

# MACHINE LEARNING IN DRUG DISCOVERY AND DRUG DESIGN

vorgelegt von  
Dipl.-Chem. Timon Schroeter  
aus Berlin

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:

Gutachter: Prof. Dr. Klaus-Robert Müller  
Prof. Dr. Gisbert Schneider  
Vorsitzender: Prof. Dr. Felix Wichmann

Wissenschaftlichen Aussprache: 3.11.2009

Berlin 2009  
D 83

# CONTENTS

<b>1</b>	<b>Preface</b>	<b>1</b>
1.1	Acknowledgements . . . . .	1
1.2	Parts Published Elsewhere . . . . .	5
<b>2</b>	<b>Introduction</b>	<b>7</b>
2.1	Drug Discovery and Drug Design . . . . .	7
2.2	Machine Learning . . . . .	11
2.3	Machine Learning in Drug Discovery and Drug Design . . . .	13
2.3.1	State of the Art . . . . .	13
2.3.2	Challenging Aspects and the Author’s Contributions .	15
<b>3</b>	<b>Methods (Utilized / Improved)</b>	<b>21</b>
3.1	Overview . . . . .	21
3.2	Data Pre-Processing . . . . .	23
3.3	Learning Algorithms . . . . .	35
3.4	Evaluation Strategies . . . . .	42
3.5	Performance Indicators . . . . .	45
3.6	Incorporating Heterogeneous Types of Information . . . . .	47
3.7	Quantifying Domain of Applicability . . . . .	47
3.8	Presenting Results to Bench Chemists . . . . .	51
<b>4</b>	<b>Methods (Newly Developed)</b>	<b>57</b>
4.1	Overview . . . . .	57
4.2	Incorporating Additional Measurements . . . . .	58
4.3	Explaining Individual Predictions . . . . .	61
4.3.1	Motivation . . . . .	61
4.3.2	Prerequisites . . . . .	63
4.3.3	Concepts & Definitions . . . . .	65
4.3.4	Examples & Discussion . . . . .	67
4.3.5	Conclusion . . . . .	79
4.4	Guiding Compound Optimization . . . . .	81
4.4.1	Introduction . . . . .	81
4.4.2	Definitions . . . . .	83
4.4.3	Related Work . . . . .	87

<b>5</b>	<b>Results</b>	<b>89</b>
5.1	Overview . . . . .	89
5.2	Partition Coefficients . . . . .	90
5.3	Aqueous Solubility . . . . .	92
5.4	Cytochrome P450 Inhibition . . . . .	99
5.5	Metabolic Stability . . . . .	105
5.6	Ames Mutagenicity . . . . .	109
5.7	hERG Channel Blockade Effect . . . . .	113
5.8	Local Gradients for Explaining & Guiding . . . . .	119
5.8.1	Ames Mutagenicity . . . . .	119
5.9	Virtual Screening for PPAR-gamma Agonists . . . . .	127
<b>6</b>	<b>Conclusion</b>	<b>133</b>
<b>A</b>	<b>Appendix</b>	<b>137</b>
A.1	Challenging Aspects of Machine Learning in Drug Discovery .	137
A.1.1	Molecular Representations . . . . .	137
A.1.2	Covariate Shift . . . . .	139
A.1.3	Multiple Mechanisms . . . . .	141
A.1.4	Activity Cliffs . . . . .	142
A.2	Data Used in Modeling Cytochrome P450 Inhibition . . . . .	143
A.3	Descriptors Used in Modelling Cytochrome P450 Inhibition . .	144
A.4	Miscellaneous Plots . . . . .	146
A.5	How to Cite this Thesis . . . . .	147
	<b>Bibliography</b>	<b>149</b>





## CHAPTER 1

# PREFACE

### 1.1 ACKNOWLEDGEMENTS

The first big "Thank You!" is dedicated to Prof. Dr. Klaus-Robert Müller. Klaus: Thank you for supervising this work and giving me the opportunity to be part of your research group. As one of my role models, you inspired me in many ways: Your scientific enthusiasm and curiosity, coupled with your ability to empathize & connect with people, make it easy for you to inspire people around you, be it members of our group, scientific collaborators or industrial clients. Thank you for the mix of guidance, support & freedom you gave me.

Another big "Thank You!" goes to Prof. Dr. Gisbert Schneider. Gisbert: Thank you for reviewing this thesis and for our fruitful discussions. Your support means a lot to me. Our joint virtual screening project gave me a distinct feeling of success: Two souls, alas, are housed within my breast. After having studied chemistry and then working in computer science for several years, it was really great for me to once again work on a project where the most important result was a structural formula & activity.

Anton Schwaighofer and Sebastian Mika were the senior members of my first project team. Thank you for patiently teaching me the basics and helping me to take my first steps. Learning from you and working together was productive and it was fun. I liked our discussions in which many extremes were represented: Elegant & beautiful vs. quick & dirty, science vs. business - this may sound a bit like a commercial for some credit card: "Having a Bayesian and a frequentist on your team - priceless!" ☺

At Fraunhofer FIRST, I shared my first office with Sören Sonnenburg and Olaf Weiss, then moved upstairs to share a two-desk office with Sören. Being present during many of your phonecalls with collaborators from Tübingen I learned more about Kernels for DNA, efficiently implementing SVMs & the MLOSS initiative than from any other source. Thank you for the friendly atmosphere and all our discussions on science & technology, Linux, BSD, PSP and countless other acronyms. Sören: Also thank you for introducing me to Klaus after our initial meeting following Gunnar's lecture at 21C3!

An individual thank you goes to each member of the ChemML team: Katja Hansen, David Baehrens, Fabian Rathke and Peter Vascovic. Katja: Assembling and then later de facto co-leading the team with you was great. The teamwork training we took at the beginning was a good start. I value the very open and direct way we regularly interact. David, Fabian, Peter: With you guys on board and productively working together, we sometimes did within a couple of days what one person could hardly have done within weeks. Each one of you surprised me repeatedly with ideas, ways of implementing things and scientific results of all kinds. Lastly: Even in cooking we are a good team ☺

Pavel Laskov, Konrad Rieck and Patrick Düssel included me in discussions on network protocols and intrusion detection. Thank you for giving me the chance to explore this topic of research for a short and joyful time!

Thank you very much to all present and former members of the IDA group at TU-Berlin and Fraunhofer FIRST! To those I met: Thank you for the friendly and stimulating atmosphere, your support of all kinds and your curious questions. To those I didn't meet: Thank you for paving the way, i.e. providing key insights, software & infrastructure and motivating big footsteps for all of us to step into! This list of remarkable people includes: Bernhard Schölkopf, Alex Smola, Gunnar Rätsch, Benjamin Blankertz, Guido Nolte, Stefan Harmeling, Julian Laub, Matthias Krauledat, Guido Dornhege, Koji Tsuda, Michael Tangermann, Joaquin Quinero Candela, Steven Lemm, Jens Kohlmorgen, Motoaki Kawanabe, Gilles Blanchard, Andreas Ziehe, Florin Popescu, Cristian Grozea, Siamac Fazli, Marton Danoczy, Alexander Binder, Christian Gehl, Olaf Weiss, Roman Krepki, Christin Schäfer, Masashi Sugiyama, Ryota Tomioka, Carmen Vidaurre, Alois Schloegl, Vojtech Franc, David Tax, Ricardo Vigario, Keisuke Yamazaki, Takashi Onoda, Noboru Murata, Frank Meinecke, Stefan Haufe, Jakob Badower, Thilo Thomas Friess, Patricia Vazquez, Christine Carl, Xichen Sun, Paul von Büna, Petra Philips, Matthias Schwan, Matthias Scholz, Stefan Krüger, Daniel Renz, Caspar von Wrede, Bettina Hepp, Irene Sturm, Wenke Burde, Markus Schubert, Rene Gerstenberger, Nils Plath and all present IDA members previously mentioned in this acknowledgement. Furthermore, I would like to thank Andrea Gerdes, Klaus' secretary. Andrea: Thank you very much for your competent engagement!

I would like to thank our collaborators at Bayer Schering Pharma. Working together was productive and it was fun. A special thank you goes to the co-authors on our various journal papers, talks and posters, namely Antonius ter Laak, Nikolaus Heinrich, Detlev Sülzle, Ursula Ganzer, Philip Lienau, Andreas Reichel, Andreas Sutter and Thomas Steger-Hartmann.

Thank you to two members of Prof. Dr. Gisbert Schneider's research

group: Matthias Rupp and Ewgenij Proschak. Thank you for the productive and very enjoyable work on our joint virtual screening project! Matthias: It was good to have you in Berlin for the final part of the implementation and producing the first results. Thank you for hosting us in Frankfurt! Ewgenij & Matthias: Goslar was fun with you guys ☺

A number of institutions and companies supported this work financially and/or by providing office space, computational resources, access to data etc. The lists of grants and supporting institutions includes: Fraunhofer FIRST, University of Potsdam, Idalab GmbH, the German Research Foundation DFG (DFG grants MU 987/4-1 and MU 987/2-1), the European Community (FP7-ICT Program, PASCAL2 Network of Excellence, ICT-216886 & the PASCAL Network of Excellence EU #506778), Bayer Healthcare, Bayer Schering Pharma, Schering & Böhringer Ingelheim.

Prof. Dr. Arne Lüchow deserves a special thank you for joining RWTH-Aachen and promptly creating a special lecture on math & statistics for advanced students who wanted more and, of course, supervising my diploma thesis on quantum monte carlo methods. Arne: I am so happy that you taught me so much math & statistics that I felt confident enough to make the jump from chemistry into computer science. Furthermore, I value the insights into quantum mechanics: Mystical to many, understandable to only few, most of whom truly enjoy it ☺

Supervised by Prof. Dr. Peter Kroll I worked on one of my four research projects that were part of studying at RWTH-Aachen. Peter: Thank you for teaching me density functional theory and the basics of pressure dependent stability of crystal structures. Thank you for taking my initial findings, adding much more and still including my name on what was later going to be my first scientific publication.

Thank you to everybody who helped improve this text with comments & corrections: Anton Schwaighofer, Katja Hansen, David Baehrens and Fabian Rathke.

Finally, I want to express my thanks to those who contributed non-scientifically to the completion of this thesis: I would like to thank my parents Sibylle and Carl Schroeter for their love and support. Furthermore, thank you Linda, Titus, Julia, Heiko, Tanja, Abiba, Samir, Anton, Estelle, David, Sarah, Simone, Patricia, Maurice, Janne, Andrea, Alexandra, Reik and Christine.



## 1.2 PARTS PUBLISHED ELSEWHERE

Table 1.1 lists journal publications (co)authored by the author of this thesis. Full journal names are given in Table 1.2. The most recent list of publications including talks & posters presented at conferences can be found on the world wide web: <http://ml.cs.tu-berlin.de/~timon>

Ref.	Journal	Topic	Sections
[1]	JMLR	Explaining Predictions	4.4, 5.8
[2]	DPMA	Explaining Predictions	4.3, 4.4
[3]	NCB	MKL, Graph Kernels, PPAR $\gamma$	3.6, 5.9
[4]	JCIM	hERG Channel Blockade	4.2, 5.7
[5]	JCIM	Mutagenicity	5.6
[6]	JCIM	Metabolic Stability	5.5
[7]	MP	Domain of Applicability	3.7, 5.2
[8, 9]	JCAMD	Domain of Applicability	3.7, 5.3
[10]	CCHTS	Domain of Applicability	3.7
[11]	JCIM	Aqueous Solubility	5.3
[12]	CMC	Partition Coefficients	5.2
[13, 14]	AC	Density Functional Theory	2.1
[15]	CMC	Pathway Analysis	2.1

**Table 1.1:** This table lists journal publications (co)authored by the author of this thesis. Full journal names are given in Table 1.2. The column “sections” lists sections where parts of each publication are discussed in this thesis.

Abbrev.	Journal / Institution
JMLR	Journal of Machine Learning Research
DPMA	Deutsches Patent- & Markenamt
AC	Angewandte Chemie (International Edition)
JCIM	Journal of Chemical Information and Modeling
MP	Molecular Pharmaceutics
JCAMD	Journal of Computer Aided Molecular Design
CCHTS	Combinatorial Chemistry & High Throughput Screening
CMC	ChemMedChem
NCB	Nature Chemical Biology

**Table 1.2:** Full journal names for the abbreviations used in Table 1.1



## CHAPTER 2

# INTRODUCTION

### 2.1 DRUG DISCOVERY AND DRUG DESIGN

The following section provides an overview over the process of drug discovery and design.<sup>1</sup> An in depth introduction can be found in these books [17, 18]. Steps in the process that have been treated in this thesis are pointed out in the last paragraphs.

Solid biological basic research on disease mechanisms is nowadays the basis of drug discovery and drug design. Ideally, one finds that stimulating or inhibiting a certain target receptor (usually a protein) will help cure the disease, ease the pain etc. Due to advances in genomics [19–21], proteomics and other parts of systems biology, the analysis of signalling pathways [15] is currently gaining importance in understanding complex disease mechanisms. This set of methods can greatly facilitate the process of identification and validation of drug targets (e.g. receptors).

One of the first steps in discovering a new drug is to develop an experiment that allows for testing whether a molecule binds to the target. This type of experiment is called *assay*. Once such an assay has been developed to the point where it is applicable in an automated way, one chooses compounds for investigation in high throughput screening (HTS). The number of compounds depends on the budget of the company and can exceed one million compounds. These are typically selected from a much larger number of compounds that are available in-house or from external vendors. The majority stems from *combinatorial libraries* and can be synthesized by machines using a certain set of chemical reactions to combine predefined suitable building blocks [22]. The step of choosing a collection of compounds for screening from a larger set of available/feasible molecules will later be referred to as *library design*. Results from HTS campaigns are very noisy, the effects of *hit compounds* identified in HTS are therefore carefully examined in regular laboratories. Out of the *confirmed hits*, one selects molecules as starting points for developing *lead compounds*.

---

<sup>1</sup>The actual process depends on the company. This summary was partly inspired by [16].

When looking for hits, assessing the potential of compounds using computational methods (*in silico* methods) can be seen as an alternative to high throughput screening. The most popular group of methods is called *virtual screening* [23, 24]. Depending on which information is available, different virtual screening protocols can be applied:

If, for example, the 3D structure of the target protein is known, one can exploit it in different ways, commonly referred to as *structure based design* [25]. One can simulate (or even calculate<sup>2</sup>) how different molecules fit into the active site of the target. Depending on the amount of human intervention in evaluating the goodness of the fit, this is either referred to as *docking* or *high throughput docking*. The latter not only requires computationally efficient algorithms and high performance computers / clusters, but also good scoring functions. Alternatively, one can virtually build a molecule out of fragments directly inside the active site of the protein model, choosing each fragment so that the fit to the target is optimal. This strategy is called “de novo design” [26–29].

Without a 3D structure of the target protein, one can investigate compounds that are similar to known *actives* (active compounds) [30, 31]. These may be natural products [32], drugs previously developed by other companies that are already on the market but still protected by patents or actives reported in the literature. In the first two cases one looks for molecules where the relevant functional groups can adopt 3D structures similar to the known active compounds. At the same time, they are supposed to be somewhat dissimilar in a chemical sense, because natural products are often expensive to synthesize and patents for compounds also cover chemically very similar structures. Looking for new compounds with similar activity as known actives is called *ligand based virtual screening*. As explained above, the goal is typically not only to find new actives, but new actives of a different *chemotype* or *scaffold* (hence the often used term *scaffold hopping* [33, 34] to describe the goal). Pseudoreceptor models can be used to bridge ligand based- and (protein-) structure based virtual screening [35].

