A. Davis, B. Small, C. Heapy, N. Abi-Gerges, F. Persson, I. Jacobson, M. Sullivan, N. Albertson, T. Hammond, E. Sullivan, J.-P. Valentin, and C. Pollard. Optimisation and validation of a medium-throughput electrophysiology-based herg assay using ionworks^{TM} ht. *Journal of Pharmacological and Toxicological Methods*, 54:189–199, 2006.

[13] N. Brown. Chemoinformatics–;an introduction for computer scientists. *ACM Computing Surveys*, 41(2):8:1–8:38, Feb. 2009. ISSN 0360-0300. article number 8 is coded in page numbers.

[14] P. Bruneau and N. McElroy. logD(7.4) modeling using bayesian regularized neural

networks. assessment and correction of the errors of prediction. *Journal of Chemical Information and Modeling*, 46:1379–1387, 2006.

[15] G. Bussi, D. Donadio, and M. Parrinello. Canonical sampling through velocity rescaling. *Journal of Chemical Physics*, 126:014101, 2007.

[16] E. Byvatov and G. Schneider. Svm-based feature selection for characterization of focused compound collections. *Journal of Chemical Information and Computer Sciences*, 44 (3):993–999, 2004.

[17] L. Carlsson, E. A. Helgee, and S. Boyer. Interpretation of nonlinear qsar models applied to ames mutagenicity data. *Journal of Chemical Information and Modeling*, 49(11): 2551–2558, 2009. PMID: 19824682.

[18] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm` (accessed June, 2011).

[19] O. Chapelle. Training a support vector machine in the primal. *Neural Computation*, 19 (5):1155–1178, May 2007.

[20] O. Chapelle, Q. Le, and A. Smola. Large margin optimization of ranking measures. In *In NIPS: Workshop on Machine Learning for Web Search*, 2007.

[21] M. Chulasiri, N. Bunyapraphatsara, and P. Moongkarndi. Mutagenicity and antimutagenicity of flavonoids extracted from millingtonia hortensis l. *Journal of Toxicological Sciences*, 23(suppl. 2):224–228, 1998.

[22] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, UK, 2000.

[23] G. Cruciani, M. Pastor, and W. Guba. VolSurf: a new tool for the pharmacokinetic optimization of lead compounds. *European Journal of Pharmaceutical Sciences*, 11 Suppl. 2: 29–39, 2000.

[24] A. Demiriz, K. P. Bennett, C. M. Breneman, and M. J. Embrechts. Support vector machine regression in chemometrics. In *In Computing Science and Statistics: Proceedings of the 33rd Symposium on the Interface.*, 2001.

[25] D. L. DeWitt, J. S. Day, W. K. Sonnenburg, and W. L. Smith. Concentrations of prostaglandin endoperoxide synthase and prostaglandin i2 synthase in the endothelium and smooth muscle of bovine aorta. *Journal of Clinical Investigation*, 72(6):1882–1888, Dec 1983.

[26] T. G. Dietterich. *Enzyclopedia of Cognitive Science*, chapter Machine Learning, pages 971–981. Nature Publishing Group, 2003.

[27] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik. Support vector regression machines. In *In Advances in Neural Information Processing Systems*, 1997.

[28] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. Wiley, 2001. ISBN 9780471056690.

[29] A. Z. Dudek, T. Arodz, and J. Gálvez. Computational methods in developing quantitative structure-activity relationships (QSAR): a review. *Combinatorial Chemistry & High Throughput Screening*, 9(3):213–228, Mar. 2006. ISSN 1386-2073. PMID: 16533155.

[30] H. Eyring. The activated complex in chemical reactions. *Journal of Chemical Physics*, 3: 107–115, 1935.

[31] R. Féraud and F. Clérot. A methodology to explain neural network classification. *Neural Networks*, 15(2):237 – 246, 2002.

[32] B. Fermini and A. Fossa. The impact of drug-induced qt interval prolongation on drug discovery and development. *Nature Reviews Drug Discovery*, 2:439–447, 2003.

[33] C. G. Fortuna, V. Barresi, G. Berellini, and G. Musumarra. Design and synthesis of trans 2-(furan-2-yl)vinyl heteroaromatic iodise with antitumor activity. *Bioorganic & Medicinal Chemistry*, 16:4150–4159, 2008.

[34] T. Fox and J. Kriegl. Machine learning techniques for in silico modeling of drug metabolism. *Current Topics in Medicinal Chemistry*, 6:1579–1591, 2006.

[35] W. P. Gardiner. *Statistical Analysis Methods for Chemist: A Software-based Approach*. The Royal Society of Chemistry: Cambridge, 1997.

[36] J. Gasteiger. *Chemoinformatics: a textbook*. Wiley-VCH, 2003. ISBN 9783527306817.

[37] H. Glatt, R. Jung, and F. Oesch. Bacterial mutagenicity investigation of epoxides: drugs, drug metabolites, steroids and pesticides. *Mutation Research/Fundamental and Molecular Mechanisms of Mutagenesis*, 111(2):99–118, 1983. ISSN 0027-5107.

[38] M. P. Gleeson. Generation of a set of simple, interpretable ADMET rules of thumb. *Journal of Medicinal Chemistry*, 51(4):817–834, 2008.

[39] W. Guba and O. Roche. *Chemogenomics in Drug Discovery: A Medicinal Chemistry Perspective*, volume 22, chapter 12, pages 325–339. Wiley-VCH Verlag GmbH, 2005.

[40] I. Guyon. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[41] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust Statistics: The Approach Based on Influence Functions*. Wiley, New York, 1986.

[42] K. Hansen, S. Mika, T. Schroeter, A. Sutter, A. ter Laak, T. Steger-Hartmann, N. Heinrich, and K.-R. Müller. Benchmark data set for in silico prediction of ames mutagenicity. *Journal of Chemical Information and Modeling*, 49(9):2077–2081, August 2009.

[43] K. Hansen, F. Rathke, T. Schroeter, G. Rast, T. Fox, J. M. Kriegl, and S. Mika. Bias-correction of regression models: A case study on herg inhibition. *Journal of Chemical Information and Modelling*, 49(6):1486–1496, May 2009.

[44] K. Hansen, D. Baehrens, T. Schroeter, M. Rupp, and K.-R. Mller. Visual interpretation of kernel-based prediction models. *Molecular Informatics*, 30(9):WILEY-VCH Verlag, 9 2011.

[45] S. Harmeling, G. Dornhege, D. Tax, F. C. Meinecke, and K.-R. Müller. From outliers to prototypes: ordering data. *Neurocomputing*, 69(13–15):1608–1618, 2006.

[46] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: data mining, inference and prediction*. Springer series in statistics. Springer, New York, N.Y., second edition, 2009. first edition 2001.

[47] D. M. Hawkins. The problem of overfitting. *Journal of Chemical Information and Computer Sciences*, 44:1–12, 2004.

[48] D. Hecht. Applications of machine learning and computational intelligence to drug discovery and development. *Drug Development Research*, 72(1):53–65, Feb. 2011. ISSN 02724391.

[49] R. Herbrich, T. Graepel, and K. Obermayer. Support vector learning for ordinal regression. In *In ICANN: Proceedings of the Ninth International Conference on Artificial Neural Networks*, 1999.

[50] J. Hert, P. Willett, D. J. Wilton, P. Acklin, K. Azzaoui, E. Jacoby, and A. Schuffenhauer. New methods for Ligand-Based virtual screening:? use of data fusion and machine learning to enhance the effectiveness of similarity searching. *Journal of Chemical Information and Modeling*, 46(2):462–470, 2006. ISSN 1549-9596.

[51] E. J. Horvitz, J. S. Breese, and M. Henrion. Decision theory in expert systems and artificial inteligence. *Journal of Approximation Reasoning*, 2:247–302, 1988. Special Issue on Uncertainty in Artificial Intelligence.

[52] M. Hutter. In silico prediction of drug properties. *Current Medicinal Chemistry*, 16: 189–202, 2009.

[53] A. Inanobe, N. Kamiya, S. Murakami, Y. Fukunishi, H. Nakamura, and Y. Kurachi. In silico prediction of the chemical block of human ether-a-go-go-related gene (herg) k(+) current. *Journal of Physiological Sciences*, 58:459–470, 2008.