Having found hits using any of the above methods, one proceeds to examining these few compounds in the lab and selects or develops [36] lead compounds (short: *leads*) that are suitable for further development (see below) in so called *hit to lead* programs.

In the following *lead optimization* phase, variants of these lead compounds are synthesized in a more or less systematic way: The many decisions which

---

<sup>2</sup>Hybrid methods of molecular mechanics simulations (for most parts of the protein) and efficient quantum mechanical methods like DFT [13, 14] (for the active site) can be used to precisely calculate interaction energies.



compound or small batch of compounds to synthesize and test next are often made on the basis of very little information. The experience (and luck) of the people making the respective decisions therefore have a lot of influence on the duration of this phase. One continues until one finds a compound that has the required properties or the project is cancelled. At this stage, compounds are synthesized by humans. Together with some basic tests the costs can easily reach 10,000 \$ per compound. Some companies regularly stop a predefined percentage of all projects to avoid investing too much into dead ends or too difficult paths. Both for projects and for individual compounds this is sometimes referred to as “fail cheap - fail early” paradigm. Sometimes thousands of compounds are made and tested until one finally succeeds; even 10,000 compounds are not uncommon. For promising compounds, extensive toxicologic and pharmacologic profiles are done using computational methods, laboratory experiments involving cells (*in vitro* testing) and in animal models. Finally, *drug candidates* are selected for first experiments with humans & patent applications are submitted. The duration of the patent protection depends on the country. Typical times for the United States and Europe are 20 years. Note that time starts running *before* clinical testing begins, i.e. years before the drug reaches the market.

Clinical testing happens in four phases involving increasing numbers of both patients and healthy volunteers (from 20-80 people in phase 1 to 1000-3000 people in phase 3). One determines characteristics like the distribution in the body, suitable dose range, potential side effects and effectiveness of the new drug. Registration can be done after successful phase 3 trials. Ideally, one then gets the permission to market the drug. After introducing the drug into the market, further studies are performed (phase 4).

The whole process can cost on the order of a billion dollars and in rare cases even exceed two billion dollars. Once the drug is on the market, one can, in theory, sell it infinitely, but as soon as the patent protecting the compound runs out, other companies are allowed to “copy” it. Not having invested into basic research and development, these *generic drug makers* can sell it at a very low price and still be profitable.

Speeding up the drug design process is therefore very desirable for the following reasons:

- one can save many millions of euros by avoiding useless experiments and finding good compounds faster
- one can increase the time that one spends alone on the market, protected by a still running patent and can recover investments into basic research and development

Traditionally, early research was very focussed on the effect on the target. Properties relating to Absorption, Digestion, Metabolism, Excretion & Toxicity (*ADME/Tox*) were investigated late in the lead optimization process. In recent years, most companies have started considering these properties as early as possible.

The author of this thesis contributed to developing models for the following properties:

- Partition Coefficients (Sec. 5.2)
- Aqueous Solubility (Sec. 5.3)
- Cytochrome P450 Inhibition (Sec. 5.4)
- Metabolic Stability (Sec. 5.5)
- Ames Mutagenicity (Sec. 5.6)
- hERG Channel Blockade Effect (Sec. 5.7)

These models allow taking the respective properties into account already in early development stages, i.e. when building *libraries* for *high throughput screening*, in *hit to lead* programs and at the beginning of *lead optimization*. Four of these models have been equipped with graphical user interfaces (see Sec. 3.8) and deployed for use by researchers at Schering (now part of Bayer Healthcare).

Furthermore, the author of this thesis participated in a *ligand based virtual screening* project leading to new PPAR $\gamma$  agonists (Sec. 5.9).

Three new machine learning technologies aimed at *lead optimization* were conceived. The first algorithm improves prediction accuracy for new compound classes by means of a local bias correction. The other two algorithms can help human experts in choosing new compounds to investigate: The first method *explains* individual predictions of kernel based models (Sec. 4.3). The second method identifies the features that are most promising for optimizing each individual molecule (Sec. 4.4 and 5.8).

## 2.2 MACHINE LEARNING

The following section introduces general machine learning paradigms and methods. The next section discusses the state of the art, challenging aspects and the authors contributions to machine learning in drug discovery and drug design.

Machine learning can be regarded as data driven generation of predictive models. *Supervised machine learning* assumes that there is a set of  $n$  given pairs  $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$  where  $\mathbf{x}_i \in \mathbb{R}^d$  denotes the vector of the  $d$  feature (descriptor) values calculated in the pre-processing for the object (chemical compound)  $i$  and  $y_i$  refers to the corresponding label. One distinguishes two types of supervised learning problems, depending on the structure of  $y$ . For classification,  $y$  consists of a finite number (very often two, sometimes more) of class labels (i.e. mutagenic vs. non-mutagenic), and the task is to correctly predict the class membership of objects. For regression,  $y_i \in \mathbb{R}$ , and the task is to predict some real valued quantity (i.e. a property like a binding constant) based on the object features. Training examples are assumed as ideally identically distributed samples from a probability distribution  $\mathbf{P}_{X \times Y}$ . One aims to find a function  $f$  which can predict the label for unknown objects represented as feature vectors  $\mathbf{x}$ .

When measuring the quality of  $f$ , 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 feature vectors  $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$  accurately. On the other hand, the estimating function should not be too complex (e.g. too closely adapted to the training data) to allow for reliable predictions of unknown objects. This tradeoff is captured mathematically in the minimization of the *regularized empirical loss function* [37]:

$$\min R_{emp}^{reg}(f) = \underbrace{\frac{1}{n} \sum_{i=1}^n \ell(f(\mathbf{x}_i), y_i)}_{\text{quality of fit}} + \underbrace{\lambda \cdot r(f)}_{\text{regularizer}} . \quad (2.1)$$

where  $\ell : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$  refers to a loss function,  $r : \mathbb{L} \rightarrow \mathbb{R}$  to a regularization function and  $\lambda$  to a positive balance parameter. The first term in Equation 2.1 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  $\lambda$  is used to adjust the influence of the regularization function  $r$ . In addition to preventing over-fitting, it is often used to ensure that the problem in Equation 2.1 is not illposed which is required by various optimization methods. The loss function  $\ell$  determines the loss resulting from

the inaccuracy of the predictions given by  $f$ . Many regression algorithms use the *squared error loss function*

$$\ell(f(\mathbf{x}_i), y_i) = (f(\mathbf{x}_i) - y_i)^2. \quad (2.2)$$

Most inductive machine learning methods minimize the empirical risk function with respect to different regularization terms  $r$  and loss functions  $\ell$ .

Popular supervised learning algorithms include Support Vector Machines [38–42], Gaussian Processes [11, 43, 44], Decision Trees [45] and Artificial Neural Networks [46–48]. The most commonly used way of representing molecules is choosing one out of many available tools to calculate a vector of so called chemical descriptors characterizing the molecule [49]. Standard learning algorithms for vectorial data can then be applied to these descriptors. Sometimes the features of the test set are already available when the model is trained. I.e. one builds a model based on a training set of chemical molecules, intending to apply this model to an already collected/generated library of molecules. This setting is generally referred to as *semi-supervised machine learning*.<sup>3</sup> If a learning machine can suggest which compounds to investigate next to achieve the maximum improvement of the model, we are in an *active learning* scenario [50]. Finally, sometimes no labels exist for a collection of objects and one seeks to detect some type of order in the data based on the features alone. This setting is commonly called *unsupervised machine learning* or clustering [51–57]. More general definitions of machine learning also include different types of signal processing (e.g. in brain computer interface systems [58–64]), and various unsupervised and supervised projection and dimensionality reduction algorithms [65–73]. A recent initiative in the machine learning community [74] advocates the use of open source software for machine learning and points to a free software repository established to facilitate this move.

In the work leading up to this thesis, non-linear Bayesian regression and classification using Gaussian Process priors have been applied to different learning tasks and are also used as the starting points for newly developed algorithms for *explaining* individual predictions and eliciting *hints for compound optimization*. Therefore, a separate section has been devoted to Gaussian Processes, namely Sec. 3.3.

---

<sup>3</sup>Other definitions of *semi-supervised machine learning* are more broad and also include procedures where *any* set of unlabeled data is used in the learning process.

## 2.3 MACHINE LEARNING IN DRUG DISCOVERY AND DRUG DESIGN

### 2.3.1 *State of the Art*

In 2005, when work leading to this thesis commenced, there were no predictive models for metabolic stability (Sec. 5.5) and the hERG channel blockade effect (Sec. 5.7) available on the market. Existing predictive tools for partition coefficients (Sec. 5.2), aqueous solubility (Sec. 5.3) and mutagenicity (Sec. 5.6) performed reasonably well on publicly available benchmark sets of compounds, but did not generalize to the in-house compounds of pharmaceutical companies. Figure 2.1, a scatter-plot from predicting  $\log D_7$  using the commercial tool ACD v. 9.0 [75] for 7013 in-house compounds of Schering, illustrates the problem: We see a lot of compounds where the predictions deviate from the true values by several log units. Most of them occur for compounds with relatively high  $\log D_7$ . Unfortunately, these are the compounds that tend to bind proteins very well and are therefore very interesting in the context of drug discovery & design (most drug targets are proteins).

Furthermore, in 2005 none of the existing commercial models for ADME/-Tox properties had the ability to quantify the confidence into each individual prediction. As explained in Sec. 3.7, this is a very desirable feature in drug discovery & drug design, where models are often operated outside of their respective domains of applicability.

The causes of the anti-inflammatory and anti diabetic effects of bermuda grass (*cynodon dactylon*) were unknown until our screening for new PPAR $\gamma$  agonists (Sec. 5.9) lead to a first hypothesis.

When understandable models for application in *lead optimization* were sought, researchers resorted to building linear models based on small training sets of compounds represented by small sets of descriptors. Using all available training data and/or complex kernel-based models while still being able to understand each individual prediction were not yet feasible (Sec. 4.3). Furthermore, there was no technology that allowed to elicit hints for compound optimization (Sec. 4.4).

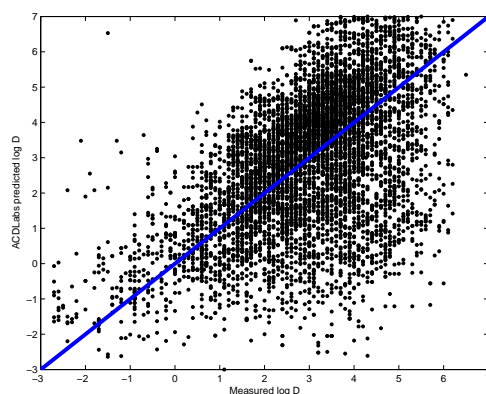
## Drug Discovery poses Challenging Machine Learning Tasks

Challenging Aspect	Author's Contribution
flawed representation concepts multiple mechanisms activity cliffs	
systematical measurement errors heteroschedastic noise (labels) outlying compounds (labels) outlying compounds (features) missing values (features) scarce training data	smart preprocessing, prior knowledge smart preprocessing, prior knowledge smart preprocessing smart preprocessing smart preprocessing publish large benchmark set Sec. 5.6

## Specific Challenges Faced in Lead Optimization

black-box unacceptable hints for optimization desired covariate shift, but new labels av. very high reliability desired	algorithm for explaining Sec. 4.3 algorithm for giving hints Sec. 4.4 algorithm for bias correction Sec. 4.2 confidence estimates Sec. 3.7
--	---

**Table 2.1:** Overview over challenging aspects arising when constructing machine learning based models for use in drug discovery and drug design and the author's contributions.



**Figure 2.1:** This Scatter-plot from predicting  $\log D_7$  using the commercial tool ACD v. 9.0 [75] for 7000 in-house compounds of Schering illustrates how commercial tools typically perform well on public benchmark data but do not perform well on in-house data. See Sec. 5.2 for details on  $\log D$  prediction.

### 2.3.2 Challenging Aspects and the Author’s Contributions

When constructing machine learning based models for use in drug discovery and drug design, many challenging aspects arise. Table 2.1 presents an overview over these challenges. The next subsection introduces challenging aspects that have *not* been treated in this thesis and points to more detailed explanations given in the appendix. The following four subsections are dedicated to challenging aspects where the author has contributed to progress in the field. This includes newly invented algorithms, algorithms initially introduced into the field, careful *pre-processing* and publishing a new large benchmark data set

#### *Challenging Aspects Not Treated*

Molecules are dynamical three dimensional objects, exhibiting many different types of flexibility (see Sec. A.1.1). Available representations of molecules for machine learning either completely ignore this fact by considering only features derived from the two dimensional graph of the molecule, or they consider a small arbitrarily chosen number of 3D structures that may or may not be relevant for the task at hand. Consequently, the accuracy that can be achieved by machine learning models based on these representations is limited. See Sec. A.1.1 for a more detailed discussion including examples.

As explained in Sec. 2.2, popular machine learning algorithms rely on the assumption that training data and future test data are sampled from the same underlying probability density, and further assume that the conditional distribution of target values given the input features (descriptors) is the same in both test and training data. Violation of the first assumption is often referred to as *covariate shift* or *dataset shift* and is encountered in most drug discovery applications (see Sec. A.1.2). If new measurements have become available since the model has been built, bias correction allows to achieve better generalization performance (see subsection on new algorithms below). As of today, there is no satisfying solution known that allows improving predictions in case of violation of the second assumption (*multiple mechanisms*, see Sec. A.1.3). In the work leading up to this thesis, datasets containing multiple mechanisms have been encountered in the studies described in Sec. 5.5 (Metabolic Stability), Sec. 5.3 (Aqueous Solubility), Sec. 5.4 (Cytochrome P450 Inhibition) and Sec. 5.6 (Ames Mutagenicity). See Sec. A.1.3 for a more detailed discussion. Considering confidence estimates to identify reliable predictions can partially alleviate this problem (see separate section below).

Machine learning in drug discovery and design relies on the assumption

that similar molecules exhibit similar activity. Unfortunately, many relevant properties exhibit sudden jumps in activity. The existence of such *activity cliffs* is not entirely surprising since molecular recognition plays a crucial role in determining properties like binding to receptors or the active sites of enzymes. In this thesis, activity cliffs are present in PPAR $\gamma$  binding (Sec. 5.9), Metabolic Stability (Sec. 5.5), Cytochrome P450 Inhibition (Sec. 5.4), Ames Mutagenicity (Sec. 5.6), hERG Channel Blockade Effect (Sec. 5.7) and to some degree even in Aqueous Solubility (Sec. 5.3). See Sec. A.1.4 for more information on activity cliffs.

#### *Data Scarcity Alleviated by New Benchmark Data Set*

Today, the *academic* part of the field of chemoinformatics still suffers from a lack of large high-quality datasets. A new dataset on Ames mutagenicity was collected from the literature by collaborators at Bayer Schering Pharma and jointly released to the public (Sec. 5.6). The set attracted the immediate attention of numerous researchers. Currently, a joint publication by researchers from eight different groups across the world is in preparation.