[54] L. Jacob and J. Vert. Protein-ligand interaction prediction: an improved chemogenomics approach. *Bioinformatics*, 24(19):2149–2156, Oct. 2008. ISSN 1367-4803. PMID: 18676415 PMCID: 2553441.

[55] C. Jamieson, E. Moir, Z. Rankovic, and G. Wishart. Medicinal chemistry of herg optimizations: Highlights and hang-ups. *Journal of Medicinal Chemistry*, 49:5029–5046, 2006.

[56] K. Järvelin and J. Kekäläinen. Ir evaluation methods for retrieving highly relevant documents. In *In SIGIR: Proceedings of the Annual ACM Conference on Research and Development in Information Retrieval*, 2000.

[57] M. Jeffrey M. Drazen. Cox-2 inhibitors - a lesson in unexpected problems. *New England Journal of Medicine*, 352:1131–1132, 2005.

[58] F. Jensen. *Introduction to Computational Chemistry*. Wiley, 2nd edition, Dec. 2006. ISBN 0470011874.

[59] T. Joachims. Optimizing search engines using clickthrough data. In *In KDD: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2002.

[60] G. H. Jóhannesson and H. Jónsson. Optimization of hyperplanar transition states. *The Journal of Chemical Physics*, 115(21):9644–9656, 2001.

[61] J. Kazius, R. McGuire, and R. Bursi. Derivation and validation of toxicophores for mutagenicity prediction. *Journal of Medicinal Chemistry*, 48:312–320, 2005.

[62] J. C. Keck. Variational theory of reaction rates. *Advances in Chemical Physics*, 13:85, 1967.

[63] G. Kimeldorf and G. Wahba. Some results on tchebycheffian spline functions. *J. Mathematical Analysis and Applications*, 33(1):82–95, 1971.

[64] A. Klenner, V. Hhnke, T. Geppert, P. Schneider, H. Zettl, S. Haller, T. Rodrigues, F. Reisen, B. Hoy, A. M. Schaible, O. Werz, S. Wessler, and G. Schneider. From virtual screening to bioactive compounds by visualizing and clustering of chemical space. *Molecular Informatics*, 31(1):21–26, Jan. 2012. ISSN 1868-1751.

[65] J. Kolpak, P. J. Connolly, V. S. Lobanov, and D. K. Agrafiotis. Enhanced sar maps: Expanding the data rendering capabilities of a popular medicinal chemistry tool. *Journal of Chemical Information and Modeling*, 49(10):2221–2230, 2009. PMID: 19791782.

[66] C. Kramer, B. Beck, J. Kriegl, and T. Clark. A composite model for herg blockade. *ChemMedChem*, 3:254–265, 2008.

[67] J. M. Kriegl, T. Arnhold, B. Beck, and T. Fox. Prediction of human cytochrome p450 inhibition using support vector machines. *QSAR & Combinatorial Science*, 24(4):491–502, 2005.

[68] R. Kühne, R.-U. Ebert, and G. Schüürmann. Model selection based on structural similarity-method description and application to water solubility prediction. *Journal of Chemical Information and Modeling*, 46:636–641, 2006.

[69] M. Kuss and C. E. Ramussen. Assessing approximate inference for bianry gaussian process classification. *Journal of Machine Learning Research*, 6:1679–1704, 2005.

[70] Y. LeCun, L. Bottou, G. Orr, and K.-R. Müller. Efficient backprop. In G. Orr and K.-R. Müller, editors, *Neural Networks: Tricks of the trade*, number LNCS 1524, pages 9–50. Springer, 1998.

[71] V. Lemaire and R. Féraud. Une méthode d'interprétation de scores. In *EGC*, pages 191–192, 2007.

[72] Q. Li, F. Joergensen, T. Oprea, S. Brunak, and O. Taboureau. herg classification model based on a combination of support vector machine method and grind descriptors. *Molecular Pharmaceutics*, 5(1):117–127, 2008.

[73] C. A. Lipinski, F. Lombardo, B. W. Dominy, and P. J. Feeney. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Adv. Drug Delivery Rev.*, 23(1-3):3–25, March 1997.