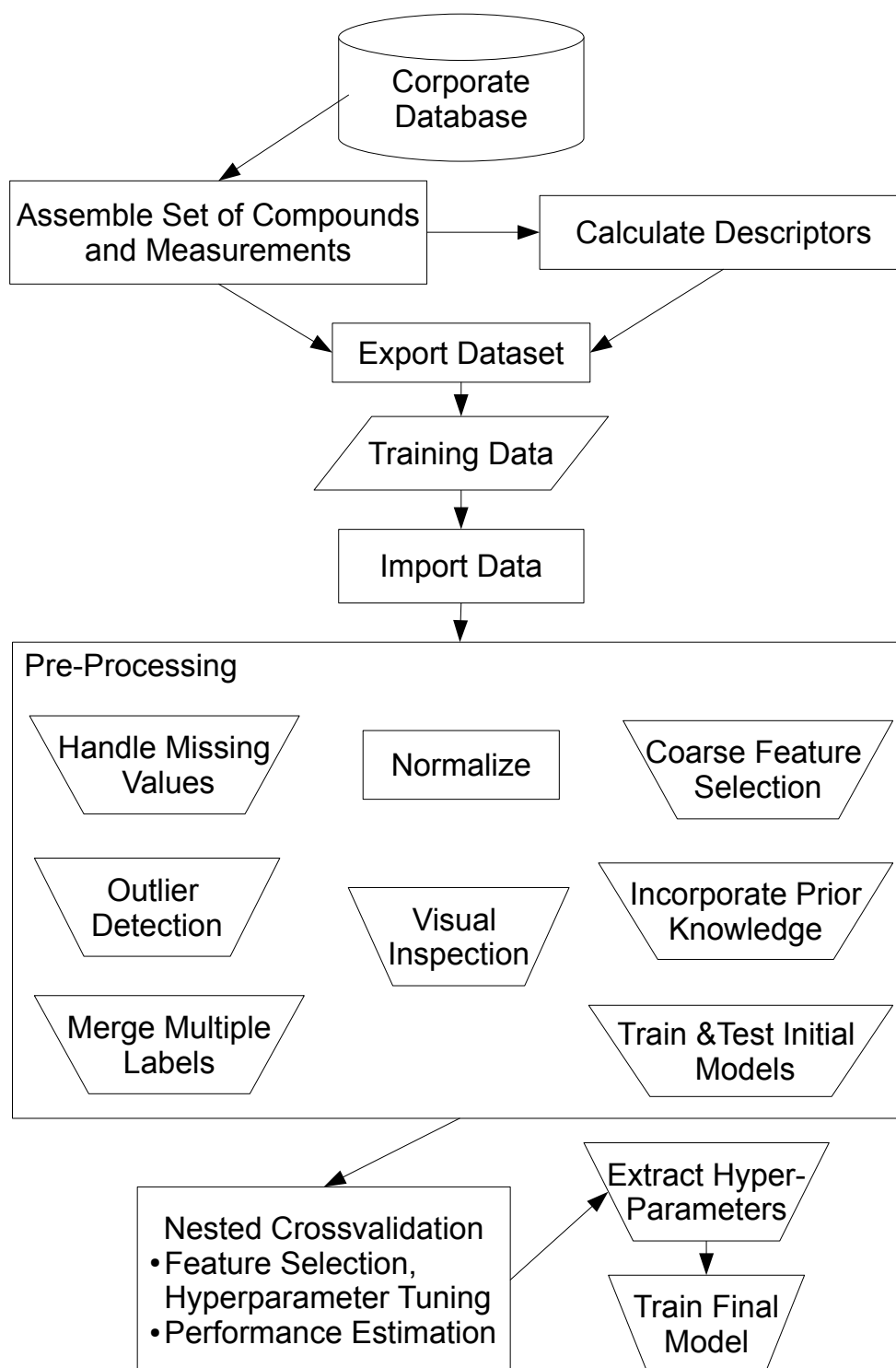
#### *Challenging Aspects Handled in Data Pre-Processing*

Systematical measurement errors, heteroschedastic noise, compounds with outlying labels or features or even missing values in some features necessitate carefully performing various pre-processing steps before training machine learning models. A typical workflow for constructing machine learning models based on data from drug discovery & design is indicated in Figure 2.2. Note the prominent position and size of the box labeled “pre-processing”. Therefore, separate sections have been developed to the most important pre-processing steps:

Visualizing many different aspects of new datasets can help detect potential problems (e.g. strong outliers) early and can give hints as to the difficulty of the modeling task, sensible choice of kernels & parameters etc. In the first subsection of Sec. 3.2, recent examples are used to illustrate how many useful hints and pieces of information can already be found using linear principle component analysis (PCA).

In the chemoinformatics community, there exists a widespread belief that *feature selection* or dimension reduction using projection techniques is essential for any modelling task. The second subsection in Sec. 3.2 acknowledges that while there are good reasons for doing feature selection, it is definitely not always necessary and can even be harmful.





**Figure 2.2:** A typical workflow for constructing machine learning models based on raw data from drug discovery & design. Note the prominent position and size of the box labeled “pre-processing”.

The third subsection in Sec. 3.2 discusses detection and analysis of *outliers* by visual inspection, using outlier indices in descriptor space, based on prior knowledge and when multiple measurements per compound are available. Furthermore, the identification of outlying predictions with respect to the predicted confidence estimates of Gaussian Processes and a “reverse engineering” exercise of models & training sets for a-posteriori explanation of these outlying predictions is presented.

#### *Increasing Reliability of Predictions by Considering Confidence Estimates*

As explained in Sec. 2.2, most machine learning algorithms rely on the fact that training data and future test data are sampled from the same underlying probability density. This assumption is typically violated in drug discovery applications (i.e. they exhibit *covariate shift* or *dataset shift*, see Sec. A.1.2). In other words: In drug discovery and drug design, predictive models are often operated outside of their respective domain of applicability (DOA). Unless some new measurements have been made since the model has been built (see next subsection for new algorithms developed for this scenario), it may not be possible to achieve better predictions. In this case, it may be very helpful to know which predictions are most likely incorrect (outside the DOA) or correct (inside the DOA). Sec. 3.7 explains this concept in more detail and lists heuristics that have been conceived in the chemoinformatics community.

Gaussian Process models can produce a predictive variance along with each individual prediction. This variance can be interpreted as an estimate of the confidence in each individual prediction. This concept has been *introduced* to the chemoinformatics community during the work leading up to this thesis. Sec. 3.3 explains Gaussian Process Models and points to relevant publications. In this thesis, the practical usefulness of predictive variances is investigated in Sec. 5.2 (Partition Coefficients), Sec. 5.3 (Aqueous Solubility), Sec. 5.5 (Metabolic Stability) and in preparing hit-lists in a virtual screening for new PPAR $\gamma$  agonists (Sec. 5.9).

#### *Newly Developed Algorithms*

As time progresses, new projects are started and new compound classes are explored. As explained in Sec. 2.2, almost all supervised machine learning algorithms rely on the fact that training data and future test data are sampled from the same underlying probability density. Violation of this assumption (*covariate shift*) may lead to a bias. More information about *covariate shift*

can be found in the appendix Sec. A.1.2. In the *lead optimization* application scenario, one question that regularly arises is how to best use the first new measurements for compounds belonging to a newly explored compound class, i.e. a new part of the chemical space. New different model selection and bias correction algorithms are introduced in Sec. 4.2 and an evaluation of these algorithms in the context of the hERG Channel Blockade Effect is presented in Sec. 5.7.

In this thesis, two separate methodologies for explaining individual predictions of (possibly non-linear) machine learning models are presented. The method presented in Sec. 4.3 *explains* predictions by the means of visualizing relevant objects from the training set of the model. This allows human experts to understand how each prediction comes about. If a prediction conflicts with his intuition, the human expert can easily find out whether the grounds for the models predictions are solid or if trusting his own intuition is the better idea.

The method presented in Sec. 4.4 utilizes local gradients of the model's predictions to explain predictions in terms of the locally most relevant features. This not only teaches the human expert which features are relevant for each individual prediction, but also gives a directional information. Abstractly speaking, one can learn in which direction a data point has to be moved to increase the prediction for the target value. In the context of lead optimization, this means that the human expert can obtain a type of *guidance in compound optimization*.



## CHAPTER 3

# METHODS (UTILIZED / IMPROVED)

### 3.1 OVERVIEW

This chapter introduces the methodology that was *applied* and partially *re-fined* in the work leading up to this thesis. Methods that have been *newly developed* are presented in Chapter 4.

The first four sections in this chapter deal with topics that are considered essential parts of any machine learning study in chemoinformatics. Each section focuses on the aspects that are most relevant to this thesis as a whole and points to the literature for more comprehensive coverage of the respective topic.

When first analyzing a new (raw) set of data, many aspects need to be considered. Systematical measurement errors, heteroschedastic noise, compounds with outlying labels or features or even missing values in some features necessitate carefully performing various pre-processing steps before training machine learning models. Therefore, Sec. 3.2 contains separate sub-sections on important pre-processing steps, namely visual data inspection, feature selection and outlier detection & analysis.

The account of *machine learning algorithms* (Sec. 3.3) focuses on non-linear Bayesian regression and classification using Gaussian Process priors (GP), because this method was *introduced* into the field of chemoinformatics and for the first time, individual confidence estimates were provided based on a solid theoretical foundation. The section explains how GPs work and discusses their advantages in the context of chemoinformatics.

Sec. 3.4 discusses *evaluation strategies* that allow reaching all the goals that one may have in a typical modeling study: Starting from a batch of data, one selects features, chooses a modeling algorithm and tunes free parameters. In the end one seeks to estimate the generalization performance including or excluding extrapolation. Care has to be taken to both obtain models that generalize well and realistically estimate this achieved generalization performance.

The section on *performance indicators* contains a mostly informal collection of the author’s insights regarding various standard loss functions,

modified versions of standard loss functions and a new loss function that was conceived to express the specific goals of a virtual screening application.

The following three sections deal with more advanced topics.

As detailed in Sec. 3.6, *multiple kernel learning* allows to simultaneously take different aspects of the same piece of information into account. Furthermore, one can use this technique to combine heterogeneous types of information (e.g. vectorial molecular descriptors & molecular graphs).

A typical challenge for statistical models in the chemical space is to adequately determine the *domain of applicability*, i.e. the part of the chemical space where the models' predictions are reliable. Sec. 3.7 treats both heuristics that have been previously used in the chemoinformatics community and recent probabilistic models.

When trying to establish new *computational methodology* that can be *relevant to* the work of *bench chemists* (who tend to focus on the synthetic aspects of chemistry), it is necessary to adapt to their needs and preferences. Sec. 3.8 describes a graphical user interface that allows easy access to models that have been developed with contributions by the author of this thesis.

## 3.2 DATA PRE-PROCESSING

A typical workflow for constructing machine learning models based on data from drug discovery & design has been introduced in Figure 2.2 on page 17. Note the prominent position and size of the box labeled “pre-processing”. In this section, three separate subsections illustrate important pre-processing steps.

The first subsection lists a number of possible first steps in analyzing a new dataset. *Visualizing* many different aspects can help detect potential difficulties early and can give hints as to sensible choices of kernels & parameters and the difficulty of the modeling task.

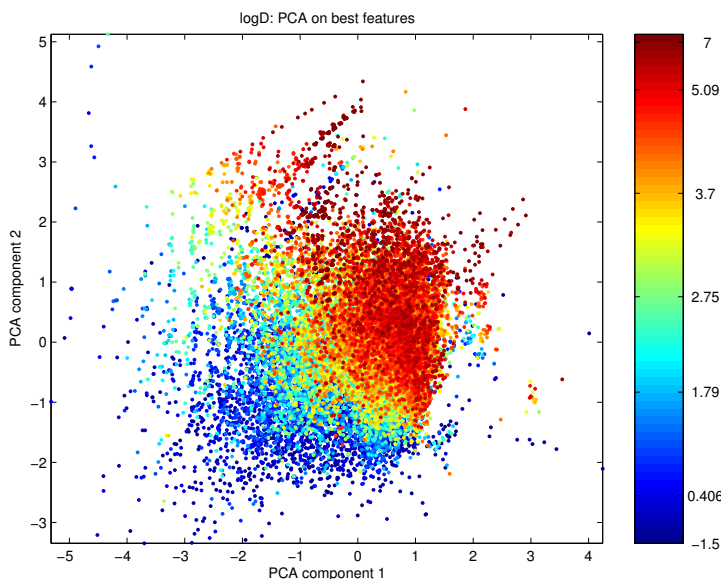
In the chemoinformatics community, there exists a widespread belief that *feature selection* or dimension reduction using projection techniques is essential for any modelling task. The second subsection acknowledges that while there are good reasons for doing feature selection, it is definitely not always necessary and can even be harmful.

The third subsection discusses detection and analysis of *outliers* by visual inspection, using outlier indices in descriptor space, based on prior knowledge and when multiple measurements per compound are available. Furthermore, the identification of outlying predictions with respect to the predicted confidence estimates of Gaussian Processes and a “reverse engineering” exercise of models & training sets for a posteriori explanation of these outlying predictions is presented.

### *Visual Data Inspection*

Research papers tend to focus on the final results of model building & validation, while skipping over some of the preliminaries. This section will list a number of possible first steps in analyzing a new dataset. Visualizing many different aspects can help detect potential problems (e.g. strong outliers) early and can give hints as to the difficulty of the modeling task, sensible choice of kernels & parameters etc. Properties generally useful to visualize include:

- histogram of the target values
- 2/3 D plots of raw/normalized descriptor values
- 2/3 D plots of correlation between descriptors (see Fig. A.4 in Sec. A.4 for an example)
- plots of linear/kernelized principle component analysis (PCA) components of all/most relevant descriptors [76–81]



**Figure 3.1:** Compounds used in modeling log D, projected onto the first two principle components obtained by linear PCA. The color of each point encodes the target value of the respective compound. One can observe a smooth transition from low log D compounds at the bottom left corner of the plot to high log D compounds at the top right corner. Under these circumstances, one can expect that modeling log D will be feasible.

- plots of kernel matrices (if kernel methods are being applied)

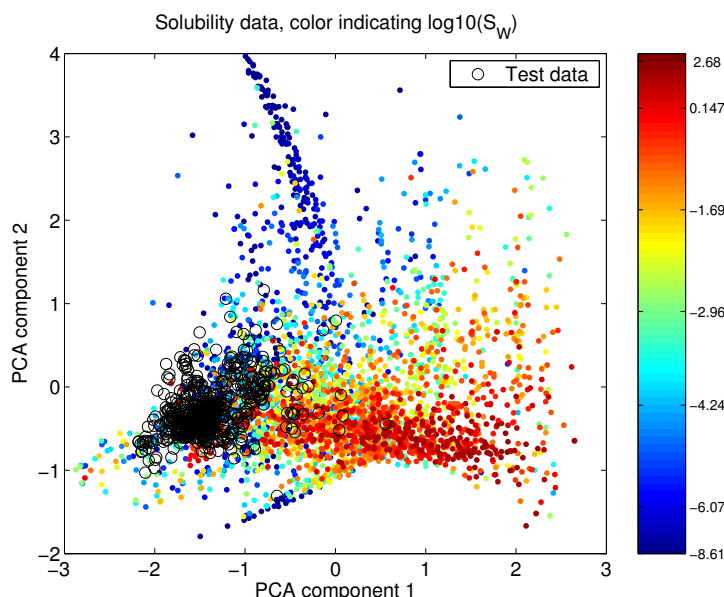
In the following paragraphs, recent examples will be used to illustrate how many useful hints and pieces of information can already be found using linear principle component analysis (PCA). The difficulty of modeling partition coefficients (log D) and aqueous solubility of compounds is estimated from the respective visualizations and the identification of outliers in a dataset on the hERG channel blockade effect is demonstrated.