[74] H. X. Liu, R. S. Zhang, X. J. Yao, M. C. Liu, Z. D. Hu, and B. T. Fan. Qsar study of ethyl 2-[(3-methyl-2,5-dioxo(3-pyrrolinyl))amino]-4-(trifluoromethyl) pyrimidine-5-carboxylate: An inhibitor of ap-1 and nf-κb mediated gene expression based on support vector machines. *Journal of Chemical Information and Computer Sciences*, 43(4): 1288–1296, July 2003. ISSN 0095-2338.

[75] C.-Y. Lu, D. E. Makarov, and G. Henkelman. κ-dynamics—an exact method for accelerating rare event classical molecular dynamics. *Journal of Chemical Physics*, 133:201101, 2010.

[76] G. M. Maggiora. On outliers and activity cliffs–why QSAR often disappoints. *Journal of Chemical Information and Modeling*, 46(4):1535, Aug. 2006. ISSN 1549-9596. PMID: 16859285.

[77] D. M. Maniyar, I. T. Nabney, B. S. Williams, and A. Sewing. Data visualization during the early stages of drug discovery. *Journal of Chemical Information and Modeling*, 46(4): 1806–1818, 2006.

[78] P. McCullagh and J. A. Nelder. *Generalized Linear Models*. Chapman & Hall, New York, 1990.

[79] J. L. Melville, E. K. Burke, and J. D. Hirst. Machine learning in virtual screening. *Combinatorial chemistry high throughput screening*, 12(4):332–343, 2009.

[80] R. L. Miller, L. B. Harding, M. J. Davis, and S. K. Gray. Bi-fidelity fitting and optimization. *The Journal of Chemical Physics*, 136(7):074102–074102–11, Feb. 2012. ISSN 00219606.

[81] T. M. Mitchell. *Machine Learning*. McGraw-Hill Science/Engineering/Math, 1 edition, Mar. 1997. ISBN 0070428077.

[82] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Neural Networks*, 12(2):181–201, May 2001.

[83] R. M. Neal. *Bayesian Learning for Neural Networks*. Lecture Notes in Statistics No. 118. 1996.

[84] A. Nicholls. What do we know and when do we know it? *Journal of Computer Aided Molecular Design*, 22:239–255, Feb. 2008.

[85] X. Ning and G. Karypis. In silico structure-activity-relationship (SAR) models from machine learning: a review. *Drug Development Research*, 72(2):138–146, Mar. 2011. ISSN 02724391.

[86] B. Nisius, A. H. Gller, and J. Bajorath. Combining cluster analysis, feature selection and multiple support vector machine models for the identification of human ether-a-go-go related gene channel blocking compounds. *Chemical Biology & Drug Design*, 73: 17–25, 2009.

[87] O. Obrezanova, G. Csanyi, J. Gola, and M. Segall. Gaussian processes: a method for automatic qsar modeling of adme properties. *Journal of Chemical Information and Modeling*, 47:1847–1857, 2007.

[88] S. O'Brien and M. de Groot. Greater than the sum of its parts: combining models for useful ADMET prediction. *Journal of Medicinal Chemistry*, 48:1287–1291, 2005.

[89] T. I. Oprea. *Chemoinformatics in Drug Discovery*. Wiley-VCH Verlag GmbH & Co, 2005. Tudor I. Oprea (Editor), Raimund Mannhold (Series Editor), Hugo Kubinyi (Series Editor), Gerd Folkers (Series Editor).

[90] F. O'Sullivan, B. S. Yandell, and W. J. Raynor. Automatic smoothing of regression functions in generalized linear models. *Journal of the American Statistical Association*, 81:96–103, 1986.

[91] S. M. Paul, D. S. Mytelka, C. T. Dunwiddie, C. C. Persinger, B. H. Munos, S. R. Lindborg, and A. L. Schacht. How to improve R&D productivity: the pharmaceutical industry's grand challenge. *Nature Reviews Drug Discovery*, 9(3):203–214, Mar. 2010. ISSN 1474-1776.

[92] D. A. Pearlman and P. S. Charifson. Improved scoring of ligand-protein interactions using owfeg free energy grids. *Journal of Medicinal Chemistry*, 44(4):502–511, Feb 2001.

[93] E. Perola, K. Xu, T. M. Kollmeyer, S. H. Kaufmann, F. G. Prendergast, and Y. P. Pang. Successful virtual screening of a chemical database for farnesyltransferase inhibitor leads. *Journal of Medicinal Chemistry*, 43(3):401–408, Feb 2000.

[94] J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods*. MIT Press, 1998. subtitle Support Vector Learning.

[95] J. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*. MIT Press, 2001.

[96] R. Polikar. Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, 6:21–44, 2006.

[97] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. Springer, 2006.

[98] M. Robnik-Šikonja and I. Kononenko. Explaining classifications for individual instances. *IEEE TKDE*, 20(5):589–600, 2008.

[99] S. Rodgers, A. Davis, N. Tomkinson, and H. van de Waterbeemd. QSAR modeling using automatically updating correction libraries: application to a human plasma protein binding model. *Journal of Chemical Information and Modeling*, 47:2401–2407, 2007.

[100] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, Nov 1958. http://www.manhattanrarebooks-science.com/rosenblatt.htm.

[101] M. Rupp and G. Schneider. Graph kernels for molecular similarity. *Molecular Informatics*, 29(4):266–273, Apr. 2010. ISSN 1868-1751.

[102] M. Rupp, T. Schroeter, R. Steri, H. Zettl, E. Proschak, K. Hansen, O. Rau, O. Schwarz, L. Müller-Kuhrt, M. Schubert-Zsilavecz, K.-R. Müller, and G. Schneider. From machine learning to natural product derivatives that selectively activate transcription factor pparγ. *ChemMedChem*, 5(2):191–194, Feb 2010.

[103] M. Rupp, A. Tkatchenko, K. Mller, and O. A. von Lilienfeld. Fast and accurate modeling of molecular atomization energies with machine learning. *Physical Review Letters*, 108(5):058301, Jan. 2012.

[104] M. Sanguinetti and M. Tristani-Firouzi. herg potassium channels and cardiac arrhythmia. *Nature*, 440:463–469, 2006.

[105] G. Schneider and K. Baringhaus. *Molecular design: concepts and applications*. Wiley-VCH, Feb. 2008. ISBN 9783527314324.

[106] G. Schneider, W. Neidhart, T. Giller, and G. Schmid. "scaffold-hopping" by topological pharmacophore search: A contribution to virtual screening. *Angewandte Chemie International Edition*, 38:2894–2896, 1999.

[107] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT, 2002.

[108] B. Schölkopf, S. Mika, C. J. C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A. J. Smola. Input space versus feature space in kernel-based methods. *IEEE Transactions on Neural Networks*, 10(5):1000–1017, 1999.

[109] B. Schölkopf, R. Herbrich, A. J. Smola, and R. Williamson. A generalized representer theorem. *Proceedings of the Annual Conference on Computational Learning Theory*, pages 416–426, 2001.

[110] T. Schroeter. *Machine Learning in Drug Discovery and Drug Design*. PhD thesis, Machine Learning Dept., University of Technology Berlin, 2009.

[111] T. Schroeter, A. Schwaighofer, S. Mika, A. T. Laak, D. Suelzle, U. Ganzer, N. Heinrich, and K.-R. Müller. Estimating the domain of applicability for machine learning QSAR models: A study on aqueous solubility of drug discovery molecules. In *Journal of Computer Aided Molecular Design - special issue on "ADME and Physical Properties"* Schroeter et al. [114], pages 485–498.

[112] T. Schroeter, A. Schwaighofer, S. Mika, A. Ter Laak, D. Suelzle, U. Ganzer, N. Heinrich, and K.-R. Müller. Machine learning models for lipophilicity and their domain of applicability. *Molecular Pharmaceutics*, 4(4):524–538, 2007.

[113] T. Schroeter, A. Schwaighofer, S. Mika, A. ter Laak, D. Suelzle, U. Ganzer, N. Heinrich, and K.-R. Müller. Predicting lipophilicity of Drug-Discovery molecules using gaussian process models. *ChemMedChem*, 2(9):1265–1267, 2007.