Figure 3.1 results from a linear PCA of a subset<sup>1</sup> of the features used in modeling the log D dataset discussed in Sec. 5.2. Each compound in the dataset is represented as a dot in the coordinate system. The first two principle components determine the x and y coordinate, respectively, and the log D value of the compound is shown through the color of the respective point. We can observe a strong relationship between the position of the points and their color. Apparently, the first two components alone contain so much information about each compound that it will be easy to estimate its log D quite accurately.

---

<sup>1</sup>A subset of relevant features was selected using the p-values for the hypothesis of linear correlation of each feature with the label as determined in a permutation test using 1000 random permutations of the labels.



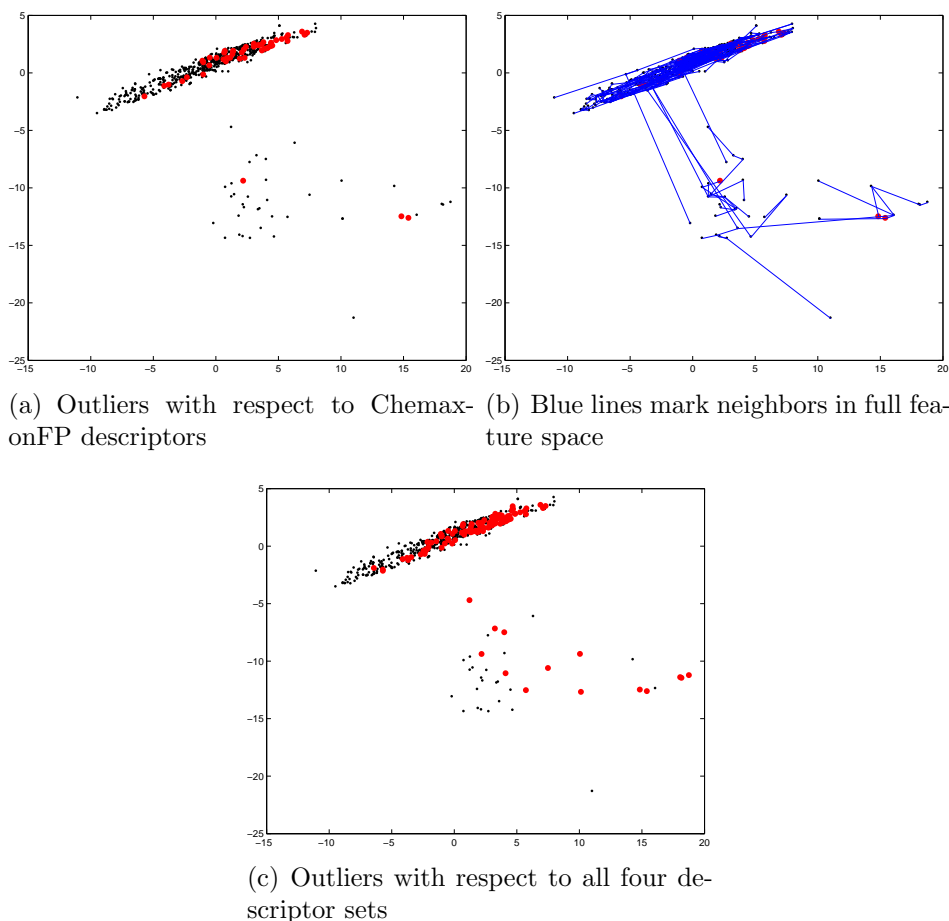


**Figure 3.2:** Compounds used in modeling aqueous solubility, projected onto the first two principle components obtained by linear PCA. The color of each point encodes the target value of the respective compound. We can observe some clusters with compounds of similar solubility and in some cases smooth transitions from one cluster to the next. Under these circumstances, one can expect that modeling will be feasible, although somewhat less easy than the log D task presented previously. Compounds in the external validation set are marked with black circles and are projected onto a small subspace of the 2 D plot.

Figure 3.2 shows a similar plot for set of data on the aqueous solubility of compounds. Again one can observe a relationship between the position and the color of each point, but this relationship seems to be somewhat less obvious than the previously discussed log D task. Additionally, the black circles representing external validation compounds only span a small subspace of the data. This can be taken as a hint that the covariate shift phenomenon may be present. See Sec. 5.3 for a discussion of modeling aqueous solubility and Sec. A.1.2 for a discussion of the covariate shift phenomenon. Indeed, modeling aqueous solubility turned out to be more challenging than the previously mentioned log D task.

Figure 3.3 shows different linear PCA plots for a dataset used to model the hERG channel blockade effect (see Sec. 5.7). In all three plots, the position of each compound is determined by the first two principle components of the ChemaxonFP descriptor set<sup>2</sup>. One can observe a sparse cloud of points below the dense region.

<sup>2</sup>Out of all four available descriptor sets, the the ChemaxonFP descriptors were chosen b.c. in the resulting visualizations the outliers stand out most clearly.



**Figure 3.3:** PCA visualizations of a dataset used to model the hERG channel blockade effect of molecules (represented by ChemaxonFP descriptors) show outliers with respect to the first two principle components (sparse cloud of points below the dense region). Very few of these are automatically labeled as outliers when considering only the ChemaxonFP descriptors (subfigure a), probably because the two dimensional plot does not conserve the neighborhood relationships present in the data (subplot b). When labeling outliers based on four different sets of descriptors, a larger fraction of the sparse bottom cloud is recovered (subplot c).

The data was analyzed using the  $\kappa$ ,  $\gamma$  and  $\delta$  indices introduced by Harmeling et al. (see third subsection in Sec. 3.2 and [82]). The first two indices,  $\kappa$  and  $\gamma$ , are variants of heuristics that have been previously used in the chemoinformatics community (see Sec. 3.7, paragraph on distance based methods):  $\kappa$  is simply the distance to the  $k^{th}$  nearest neighbor and  $\gamma$  is the mean distance to  $k$  nearest neighbors (i.e. a mean of scalars). The last index,  $\delta$ , corresponds to the length of the mean vector to  $k$  nearest neighbors (i.e.

a mean of vectors). Since  $\kappa$  and  $\gamma$  are only based on distances, they do not explicitly indicate whether interpolation or extrapolation is going to be necessary to make a prediction.  $\delta$  allows to make this distinction and indicates exactly how much extrapolation is necessary.

When marking the  $50^3$  most outlying points out of all 676 points based on each points  $\delta$  index (for  $k = 5$ ), we see that only very few points from the sparse cloud below the dense region are selected (Figure 3.3a). This lack of agreement between the  $\delta$  indices and the two dimensional projection probably results from the fact that neighborhood relationships are not conserved in the projection. Figure 3.3b illustrates this issue: The blue lines indicating each points nearest neighbor with respect to all descriptor dimensions often connect points that are not neighbors in the two dimensional projection. In Figure 3.3c, a point is marked in red if it is in the top 50 of  $\delta$  indices for any of the four sets of descriptors used to model the hERG channel blockade effect. When selecting the 50 most outlying compounds with respect to four different descriptor sets, a maximum of 200 compound could be marked as outliers. However, only 117 compounds are selected as outliers, because many compound are considered outliers with respect to more than one set of descriptors. Comparing Figure 3.3a and Figure 3.3c, we find that incorporating information about compounds that are outliers with respect to other descriptor sets than ChemaxonFP helps to recover a larger fraction of points in the sparsely populated cloud below the dense region in all three plots in Figure 3.3. The complete evaluation of single models (see [4]) was then performed twice, once including and once excluding the 117 potential outliers. For the more advanced algorithms (SVM and GP) the resulting performance did not differ significantly for the two cases. It was concluded that the GP and SVM learning algorithms are robust enough to deal with the outliers included in the present set of data. Therefore all compounds from the original set were included in the evaluation of both single and ensemble models (see Sec. 3.2 and Sec. 5.7 for details).

In conclusion, using linear PCA, the relative difficulty of modeling log D and aqueous solubility of compounds was correctly estimated from the respective visualizations. The identification of outliers in a dataset on the hERG channel blockade effect is demonstrated: From the visualizations, one can learn that the  $\kappa$ ,  $\gamma$  and  $\delta$  indices can be used to automatically find outliers. The agreement of visually and automatically identified outliers increases strongly when indices calculated from additional feature sets are

---

<sup>3</sup>Motivated by the histogram of  $\delta$  indices for each descriptor set, the number of outliers was set to 50, i.e. by this working definition, a compound is an outlier if its  $\delta$ -index is in the top 50 of  $\delta$ -indices for each set of descriptors.

considered in the automatic detection procedure.

*Feature Selection vs. Identifying Important Features*

In the chemoinformatics community, there exists a widespread belief that feature selection (or dimension reduction using projection techniques such as linear PCA oder Kernel PCA [76–81]) is essential for any modelling task (see [83–85], [86] and references therein). The following section acknowledges that while there are good reasons for doing feature selection, it is definitely not always necessary and can even be harmful [87, 88].

Selecting a small set of features from a given larger set is usually done with one or both of these goals in mind:

- Reduce the number of features to be used by a given machine learning algorithm.
- Learn more about a given dataset, i.e. make sure that features that correlate with the target value are not artifacts, but do make sense in a physical / chemical way.

There are different possible reasons why one might want to reduce the number of features that will be used in learning:

- Some learning algorithms (e.g. plain unregularized linear regression) fail to converge when correlated features are present. Eliminating correlated features by feature selection or projection methods is therefore essential.
- In some learning algorithms controlling the complexity of the learned models is made more difficult by using a larger number of features.
- Using a smaller number of features reduces the computational cost of any learning algorithm. Depending on the algorithm, the difference may or may not be a good reason to perform feature selection.

When using a learning algorithm that provides an easy way of controlling the complexity of the resulting model, like Gaussian Processes or Support Vector Machines, including many features does not have a negative impact on the models generalization performance, even if they are correlated, misleading or just noise. This finding was confirmed in a number of studies [3–12]. Feature selection algorithms were therefore usually applied to investigate the importance of individual descriptor dimensions to learn more about the data (see the last paragraphs of this section).

There is a number of ways in which feature selection can have a negative impact on modeling: In the end, one typically wants to estimate the

generalization performance of the models one constructed. Feature selection easily leads to overfitting: The resulting model will be too closely adapted to the given dataset. This has two big backdraws at the same time: Firstly, the performance on new data is worse than it could have been with proper feature selection (or maybe no feature selection at all) [87, 88]. Secondly, generalization estimation based on models that overfit via global feature selection leads to overoptimistic results, even if the models themselves are evaluated in a sensible way (see Sec. 3.4 for a discussion).

In the special case of Gaussian Processes we found that using a small subset of descriptors sometimes results in only slightly decreased accuracy when comparing to models built on the full set of descriptors. The predictive variances, however, turn out to be too optimistic [7, 8, 12]. In other words: The target value is predicted accurately for most compounds, but the model cannot correctly detect whether the test compound has, for example, additional functional groups. These functional groups might not have occurred in the training data, and were thus excluded by the feature selection step. In the test case, the information about these additional functional groups is important since it helps to detect that these compounds are different from those the model has been trained on, i.e., the predictive variance should increase. Including whole blocks containing important descriptors leads to both accurate predictions and predictive variances. For a GP model with individual feature weights in the covariance function these surplus descriptors can be given small (but non-zero) weights during training.<sup>4</sup> In consequence the model has more information than it needs for just predicting the target value and can respond to new properties (functional groups etc.) of molecules by estimating a larger prediction error.

As mentioned in that last paragraph, Gaussian Process models can use covariance functions with individual weights for each feature dimension. These parameters are then automatically set in the learning phase. Unless one uses adequate priors, the algorithm tends to set many feature weights to zero, thereby effectively turning the weighting into an internal feature selection mechanism. Just as any other feature selection procedure, this will sometimes lead to overfitting the data [88]. Nevertheless, this procedure can be used to learn something about the training data: During our studies on aqueous solubility, and partition coefficients (log D and log P) features with high weights included the number of hydroxy groups, carboxylic acid groups, keto groups, nitrogen atoms, oxygen atoms and total polar surface area. These features are plausible when considering the physics involved, see [7, 11, 12] for details.

---

<sup>4</sup>This goal can be achieved by imposing appropriate priors on the feature weights.

In trying to understand more complex chemical or biological phenomena, a more fine grained analysis of the relevance of individual features may be helpful. A procedure for identifying *locally* relevant features, i.e. features relevant to the prediction produced for each *individual* molecule, is introduced in Sec. 4.4.

### *Outlier Detection & Analysis*

In the work leading up to this thesis, the following types of detection and analysis of outliers have been applied:

- Identification of outlying compounds by visual inspection
- Identification of outlying compounds using outlier indices in descriptor space
- Identification of outlying measurements based on prior knowledge
- Identification of outlying measurements when combining multiple measurements per compound
- “Reverse engineering” models & training sets for a posteriori explanation of outlying predictions (outlying with respect to the predicted confidence estimates)

Visual inspection of different aspects of each new dataset is useful in many ways beyond pointing to possible outliers. Therefore, a separate subsection has been devoted to this topic, namely the first subsection in Sec. 3.2. Each of the remaining four topics is discussed in one of the following paragraphs.

### *Identification of outlying compounds using outlier indices in descriptor space*

Many kernel based learning algorithms (if regularized properly) are robust enough to deal with outliers in descriptor space as encountered during the work leading up to this thesis. Therefore it is often not necessary to remove any compounds from these datasets. The following paragraph illustrates this typical case: While constructing models for the hERG channel blockade effect (see Sec. 5.7), visual inspection of the raw descriptors and different PCA visualizations indicated that several percent of all compounds in the data set might be outliers (see the first subsection in Sec. 3.2 and [4]). Therefore, it was decided to analyze the data considering the  $\kappa$ ,  $\gamma$  and  $\delta$  indices introduced by Harmeling et al. [82].  $\kappa$  and  $\gamma$  are variants of local density estimators

that are already established in the chemoinformatics community (see Sec. 3.7, paragraph on distance based methods).  $\delta$  is the length of the mean vector to  $k$  nearest neighboring compounds in descriptor space. It measures locally just how much extrapolation is required to make a prediction for each new compound. As described in the context of visual data inspection (see the first subsection in Sec. 3.2), the first modeling experiments have been conducted twice. Once after removing outliers based on their  $\delta$  indices and once with all compounds included. Both kernel based algorithms (Gaussian Process regression (GP) and Support Vector regression (SVR)) performed well, even with outlying compounds in the training set. It was concluded that the GP and SVR learning algorithms are robust enough to deal with the outliers in this set of data and all compounds were used throughout the study. See Sec. 5.7 for details.