[114] T. S. Schroeter, A. Schwaighofer, S. Mika, A. T. Laak, D. Suelzle, U. Ganzer, N. Heinrich, and K.-R. Müller. Estimating the domain of applicability for machine learning qsar models: a study on aqueous solubility of drug discovery molecules. *J. Comput.-Aided Mol. Des.*, 21(12):651–664, Dec 2007.

[115] A. Schwaighofer, T. Schroeter, S. Mika, K. Hansen, A. ter Laak, P. Lienau, A. Reichel, N. Heinrich, and K.-R. Müller. A probabilistic approach to classifying metabolic stability. *Journal of Chemical Information and Modelling*, 48(4):785–796, 2008.

[116] R. P. Sheridan, S. B. Singh, E. M. Fluder, and S. K. Kearsley. Protocols for bridging the peptide to nonpeptide gap in topological similarity searches. *Journal of Chemical Information and Computer Sciences*, 41(5):1395–1406, 2001.

[117] R. P. Sheridan, B. P. Feuston, V. N. Maiorov, and S. K. Kearsley. Similarity to molecules in the training set is a good discriminator for prediction accuracy in qsar. *Journal of Chemical Information and Computer Sciences*, 44(6):1912–1928, 2004.

[118] G. J. Siegel, B. W. Agranoff, R. W. Albers, and S. T. Brady, editors. *Basic Neurochemistry: Molecular, Cellular and Medical Aspects*. Lippincott, Williams and Wilkins: Philadelphia, 1999.

[119] A. J. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.

[120] S. Sorota, X.-S. Zhang, M. Margulis, K. Tucker, and T. Priestly. Characterizing of a herg screen using the ionworks ht: Comparison to a herg rubidium efflux screen. *Assay and Drug Development Technologies*, 3:47–57, 2005.

[121] P. Stansfeld, M. Sutcliffe, and J. Mitcheson. Molecular mechanisms for drug interactions with herg that cause long qt syndrome. *Expert Opinion on Drug Metabolism and Toxicology*, 2:81–94, 2006.

[122] H. Suermondt. *Explanation in Bayesian Belief Networks*. PhD thesis, Department of Computer Science and Medicine, Stanford University, Stanford, CA, 1992.

[123] Y. Sugita and Y. Okamoto. Replica-exchange molecular dynamics method for protein folding. *Chemical Physics Letters*, 314(1-2):141–151, Nov. 1999. ISSN 0009-2614.

[124] I. Sushko, S. Novotarskyi, R. Körner, A. K. Pandey, A. Cherkasov, J. Li, P. Gramatica, K. Hansen, T. Schroeter, K. Müller, L. Xi, H. Liu, X. Yao, T. berg, F. Hormozdiari, P. Dao, C. Sahinalp, R. Todeschini, P. Polishchuk, A. Artemenko, V. Kuz'min, T. M. Martin, D. M. Young, D. Fourches, E. Muratov, A. Tropsha, I. Baskin, D. Horvath, G. Marcou, C. Muller, A. Varnek, V. V. Prokopenko, and I. V. Tetko. Applicability domains for classification problems: Benchmarking of distance to models for ames mutagenicity set. *Journal of Chemical Information and Modeling*, 50(12):2094–2111, Oct. 2010. ISSN 1549-9596.

[125] J. J. Sutherland, L. A. O'Brien, and D. F. Weaver. Spline-fitting with a genetic algorithm: a method for developing classification structure-activity relationships. *Journal of Chemical Information and Computer Sciences*, 43(6):1906–1915, 2003.

[126] C. Swain. Open access and medicinal chemistry. *Chemistry Central Journal*, 1:2, Feb. 2007. ISSN 1752-153X. PMID: 17880736 PMCID: 1975823.

[127] B. Taskar, C. Guestrin, and D. Koller. Max-margin markov networks. In *In Advances in Neural Information Processing Systems*. MIT Press, 2003.

[128] I. Tetko. Neural network studies. 4. intoduction to associative neural networks. *Journal of Chemical Information and Computer Sciences*, 42:717–728, 2002.

[129] I. Tetko. Associative neural network. *Methods in Molecular Biology*, 458:185–202, 2008.

[130] K. Thai and G. Ecker. Predictive models for herg channel blockers: ligand-based and structure-based approaches. *Current Medicinal Chemistry*, 14:3003–3026, 2007.

[131] R. Todeschini and V. Consonni. *Handbook of Molecular Descriptors*. Wiley-VCH, Weinheim, Germany, first edition edition, 2000.

[132] R. Todeschini, V. Consonni, A. Mauri, and M. Pavan. Dragon for Windows and Linux 2007. `http://www.talete.mi.it/help/dragon_help/` (accessed 6 September 2009), 2007.

[133] L. Torgo. Functional models for regression tree leaves. In *ICML '97: Proceedings of the Fourteenth International Conference on Machine Learning*, pages 385–393, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[134] A. Tropsha. *Chemoinformatics in Drug Discovery*, chapter Application of Predictive QSAR Models to Database Mining, pages 437–455. Wiley-VCH Verlag GmbH, 2005.

[135] A. Tropsha and A. Golbraikh. Predictive QSAR modeling workflow, model applicability domains, and virtual screening. *Current Pharmaceutical Design*, 13:3494–3504, Dec. 2007.

[136] J.-F. Truchon and C. I. Bayly. Evaluating virtual screening methods: good and bad metrics for the "early recognition" problem. *Journal of Chemical Information and Modeling*, 47:488–508, 2007.

[137] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.

[138] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

[139] V. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

[140] A. F. Voter. A method for accelerating the molecular dynamics simulation of infrequent events. *Journal of Chemical Physics*, 106(11):4665–4677, 1997.

[141] E. Štrumbelj and I. Kononenko. Towards a model independent method for explaining classification for individual instances. In I.-Y. Song, J. Eder, and T. M. Nguyen, editors, *Data Warehousing and Knowledge Discovery*, volume 5182 of *Lecture Notes in Computer Science*, pages 273–282. Springer, 2008.

[142] N. Wale. Machine learning in drug discovery and development. *Drug Development Research*, 72(1):112–119, Feb. 2011. ISSN 1098-2299.

[143] W. P. Walters and B. B. Goldman. Feature selection in quantitative structure-activity relationships. *Current Opinion in Drug Discovery & Development*, 8(3):329–333, May 2005. ISSN 1367-6733. PMID: 15892248.

[144] W. P. Walters, M. T. Stahl, and M. A. Murcko. Virtual screening - an overview. *Drug Discovery Today*, 3(4):160–178, 1998. ISSN 1359-6446.

[145] M. J. Waring and C. Johnstone. A quantitative assessment of hERG liability as a function of lipophilicity. *Bioorganic & Medicinal Chemistry Letters*, 17:1759 – 1764, 2007.

[146] Y. M. Wen, B. L. Lu, and H. Zhao. Equal clustering makes min-max modular support vector machine more efficient. In *Proceedings of the 12th International Conference on Neural Information Processing (ICONIP 2005)*, pages 77–82, Taipei, 2005.

[147] E. Wigner. The transition state method. *Transactions of the Faraday Society*, 34:29–41, 1938.

[148] P. Willett. A bibliometric analysis of the literature of chemoinformatics. *Aslib Proceedings*, 60(1):4–17, Jan. 2008. ISSN 0001-253X.

[149] W. L. Xie, J. G. Chipman, D. L. Robertson, R. L. Erikson, and D. L. Simmons. Expression of a mitogen-responsive gene encoding prostaglandin synthase is regulated by mrna splicing. *Proceedings of the National Academy of Sciences of the United States of America*, 88 (7):2692–2696, Apr 1991.

[150] X. J. Yao, A. Panaye, J. P. Doucet, R. S. Zhang, H. F. Chen, M. C. Liu, Z. D. Hu, and B. T. Fan. Comparative study of qsar/qspr correlations using support vector machines, radial basis function neural networks, and multiple linear regression. *Journal of Chemical Information and Computer Sciences*, 44:1257–1266, 2004.

[151] C.-N. J. Yu, T. Joachims, R. Elber, and J. Pillardy. Support vector training of protein alignment models. *Journal of Computational Biology*, 15(7):867–880, 2008.

[152] Y. Yue, T. Finley, F. Radlinski, and T. Joachims. A support vector method for optimizing average precision. In *In SIGIR: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2007.