#### *Identification of outlying measurements based on prior knowledge*

Machine learners (i.e. experts for machine learning) are typically not experts in every field of science and engineering where they apply their algorithms. However, it is very desirable to understand as much as possible about the problem to be modeled: Any prior information may be useful in modeling. In the case of predicting the metabolic stability of compounds (see Sec. 5.5), the training set of data was generated by measuring the percentage of each compound remaining after incubation with liver microsomes of humans, rats and mice, respectively (for details on the procedure see [6]). It follows that measurements should span the range between 0 % and 100 %. In practice, however, some values exceed 150 %. Upon noticing this fact in a first visual inspection of the data (first subsection in Sec. 3.2), it was found that measurements exceeding 100 % by up to 20 % can be explained by measurement noise. The most plausible explanation for measurement values exceeding 150 % was issues like too slow dissolution: In these cases, the compounds are only partially dissolved when the incubation is started and continue to dissolve during the incubation period. Therefore, it was decided to filter out the most extreme measurements and otherwise treat metabolic stability as a classification problem (see Sec. 5.5 for details).

#### *Identification of outlying measurements when combining multiple measurements per compound*

Datasets with multiple measurements for some of the compounds in the respective datasets were available in the studies described in Sections 5.3 and



5.5. Such measurements are merged to obtain a consensus value for model building. For each compound, one generates the histogram of experimental values. Characteristic properties of histograms are the spread of values (y-spread) and the spread of the bin heights (z-spread). If all measured values are similar (small y-spread), the median value is taken as consensus value. If a group of similar measurements and smaller number of far apart measurements exists, both y-spread and z-spread are large. In this case one treats the far apart measurements as outliers, i.e., one removes them and then uses the median of the agreeing measurements as consensus value. If an equal number of measurements supports one of two (or more) far apart values (high y-spread and zero z-spread), one discards the compound. The only free parameter in this procedure is the threshold between small and large y-spreads. When modeling aqueous solubility, this value was set to the experimental noise value (0.5 log-units). In modeling metabolic stability, this threshold was chosen as 25 units.

In addition to removing outlying measurements, the merging procedure introduced in the last paragraph also allows for passing extra information to learning algorithms capable of using it. Based on the assumption that consensus values based on several agreeing measurements will be more reliable than single measurements one can divide all compounds into groups encoding consideration of 1, 2 or more agreeing measurements, and pass this "noise group" information to the Gaussian Process learning algorithm. In a study on aqueous solubility, a noise level of 0.5 log units was learned for the group of compounds where only one measurement was available. This observation agrees with the prior knowledge about the experimental uncertainty. Compared to single measurements, smaller noise values were learned for compounds where 2, 3 or more agreeing measurements were available, see [11].

*A posteriori explanation of outlying predictions (outlying with respect to the predicted confidence estimates)*

The predictions of GP models are Gaussian distributions, characterized by a mean and a variance. The variance can be transformed into a standard deviation  $\sigma$  for use as a confidence estimate (see also Sec. 3.7). If all assumptions concerning the data and its distribution are met (see Sec. A.1.2, A.1.3), 68 % of all predictions should be within  $1 * \sigma$ , 95 % within  $2 * \sigma$  and 99,7 % within  $3 * \sigma$ . Table 3 in [11] lists compounds where the measured solubility is outside the  $3 * \sigma$  interval in more than 5 (out of the 10) cross-validation trials. With each test compound, the three training compounds with the highest value

for the covariance function are listed (i.e. the training compounds with the highest impact on the respective inaccurate prediction). Interestingly, most of these predictions can be attributed to misleading measurements in the training data (sometimes leading to corrections being made to the respective database) or limitations in the descriptors (i.e. almost identical descriptors for compounds with very different correct measurements of their solubility). See [11] for a discussion of each compound presented.

## 3.3 LEARNING ALGORITHMS

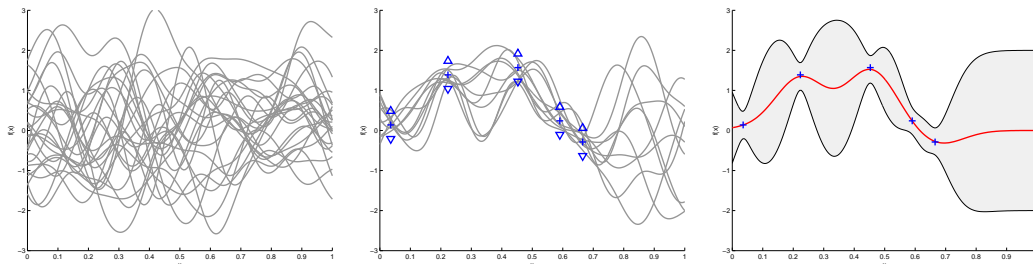
*Non-linear Bayesian Regression using Gaussian Process Priors*

Gaussian Process (GP) models are techniques from the field of Bayesian statistics. O’Hagan [89] presented one of the seminal work on GPs, a recent book [43] presents an in-depth introduction. The first application of GPs in the field of chemoinformatics (including an evaluation of the quality of the confidence estimates) is documented in our paper on aqueous solubility [11]. This paper was quickly followed by a study conducted at BioFocus[90], in which blood-brain barrier penetration, hERG inhibition and aqueous solubility were modeled. Today, two different very powerful commercial implementations of Gaussian Process training tools are available from BioFocus and idalab GmbH, respectively. A number of free implementations [74] have been posted on [91]. This section explains how GPs work, followed by a discussion of their advantages in the context of chemoinformatics.

In GP modelling, one considers a family of functions that could potentially model the dependence of the property to be predicted (function output, denoted by  $y$ , also called “target function”, “target property”, or “target”) from the features (function input, denoted by  $\mathbf{x}$ , in chemoinformatics also referred to as “descriptors”). This space of functions is described by a *Gaussian Process prior*. 25 such functions, drawn at random from the prior, are shown in Figure 3.4 (left). The prior captures, for example, the inherent variability of the target value as a function of the features. This prior belief is then updated in the light of new information, that is, the measurements (“labels”) at hand. In Figure 3.4 (middle), the available measurements are illustrated by three crosses. Principles of statistical inference are used to identify the most likely posterior function, that is, the most likely target function as a combination of prior assumptions and observed data (shown in the right panel of Figure 3.4). The formulation with a prior function class is essential in order to derive predictive variances for each prediction. Note also that the uncertainty increases on points that are far from the measurements.

The main assumption of a Gaussian Process model is that the target function can be described by an (unknown) function  $f$  that takes a vector of molecular descriptors as input, and outputs the target.  $\mathbf{x}$  denotes a vector of descriptors, which is assumed to have length  $d$ . The target property of a compound described by its descriptor vector  $\mathbf{x}$  can thus be written as  $f(\mathbf{x})$ . It is assumed that  $f$  is inherently random.<sup>5</sup>

<sup>5</sup>The notation here is chosen to allow an easy understanding of the material, thus dropping, e.g., a clear distinction between random variables and their outcome.



**Figure 3.4:** Modelling with Gaussian Process priors. *Left:* 25 samples from a Gaussian Process prior over functions, each plotted as  $y = f(x)$ . For illustration, only functions for one-dimensional input  $x$  are considered. *Middle:* After observing 3 data points (crosses), one only believes in functions from the prior that pass through a “tunnel” (depicted by the triangles) near the data (crosses). These functions are samples from the “posterior” distribution. *Right:* Summarizing representation of beliefs about the plausible true functions, obtained from the 25 samples from the posterior shown in the middle pane. For each input one computes the mean of these functions (red line) and the standard deviation. The shaded area encompasses  $\pm 2$  standard deviations around the mean.

The Gaussian Process model is built from measurements for a set of  $n$  compounds. For each of these  $n$  compounds, one has a descriptor vector,  $\mathbf{x}_1 \dots \mathbf{x}_n$ , (each of length  $d$ ), together with a measurement of the target property,  $y_1, \dots, y_n$ . Additionally, one accounts for the fact that these measurements are not accurate, and assumes that the  $n$  measured values are related to actual target property by

$$y_i = f(\mathbf{x}_i) + \epsilon, \quad (3.1)$$

where  $\epsilon$  is Gaussian measurement noise<sup>6</sup> with mean 0 and standard deviation  $\sigma$ .

The name “Gaussian Process” stems from the assumption that  $f$  is a random function, where functional values  $f(\mathbf{x}_1), \dots, f(\mathbf{x}_n)$  for any finite set of  $n$  points form a Gaussian distribution.<sup>7</sup> This implies that one can describe the process also by considering pairs of compounds  $\mathbf{x}$  and  $\mathbf{x}'$ . The covariance for the pair is given by evaluating the covariance function,

$$\text{cov}(f(\mathbf{x}), f(\mathbf{x}')) = k(\mathbf{x}, \mathbf{x}'), \quad (3.2)$$

similar to kernel functions in Support Vector Machines [38, 92]. Note, that all assumptions about the family of functions  $f$  are encoded in the covariance

<sup>6</sup>In the typical Gaussian Process model, all measurements share the same measurement noise. This condition can be relaxed, i.e. to incorporate prior knowledge about distinct groups of compounds with different measurement noise.

<sup>7</sup>For simplicity, it is assumed that the functional values have zero mean. In practice, this can be achieved easily by simply shifting the data before model fitting.

function  $k$ . Each of the possible functions  $f$  can be seen as one realization of an “infinite dimensional Gaussian distribution”.

Let us now return to the problem of estimating  $f$  from a data set of  $n$  compounds with measurements of the target property  $y_1, \dots, y_n$ , as described above in Eq. (3.1). Omitting some details here (the derivation can be found in appendix B in [11]), it turns out that the prediction of a Gaussian Process model has a particularly simple form. The predicted function for a new compound  $\mathbf{x}_*$  follows a Gaussian distribution with mean  $\bar{f}(\mathbf{x}_*)$ ,

$$\bar{f}(\mathbf{x}_*) = \sum_{i=1}^n \alpha_i k(\mathbf{x}_*, \mathbf{x}_i). \quad (3.3)$$

Coefficients  $\alpha_i$  are found by solving a system of linear equations,

$$\begin{pmatrix} k(\mathbf{x}_1, \mathbf{x}_1) + \sigma^2 & k(\mathbf{x}_1, \mathbf{x}_2) & \dots & k(\mathbf{x}_1, \mathbf{x}_n) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) + \sigma^2 & \dots & k(\mathbf{x}_2, \mathbf{x}_n) \\ \vdots & \vdots & & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & k(\mathbf{x}_n, \mathbf{x}_2) & \dots & k(\mathbf{x}_n, \mathbf{x}_n) + \sigma^2 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \quad (3.4)$$

In matrix notation, this is the linear system  $(K + \sigma^2 I)\alpha = \mathbf{y}$ , with  $I$  denoting the unit matrix. In this framework, one can also derive that the predicted property has a standard deviation of

$$\text{std } f(\mathbf{x}_*) = \sqrt{k(\mathbf{x}_*, \mathbf{x}_*) - \sum_{i=1}^n \sum_{j=1}^n k(\mathbf{x}_*, \mathbf{x}_i) k(\mathbf{x}_*, \mathbf{x}_j) L_{ij}} \quad (3.5)$$

where  $L_{ij}$  are the elements of the matrix  $L = (K + \sigma^2 I)^{-1}$ .

**Relations to Support Vector Machines** Gaussian Process models share with the widely known support vector machines the concept of a kernel (co-variance) function. Support vector machines (SVM) implicitly map the object to be classified,  $\mathbf{x}$ , to a high-dimensional feature space  $\phi(\mathbf{x})$ . Classification is then performed by linear separation in the feature space, with certain constraints that allow this problem to be solved in an efficient manner. Similarly, support vector regression [92] can be described as linear regression in the feature space. Gaussian Process models can as well be seen as linear regression in the feature space that is implicitly spanned by the covariance (kernel) function [43]. The difference lies in the choice of the loss function: SVM regression has an insensitivity threshold, that amounts to ignoring small prediction errors. Large prediction errors contribute linearly to the loss. GP

models assume Gaussian noise, equivalent to square loss. As for support vector machines, the mapping of input space to feature space is never computed explicitly. In SVMs and GP models, only dot-products of the form  $\langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle$  are used in the algorithm. Such dot products can be evaluated cheaply via the covariance (kernel) function, since  $\langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle = k(\mathbf{x}, \mathbf{x}')$ .

Note, however, that SVMs are completely lacking the concept of uncertainty. SVMs have a unique solution that is optimal under certain conditions [92, 93]. Unfortunately, these assumptions are violated in some practical applications.

**Relations to Neural Networks** Radial Basis Function networks with a certain choice of prior distribution for the weights yield the same predictions as a Gaussian Process model [43]. More interestingly, it can be shown that a two-layer neural network with an increasing number of hidden units converges to a Gaussian Process model with a particular covariance function [94].

**Using GP Models** For predicting Partition Coefficients (Sec. 5.2) and Aqueous Solubility (Sec. 5.3), a covariance function of the form

$$k(\mathbf{x}, \mathbf{x}') = \left( 1 + \sum_{i=1}^d w_i (x_i - x'_i)^2 \right)^{-\nu} \quad (3.6)$$

is used (the “rational quadratic” covariance function [43]).  $k(\mathbf{x}, \mathbf{x}')$  describes the “similarity” (covariance) of the target property for two compounds, given by their descriptor vectors  $\mathbf{x}$  and  $\mathbf{x}'$ . The contribution of each descriptor to the overall similarity is weighted by a factor  $w_i > 0$  that effectively describes the importance of descriptor  $i$  for the task of predicting the respective target property.

Clearly, one cannot set the weights  $w_i$  and the parameter  $\nu$  a priori. Thus, one extends the GP framework by considering a superfamily of Gaussian Process priors, each prior encoded by a covariance function with specific settings for  $w_i$ . The search is guided through the superfamily by maximizing a Bayesian criterion called the evidence (marginal likelihood). For  $n$  molecules  $\mathbf{x}_1, \dots, \mathbf{x}_n$  with associated measurements  $y_1, \dots, y_n$ , this criterion is obtained by “integrating out” everything unknown, namely all the true functional values  $f(\mathbf{x}_i)$ . Using vector notation for  $\mathbf{f} = (f(\mathbf{x}_1), \dots, f(\mathbf{x}_n))$  and  $\mathbf{y} = (y_1, \dots, y_n)$ , one obtains

$$\mathcal{L} = p(\mathbf{y} \mid \mathbf{x}_1, \dots, \mathbf{x}_n, \theta) = \int p(\mathbf{y} \mid \mathbf{f}, \theta) p(\mathbf{f} \mid \mathbf{x}_1, \dots, \mathbf{x}_n, \theta) d\mathbf{f}. \quad (3.7)$$

This turns out to be

$$\mathcal{L} = -\frac{1}{2} \log \det(K_\theta + \sigma^2 I) - \frac{1}{2} \mathbf{y}^\top (K_\theta + \sigma^2 I)^{-1} \mathbf{y} - \frac{n}{2} \log 2\pi \quad (3.8)$$

Here,  $\det$  denotes the determinant of a matrix, and  $^\top$  is vector transpose.  $K_\theta$  is used to explicitly denote the dependence of the covariance matrix  $K$  on a set of parameters  $\theta$  of the covariance function.<sup>8</sup> Gradient ascent methods<sup>9</sup> can now be used to maximize  $\mathcal{L}$  with respect to covariance function parameters  $\theta$  and the measurement variance  $\sigma^2$ . References [43, 96] present further details.

The following advantages of Gaussian Processes are relevant for their application in drug discovery & design:

- As illustrated above, Gaussian Process models allow for quantifying the confidence in each individual prediction. Since this feature is very useful in chemoinformatics, a separate section has been devoted to this topic, namely Sec. 3.7.
- In GP modeling, heterogeneous types of information about one set of compounds can be combined via combined covariance functions. This Bayesian concept is similar in spirit to the frequentist concept of multiple kernel learning, see Sec. 3.6.
- Data sets for multiple target values (e.g. one set for aqueous solubility and a different set for partition coefficients) can be treated in a single learning procedure. Even if the datasets overlap only partially, information gained from one set eases solving the remaining task(s) [44]. In the machine learning community, this is called multi task learning. A concise overview of multi task learning (using a variety of algorithms) in chemoinformatics can be found in this recent article [97].
- When using kernels with separate width parameters for each dimension, the weighting of the features can be optimized by a gradient ascent on the evidence. Optionally, one can later use this information to reduce the number of features to present to a second GP or other learning algorithm or to identify important features (see the second subsection in Sec. 3.2 for a discussion).
- Many types of previous knowledge that one might have about the data can be included into GP models. In [11], it was known that different

---

<sup>8</sup>In the case of Eq. (3.6),  $\theta = \{\nu, w_1, \dots, w_d\}$  for a total of  $d$  descriptors.

<sup>9</sup>In the actual implementation, the Broyden-Fletcher-Goldfarb-Shanno method [95] is used.

subsets of measurements in the training data were moderately reliable, more reliable and very reliable, respectively. We then trained GP models such that a different noise variance was learned for each group of compounds. Even without specifying different priors for the groups, the models automatically learned lower noise levels for the more reliable measurements. See [11] for details.

- The algorithm presented in Sec. 4.3 *explains* predictions of machine learning models by the means of visualizing relevant objects from the training set of the model. This allows human experts to understand how each prediction comes about. In case of kernel methods (including Gaussian Processes), the contribution of individual testpoints can be calculated analytically.
- Sec. 4.4 proposes a method that can explain the local decisions taken by arbitrary (possibly) non-linear classification algorithms. In a nutshell, 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, a probabilistic approximate mimic of the learning machine to be explained is employed. In the special case of Gaussian Processes, local gradients of predictions can be calculated analytically. See Sec. 4.4 for derivations & illustrations and Sec. 5.8 for an application in drug discovery & design.

Due to these advantages, GP regression models were used in the following studies: [3–12, 90, 98]. The first use of GP *classification* models in the context of drug discovery and design is described in our study [6], further applications include [1, 5].

### *Other Learning Algorithms*

When introducing new methods, it generally makes sense to compare their performance with established models of equal or lesser complexity. Brief introductions to each method can be found in [10]. The reader is invited to consult the literature cited directly following each algorithm’s name below to learn more about the respective method.

- Support Vector Machines [38–42] for regression and classification served as equally complex but more established [99, 100] type of model in [3–8, 12].



- 
- Random forests [101] (i.e. ensembles of decision trees [45]) have been previously used in drug discovery & drug design. Benchmarks were included in [4, 5, 7, 8, 12].
  - k-Nearest-Neighbor models were applied in [5], Sec. 4.3 and 4.4.
  - Linear models [102] represent the least complex type of models and were included as baselines in [3–8, 12].

### 3.4 EVALUATION STRATEGIES

Many different strategies for evaluating models exist:

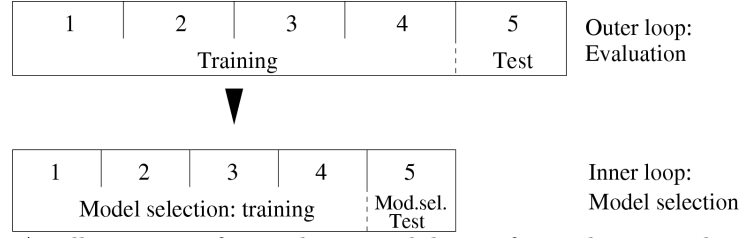
- single validation set
- leave one out cross-validation
- leave k-percent out cross-validation
- leave one cluster out cross-validation
- nested cross-validation variants
- blind tests / prospective evaluation

When choosing an evaluation strategy, the most important question has to be: What is the goal of this evaluation? One may want to

1. select features / descriptors
2. select a modeling algorithm
3. tune model parameters
4. estimate generalization to unseen data following the *same distribution* as the training data
5. estimate generalization to unseen data following a *different distribution* than the training data.

Very often, one wants to do all of the above: Starting from a batch of data, one selects features, chooses a modeling algorithm and tunes free parameters. In the end one seeks to estimate the generalization performance including or excluding extrapolation.

The most common mistake is trying to estimate the generalization performance using data that has been used in model building. If any of the test data has been used in any of the model building steps (selecting features, algorithm and tuning parameters), estimates of generalization performance will invariably turn out too optimistic. This means that one needs to split the data into at least three sets: The *first set* is used to train models using different algorithms, feature sets and parameters. These different models are then applied to the *second* set of data. Based on the performance on this set, one selects one (!) model. This final model is then applied to the *third* set to estimate generalization performance. This is the exact point where mistakes



**Figure 3.5:** An illustration of nested cross-validation for evaluating a learning system. In an outer cross-validation loop, the overall data are partitioned randomly into  $K$  parts.  $K - 1$  of these splits are used as the training data. The training data are in turn split again, and used in an inner cross-validation loop. Model parameters are chosen such that the cross-validation estimate of the error rate in the *inner* loop is minimized. The resulting model is then evaluated on the test data in the *outer* loop.

are made, or phrasing this criticism differently: This is where acceptable experiments are interpreted in a misleading way. Often, multiple models are evaluated on the third set and the results are presented side by side, followed by an interpretation that focuses on the best model. It is easy to understand that this interpretation will lead to too optimistic results, if one considers an extreme case: One constructs models that generate completely random predictions. If one evaluates a large number of such models on any finite set of data, one finds a number of models that perform very well. These models are, however, very unlikely to perform well on any other set of data.

Following the three-set strategy outlined in the last paragraph has one drawback: Depending on the nature of the data, the chosen random split can have a big impact on model building (including selecting features, algorithm and free parameters) and, of course, on the estimated generalization performance. Therefore it is a good idea to investigate an (ideally large) number of different random splits of the data. In the case of a two set evaluation strategy (for example for just selecting a model), investigating many different splits is equivalent to leave k-percent out cross-validation. *Nested* cross-validation (see Figure 3.5 for an explanation) allows investigating many different splits of the data without losing the advantages of the three-set strategy outlined above: The procedure ensures that the generalization performance is estimated using data that has not been used in the construction of each respective model.

All above mentioned strategies estimate generalization to unseen data following the *same distribution* as the training data. This is no coincidence: As explained in Sec. A.1.2, most machine learning algorithms rely on this assumption. Strategies exist that allow achieving good results if training and test data follow different, but overlapping distributions, provided that the conditional distribution of target values given the input features

(descriptors) is the same in both test and training data. In the machine learning community, this scenario is known as *covariate shift* and corresponds to "mild extrapolation", see Sec. A.1.2 for details. The assumption of equal conditionals is often violated in drug discovery applications: In new projects, new compound classes may be investigated. These new compounds can exhibit new mechanisms of action (see Sec. A.1.3), possibly leading to complete failure of previously constructed models.

Imagine a set of measurements for 100 compounds, belonging to 10 structural classes. If we evaluate models on any single random split of the data or in a number of random splits (e.g. in leave k-percent out cross-validation) it is very likely that for every molecule in every test set, structurally similar compounds exist in the respective training set. To simulate the effect of new structural classes being introduced in new projects, one can generate splits where the distributions of test and training data differ substantially. One possible way to do this is leave one cluster out cross-validation, see [3, 4, 10, 103] for a description. Alternatively, one can take the temporal structure of the dataset into account, i.e. train on compounds measured before a given date and test on compounds that were measured after this date. If the evaluation is done by a group of people who were not involved in model building, using measurements that were not available to the modellers, this is called "blind evaluation". In [5–9, 11, 12] a combination of the two last-mentioned strategies was used to evaluate model performance. Models were constructed by researchers at Fraunhofer FIRST and Idalab. The final evaluation of each model was done by scientists at Bayer Schering Pharma, using new data that had accumulated over last couple of months, including compounds from new projects. This way of evaluating models is very similar to a strategy often referred to as "prospective evaluation". Here, one also applies the models to newly measured compounds. However, the new compounds were typically selected using the same model that is finally evaluated. Depending on the type of model, this can introduce a bias towards easily predictable compounds and will (as intended) introduce a bias towards compounds with desirable properties.

### 3.5 PERFORMANCE INDICATORS

In drug discovery & design, performance indicators (or loss functions) are used for a number of different purposes, including:

- steering the process of model fitting inside a learning algorithm
- guiding automatic tuning of hyperparameters of the learning algorithm
- allowing the modeller to assess the quality of a model
- communicating the quality of a model to non-experts

In the work leading up to this thesis, various standard regression, classification and ranking loss functions were used. This includes mean absolute error, mean squared error, root mean squared error, percentage of prediction within different intervals from the true value, correlation coefficients, Kendall's  $\tau$  and area under the ROC curve. Additionally, modified versions of standard loss functions and one completely new loss function were conceived.

#### *Hints for Practitioners*

The following subsection contains a somewhat informal collection of findings on loss functions that the author beliefs may be useful for practitioners in the field of chemoinformatics.

- Loss functions *inside learning algorithms* can be implicitly contained in the formulation of the optimization problem. One example is the standard formulation of Gaussian Processes, based on the assumption of Gaussian noise. This noise-model leads to an implicit squared error loss function. See paragraph "Relations to Support Vector Machines" on page 37.
- When using loss functions for tuning real valued *hyperparameters*, differentiable loss functions have the obvious advantage of enabling the use of gradient descend methods. If different local minima exist or when tuning hyperparameters expressed by integers, network architectures etc., grid search algorithms are a good choice. In this case, loss functions need not be differentiable, but are required to be informative with respect to the goal of the optimization.
- In the presence of *outliers*, the often used mean squared error loss-function can be misleading, because it is easily dominated by small

numbers of outliers. The mean absolute error is somewhat more robust in this respect, but in the presence of very extreme outliers it will also be misleading. Unless other ways of avoiding these far off predictions can be found (e.g. detection of outliers based on their features, see third subsection in 3.2), a simple way of avoiding their dominance in parameter tuning is to redefine the loss function to disregard a certain percentage of least accurate predictions.

- When assessing the quality of a model as a *human*, it is generally a good idea to not rely on a single number statistic. [104] illustrates how correlation coefficients can be misleading. Instead, one should look at plots such as scatterplots, receiver operating characteristic curves (ROC) and cumulative histograms.
- “Staircase plots” can be very useful for providing a *visual impression* of the quality of confidence estimates (see Figure 5.2 in Sec. 5.2 and Figure 5.5 in Sec. 5.3). If a single number statistic is desired, Kendall’s  $\tau$  [105] can be used.

#### *A New Loss Function for Virtual Screening*

It is important to use a loss function that represents the application scenario as closely as possible. When preparing a virtual screening for new PPAR $\gamma$  agonists (see Sec. 5.9) a new loss function was defined to this end: When performing the retrospective evaluation, it was known that the prospective evaluation would comprise experimentally investigating the activity of 20 or less compounds chosen out of a database containing more than 300.000 compounds. Including inactive compounds in the top 20 is very undesirable, whereas errors in the remaining bulk of predictions are irrelevant. Therefore the loss function “ $FI_{20}$ ” was defined as the fraction of inactive compounds in the top 20 ranked compounds. Additionally, standard regression performance indicators considering the whole set of test data were calculated to find out whether the newly introduced  $FI_{20}$  loss function actually results in choosing different models. Different models were chosen using different loss functions. The model chosen based on the  $FI_{20}$  loss function turned out to suggest the most promising candidate molecules according to a panel of human experts. The activity of a number of molecules was measured and a number of new PPAR $\gamma$  agonists were identified. Interestingly, the  $FI_{20}$  score for the actual experiments was almost identical to the score estimated during the prospective evaluation. See Sec. 5.9 for details.

### 3.6 INCORPORATING HETEROGENEOUS TYPES OF INFORMATION

Chemical molecules are commonly represented by their structural formula, sometimes referred to as the molecular graph. For the use in machine learning, one typically calculates vectorial features, so called chemical descriptors. These can then be used as inputs for almost all learning methods. Kernel based learning methods access the data exclusively via a kernel function. Therefore any type of data can be used, provided one can define a kernel function to handle it. A number of kernel functions for graphs, including molecular graphs, have been proposed, see [106] and references therein. These allow to directly use structural formulas in kernel based learning methods and have first been used with Support Vector Machines (see Sec. 3.3).

A linear combination of two kernels is again a kernel. By using different types of kernels defined on the same type of input, one can take different aspects of the same piece of information into account, see [107, 108] and references therein. By combining two kernels defined on different types of input, one can combine heterogeneous types of information. In Gaussian Process learning one can optimize the weighting of the different kernels by a gradient ascend on the evidence (see Sec. 3.3). Sec. 5.9 describes a first study using both a graph kernel based directly on the molecular graph and multiple standard kernels based on different sets of vectorial molecular descriptors. The successful application illustrates how the different types of information complement each other.

### 3.7 QUANTIFYING DOMAIN OF APPLICABILITY

A typical challenge for statistical models in the chemical space is to adequately determine the domain of applicability, i.e. the part of the chemical space where the model’s predictions are reliable. To this end several methods have been conceived. An informal graphical overview is presented in Figure 3.6, details are given in the following paragraphs and references therein.

*Range based methods* are based on checking whether descriptors of test set compounds exceed the range of the respective descriptor covered in training [109, 110]. A warning message is raised when this occurs. Also, *geometric methods* that estimate the convex hull of the training data can be used to further detail such estimates [111]. Mind that both these methods are not able to detect “holes” in the training data, that is, regions that are only scarcely populated with data.<sup>10</sup>

---

<sup>10</sup>Holes in the training data can, in principle, be detected using geometric methods in a suitable feature space. To the best of the author’s knowledge, there exists no published

<div> <div>methods</div> <div>advantages</div> </div>	range based	geometric	distance	density	ensemble	bayesian (GP)
indicate poor model fit						
applicable to small data sets						
easy to train and apply						
not comp. demanding						
output intuitively understandable						
reliable in holes in the training data						
reliable far from the training data						
reliable at boundaries of training data						

**Figure 3.6:** Different methods for producing individual confidence estimates for predictions have quite different advantages. Fully, partially or not possessing an advantage is indicated by green, yellow or red color, see [7] for details.

If experimental data for some new compounds are available, error estimates based on the *library approach* can be used. By considering the closest neighbors in the library of new compounds with known measurements, it is possible to get a rough estimate of the bias for the respective test compound. Alternatively, one can use this bias estimate to correct the prediction (see Sec. 4.2). This approach has recently been evaluated, see Sec. 5.7 and [4] for a discussion.

*Probability density distribution based methods* could, theoretically, be used to estimate the model reliability [111]. Still, high dimensional density estimation is recognized as an extremely difficult task, in particular since the behavior of densities in high dimensions may be completely counterintuitive [112]. Furthermore, regions where the model does not fit the data or where the labels of the training data are inconsistent can not be detected based on the density of training data.

*Distance based methods* and *extrapolation measures* [109, 111, 113–115] consider one of a number of distance measures (Mahalanobis, Euclidean etc.) to calculate the distance of a test compound to its closest neighbor(s) or the whole training set, respectively. Another way of using distance measures is to define a threshold and count the number of training compounds closer than study about this kind of approach.



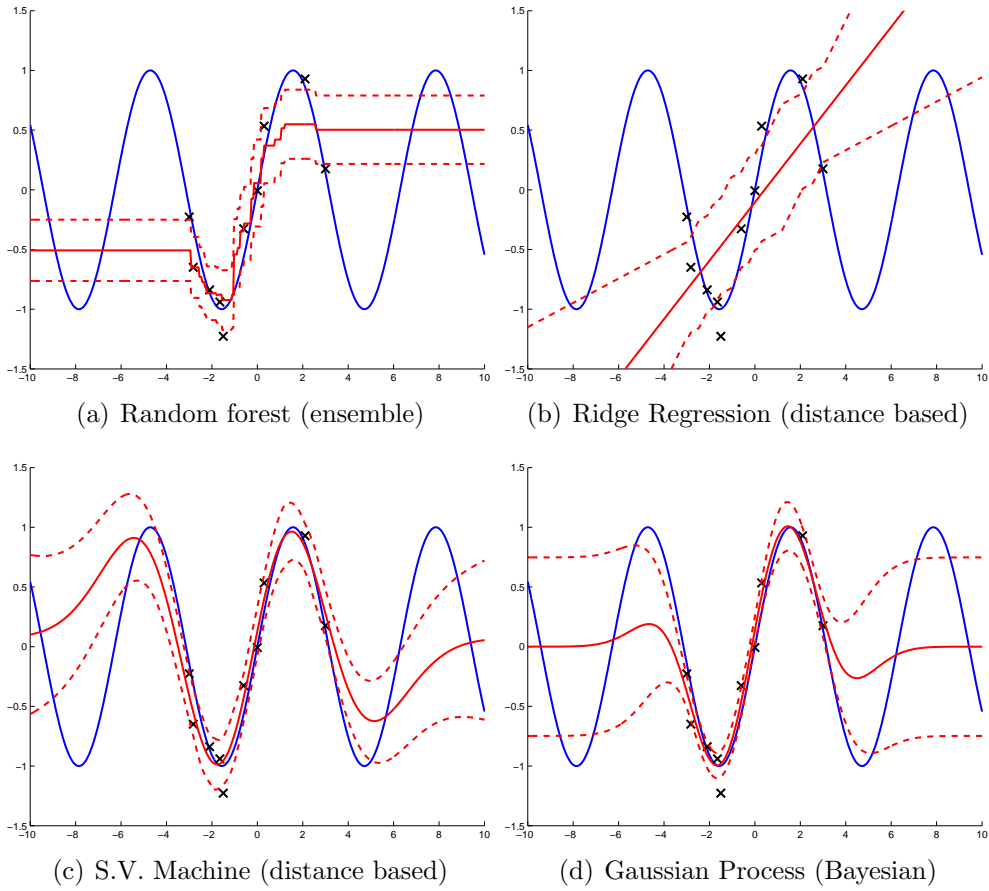
the threshold. Hotellings test or the leverage rely on the assumption that the data follows a Gaussian distribution in descriptor space and compute the Mahalanobis distance to the whole training set. Tetko correctly states in [114] that descriptors have different relevance for predicting a specific property and concludes, that property specific distances (resp. similarities) should be used<sup>11</sup>.

When estimating the domain of applicability with *ensemble methods*, a number of models is trained on different sets of data. Typically the sets are generated by (re)sampling from a larger set of available training data. Therefore the models will tend to agree in regions of the descriptor space where a lot of training compounds are available and will disagree in sparsely populated regions. Alternatively, the training sets for the individual models may be generated by adding noise to the descriptors, such that each model operates on a slightly modified version of the whole set of descriptors. In this case the models will agree in regions where the predictions are not very sensitive to small changes in the descriptors and they will disagree in descriptor space regions where the sensitivity with respect to small descriptor changes is large. This methodology can be used with any type of models, but ensembles of artificial neural networks (ANNs) [113–117] and ensembles of decision trees [4, 110, 113] (“random forests”, [101]) are most commonly used.

The idea behind *Bayesian methods* is to treat all quantities involved in modeling as random variables. By means of Bayesian inference, the *a priori* assumptions about parameters are combined with the experimental data, to obtain the *a posteriori* knowledge. Hence, such models naturally output a probability distribution, instead of the “point prediction” in conventional learning methods. Regions of high predictive variance not only indicate compounds outside the domain of applicability, but also regions of contradictory or scarce measurements. Gaussian Process regression and classification are popular examples of Bayesian methods, see Sec. 3.3.

Figure 3.7 shows a simple one-dimensional example of the four different methods of error estimation that were employed in [7]. The sine function (shown as a blue line in each subplot) is to be learned. The available training data are ten points marked by black crosses. These are generated by randomly choosing  $x$ -values and evaluating the sine function at these points. We simulate measurement noise by adding Gaussian distributed random numbers with standard deviation 0.2 to the  $y$ -values.

<sup>11</sup>There is an interesting parallel to Gaussian Process models: When allowing GP models to assign weights to each descriptor that enters the model as input, they explicitly construct a property specific distance measure and use it both for making predictions and for estimating prediction errors.



**Figure 3.7:** The four different regression models employed in [7] are trained on a small number of noisy measurements (black crosses) of the sine function (blue line). Predictions from each model are drawn as solid red lines, while dashed red lines indicate errors estimated by the respective model (in case of the Gaussian Process and random forest) or a distance based approach (in case of the Support Vector Machine and Ridge Regression model).

The random forest, Figure 3.7 (a), does provide a reasonable fit to the training points (yet the prediction is not smooth, due to the space dividing property of the decision trees). Predicted errors are acceptable in the vicinity of the training points, but overconfident when predictions far from the training points are sought. It should be noted that the behavior of error bars in regions outside of the training data depends solely on the ensemble members on the boundary of the training data. If the ensemble members, by chance, agree in their prediction, an error bar of zero would be the result.

The linear model, Figure 3.7 (b), clearly cannot fit the points from the non-linear function. Therefore, the distance based error estimations are mis-

leading: Low errors are predicted in regions close to the training points, but the actual error is quite large due to the poorly fitting model. This shows that the process of error estimation should not be decoupled from the actual model fitting: The error estimate should also indicate regions of poor fit.

The Support Vector Machine, Figure 3.7 (c), adapts to the non-linearity in the input data and extrapolates well. The error estimation (distance based procedure as described in Sec. 4.5 in [7]) produces slightly conservative (large) error bars in the region close the training points, and too small errors when extrapolating.

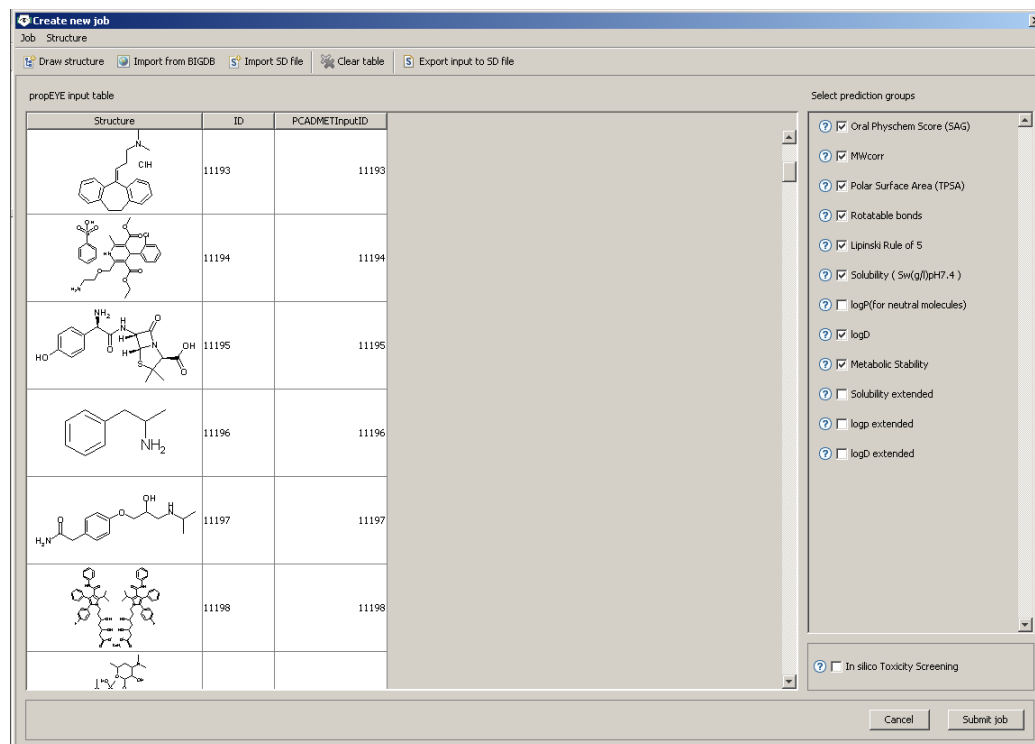
The Gaussian Process, Figure 3.7 (d) also captures the non-linearity in the input data and is able to extrapolate. Predicted errors are small in the region close to the training points and increase strong enough in the extrapolation region.

Gaussian Process based domain of applicability estimation is also discussed in Sec. 5.2 (partition coefficients) and Sec. 5.3 (aqueous solubility). Ensembles and distance based methods were also applied, for a discussion, see the respective journal publications on partition coefficients [7, 12] and aqueous solubility [8, 11]. Furthermore, predictive variances of Gaussian Process have been considered in Sec. 5.5 (Metabolic Stability) and in preparing hit-lists in a virtual screening for new PPAR-gamma agonists (Sec. 5.9).

A completely new approach to the domain of applicability question has been conceived for application in lead optimization (see Sec. 2.1). In this application scenario, human experts periodically decide which small group of molecules is going to be synthesized and investigated next. If a model could visually express how the prediction was constructed from the relevant (ideally small) part of the training data, the human expert could judge the applicability of the model by himself. For a description of this new type of technology, see Sec. 4.3.

### 3.8 PRESENTING RESULTS TO BENCH CHEMISTS

Scientists working in fields like chemoinformatics and computational chemistry tend to have a good knowledge of many different software tools, different operating systems and are often able to use (or sometimes even prefer) command line tools. On the other hand, bench chemists tend to focus on the synthetic aspects of chemistry, and are not always willing to spend a lot of time on learning how to use many different (possibly complicated) software tools. When trying to establish new computational methodology that can be relevant to the work of bench chemists, it is therefore necessary to adapt to their needs and preferences. This includes using the same units that they use



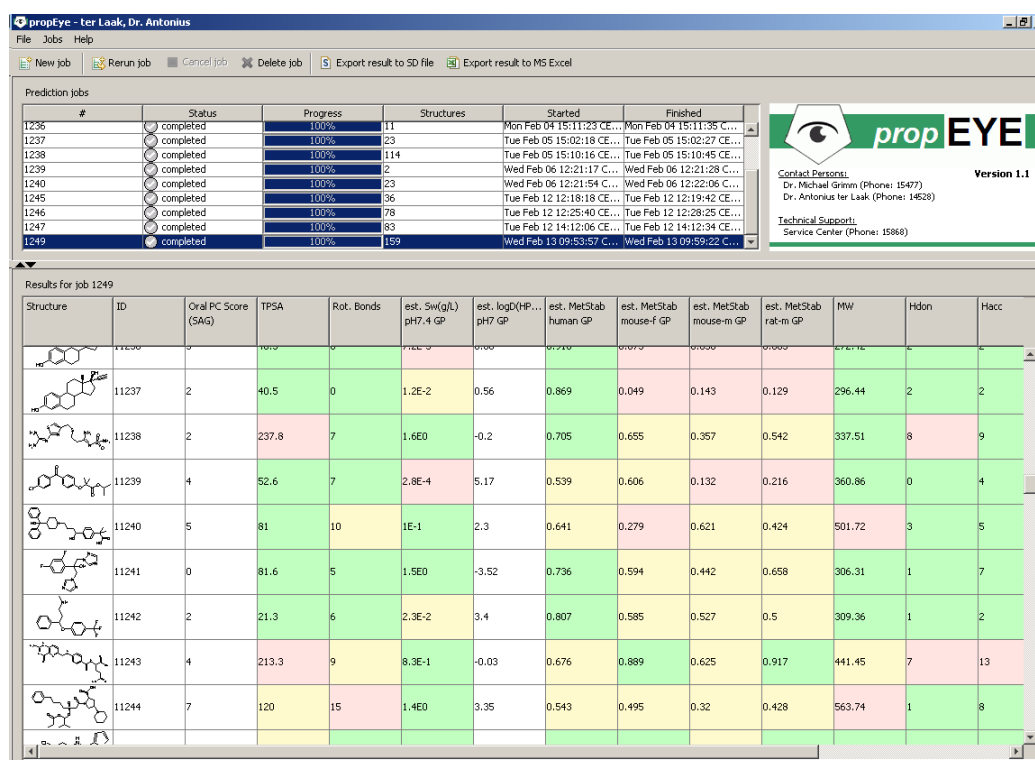
**Figure 3.8:** The propEYE toolbox provides an easy to use graphical user interface for submitting property calculation jobs to the backend-servers running the tools developed at Fraunhofer FIRST and Idalab.

in the lab as well as interfacing with their favorite molecule drawing programs and databases.

In a joint effort by the computational chemistry and chemoinformatics groups at Schering, an easy to use interactive graphical user interface called propEYE was developed. This section details how the models developed at Fraunhofer FIRST and Idalab (see Secs. 5.2 - 5.5) can be accessed using propEYE.

When creating a new prediction job using propEYE (see Figure 3.8), structures of molecules can be imported directly from the main corporate database of Schering or from local files in SD-Format. Alternatively, one can draw molecules using ISIS-draw and drag and drop them into the input-table. The checkboxes on the right hand side can be checked to ask for the corresponding property to be calculated.

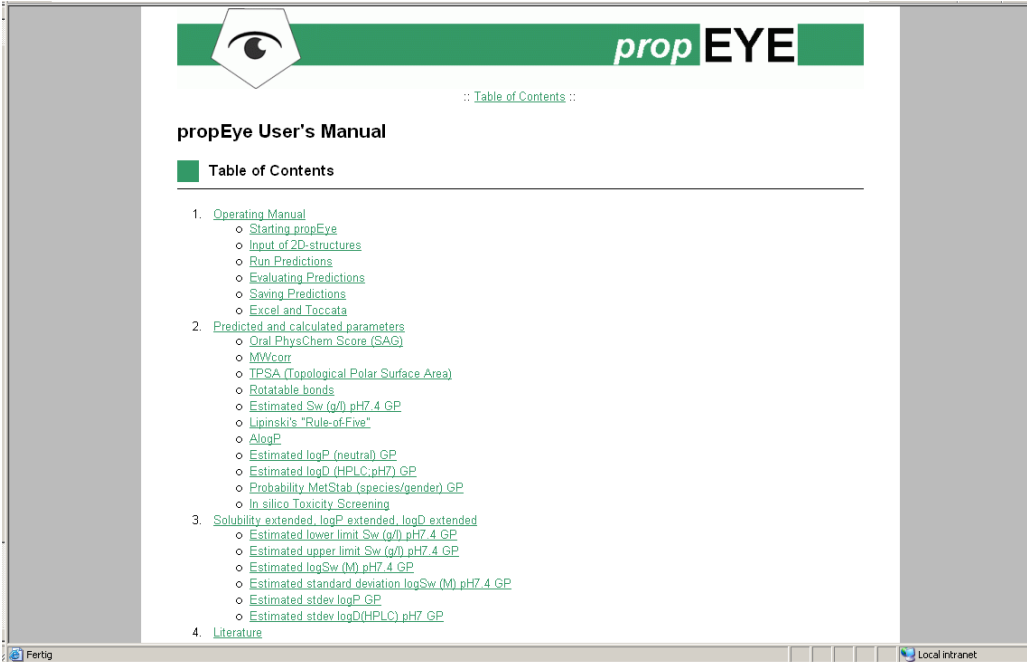
On the backend-servers, the Gaussian Process models developed at Fraunhofer FIRST and Idalab are implemented in the form of daemons. This has the advantage that the rather large inverted kernel matrices for the log D<sub>7</sub> models based on big sets of training data can be kept in memory. Conse-



**Figure 3.9:** The propEYE results window lists both running and completed jobs. Upon selecting a completed job, results are displayed in a table. Molecules are displayed as 2 D graphics, together with their ID and the different calculated properties. A traffic light coloring scheme indicates whether each property is already in the range that is desirable for drug molecules.

quently, the system can react very quickly. Small jobs are given priority, so interactive use of the system is possible even if properties for whole libraries of compounds are being calculated at the same time. The status of jobs in the queue is graphically indicated in the output window of propEYE, see Figure 3.9. Clicking a completed job makes the program display the results in the form of a table. The first column contains images of the 2 D-structures of all molecules. This is an important aspect, because it makes browsing much more comfortable for chemists. The next columns contain the compound ID and all calculated properties. The background of each cell is made either green, yellow or red, depending on whether the property displayed in the cell is already in the range desired for drug molecules (green), unacceptable (red) or in between (yellow). Units were chosen to match those that bench chemists use in the lab, e.g.  $\frac{mg}{ml}$  for aqueous solubility rather than the  $\frac{mol}{l}$  that is often used in computational studies.

The propEYE toolbox contains an extensive documentation of all fea-

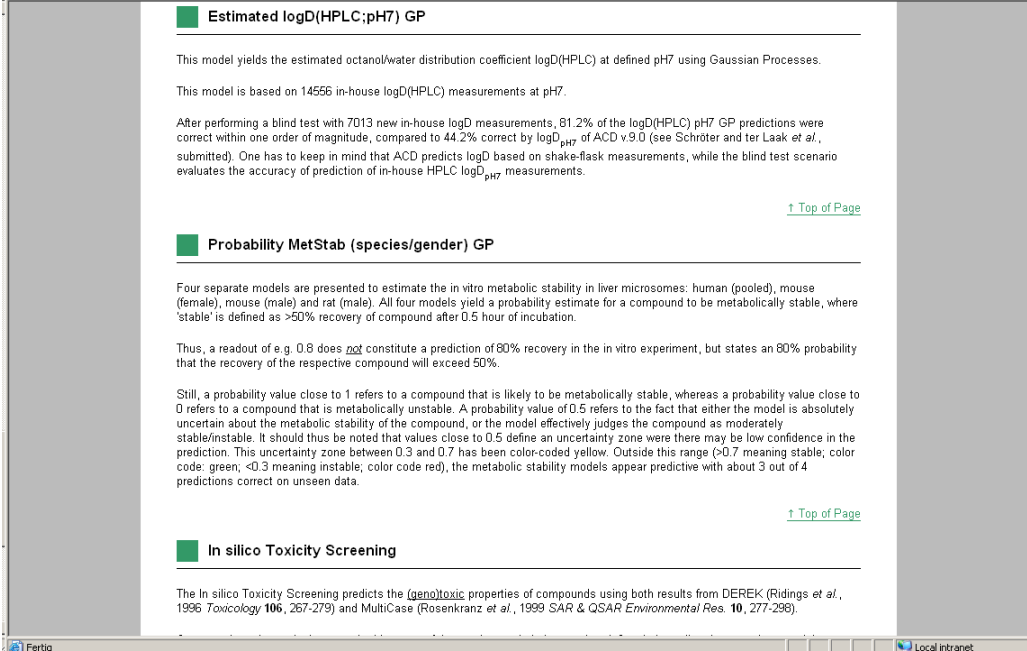


**propEYE User's Manual**

**Table of Contents**

- [Operating Manual](#)
  - [Starting propEYE](#)
  - [Input of 2D-structures](#)
  - [Run Predictions](#)
  - [Evaluating Predictions](#)
  - [Saving Predictions](#)
  - [Excel and Toccata](#)
- [Predicted and calculated parameters](#)
  - [Oral PhysChem Score \(SAG\)](#)
  - [MWcorr](#)
  - [TPSA \(Topological Polar Surface Area\)](#)
  - [Rotatable bonds](#)
  - [Estimated Sw \(g/l\) pH7.4 GP](#)
  - [Lipinski's "Rule-of-Five"](#)
  - [AlogP](#)
  - [Estimated logP \(neutral\) GP](#)
  - [Estimated logD \(HPLC; pH7\) GP](#)
  - [Probability MetStab \(species/gender\) GP](#)
  - [In silico Toxicity Screening](#)
- [Solubility extended, logP extended, logD extended](#)
  - [Estimated lower limit Sw \(g/l\) pH7.4 GP](#)
  - [Estimated upper limit Sw \(g/l\) pH7.4 GP](#)
  - [Estimated logSw \(M\) pH7.4 GP](#)
  - [Estimated standard deviation logSw \(M\) pH7.4 GP](#)
  - [Estimated stdev logP GP](#)
  - [Estimated stdev logD \(HPLC\) pH7 GP](#)
- [Literature](#)

Figure 3.10: Contents of the propEYE documentation.



**Estimated logD(HPLC;pH7) GP**

This model yields the estimated octanol/water distribution coefficient logD(HPLC) at defined pH7 using Gaussian Processes.

This model is based on 14556 in-house logD(HPLC) measurements at pH7.

After performing a blind test with 7013 new in-house logD measurements, 81.2% of the logD(HPLC) pH7 GP predictions were correct within one order of magnitude, compared to 44.2% correct by logD<sub>pH7</sub> of ACD v9.0 (see Schröter and ter Laak *et al.*, submitted). One has to keep in mind that ACD predicts logD based on shake-flask measurements, while the blind test scenario evaluates the accuracy of prediction of in-house HPLC logD<sub>pH7</sub> measurements.

[↑ Top of Page](#)

**Probability MetStab (species/gender) GP**

Four separate models are presented to estimate the in vitro metabolic stability in liver microsomes: human (pooled), mouse (female), mouse (male) and rat (male). All four models yield a probability estimate for a compound to be metabolically stable, where 'stable' is defined as >50% recovery of compound after 0.5 hour of incubation.

Thus, a readout of e.g. 0.8 does not constitute a prediction of 80% recovery in the in vitro experiment, but states an 80% probability that the recovery of the respective compound will exceed 50%.

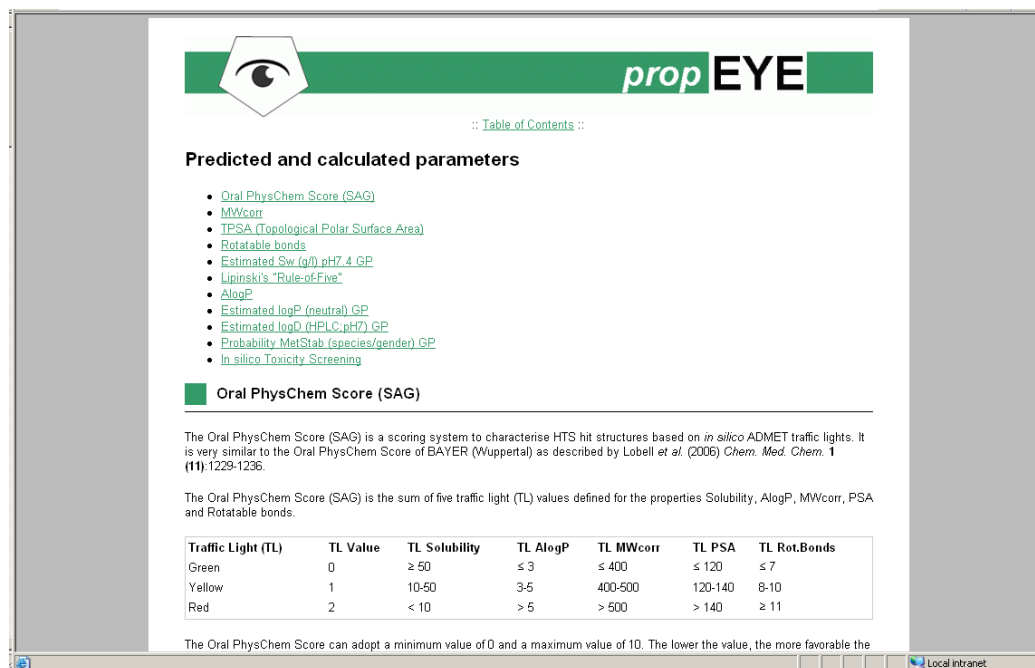
Still, a probability value close to 1 refers to a compound that is likely to be metabolically stable, whereas a probability value close to 0 refers to a compound that is metabolically unstable. A probability value of 0.5 refers to the fact that either the model is absolutely uncertain about the metabolic stability of the compound, or the model effectively judges the compound as moderately stable/unstable. It should thus be noted that values close to 0.5 define an uncertainty zone where there may be low confidence in the prediction. This uncertainty zone between 0.3 and 0.7 has been color-coded yellow. Outside this range (>0.7 meaning stable; color code: green; <0.3 meaning unstable; color code red), the metabolic stability models appear predictive with about 3 out of 4 predictions correct on unseen data.

[↑ Top of Page](#)

**In silico Toxicity Screening**

The In silico Toxicity Screening predicts the (geno)toxic properties of compounds using both results from DEREK (Ridings *et al.*, 1996 *Toxicology* **106**, 267-279) and MultiCase (Rosenkranz *et al.*, 1999 *SAR & QSAR Environmental Res.* **10**, 277-298).

Figure 3.11: The propEYE documentation contains background information on every underlying model.



**Predicted and calculated parameters**

- [Oral PhysChem Score \(SAG\)](#)
- [MWcorr](#)
- [TPSA \(Topological Polar Surface Area\)](#)
- [Rotatable bonds](#)
- [Estimated Sw \(g/l\) pH7.4 GP](#)
- [Lipinski's "Rule-of-Five"](#)
- [AlogP](#)
- [Estimated logP \(neutral\) GP](#)
- [Estimated logD \(HPLC, pH7\) GP](#)
- [Probability MetStab \(species/gender\) GP](#)
- [In silico Toxicity Screening](#)

**Oral PhysChem Score (SAG)**

The Oral PhysChem Score (SAG) is a scoring system to characterise HTS hit structures based on *in silico* ADMET traffic lights. It is very similar to the Oral PhysChem Score of BAYER (Wuppertal) as described by Lobell *et al.* (2006) *Chem. Med. Chem.* **1** (11):1229-1236.

The Oral PhysChem Score (SAG) is the sum of five traffic light (TL) values defined for the properties Solubility, AlogP, MWcorr, PSA and Rotatable bonds.

Traffic Light (TL)	TL Value	TL Solubility	TL AlogP	TL MWcorr	TL PSA	TL Rot.Bonds
Green	0	≥ 50	≤ 3	≤ 400	≤ 120	≤ 7
Yellow	1	10-50	3-5	400-500	120-140	8-10
Red	2	< 10	> 5	> 500	> 140	≥ 11

The Oral PhysChem Score can adopt a minimum value of 0 and a maximum value of 10. The lower the value, the more favorable the

**Figure 3.12:** The propEYE documentation includes links to research papers on the respective properties.

tures and even includes research papers written about the different models implemented on the backend-servers, see Figs. 3.10, 3.11 and 3.12. In case any questions remain open, support telephone numbers are displayed in every application window.





## CHAPTER 4

# NEWLY DEVELOPED METHODS

### 4.1 OVERVIEW

Three new algorithms were developed to cope with the specific requirements of lead optimization, the most challenging part of the drug discovery process. The first new algorithm can *improve* the *accuracy* of models in the early stages of lead optimization projects. The second new algorithm can *explain individual predictions* made by complex models to human experts and the third new algorithm generates *hints for compound optimization*.

Imagine a setting where human experts have access to multiple predictive models (including commercial black box models) and the *first measurements* of compounds from a certain *new compound class* (e.g. part of the chemical space) are available. How can one obtain the best possible predictions from all these sources of knowledge simultaneously? Sec. 4.2 introduces several *ensemble modeling approaches* where the most recent measurements are used to either select the best model from the ensemble, construct a suitable weighting of multiple models in the ensemble or even correct a bias that one (or more) ensemble members may consistently exhibit when making predictions for compounds that are similar to certain test compounds.

Two separate methodologies for explaining individual predictions of (possibly non-linear) machine learning models are presented. The method presented in Sec. 4.3 *explains predictions by* the means of *visualizing* relevant *objects* from the training set of the model. This allows human experts to understand how each prediction comes about. If a prediction conflicts with his intuition, the human expert can easily find out whether the grounds for the models predictions are solid or if trusting his own intuition is the better idea [2].

The method presented in Sec. 4.4 utilizes *local gradients* of the models predictions to explain predictions in terms of the locally most relevant features. This not only teaches the human expert which features are relevant for each individual prediction, but also gives a *directional information*. Abstractly speaking, one can learn in which direction a data point has to be moved to increase the prediction for the target value [1]. In the context of

lead optimization, this means that the human expert can obtain a type of *guidance in compound optimization*.

#### 4.2 INCORPORATING ADDITIONAL MEASUREMENTS

In drug discovery & design, modellers typically have access to a number of predictive tools for some of the relevant properties. Tools may have been bought from vendors or trained in-house. For each of these tools, the respective training set may or may not be known, the algorithm used in training each model may or may not be well documented. If re-trainable in-house tools are available, one often has access to measurements for additional compounds that have become available after they have been trained last, and that are not included in any of the commercial tools. An idea originally published by Kühne et al. [118] comprises not just retraining the in-house tools including all new compounds. Instead, they used new compounds to choose one out of a whole collection of tools that will probably predict a certain unknown compound most accurately. The set of new compounds is called "correction set", and models are evaluated in a leave one compound out cross-validation setting where each compound from the correction set is in turn left out and treated as new. The following section summarizes the selection approach conceived by Kühne et al. and several ideas going beyond the original publication. An evaluation of these ensemble modeling algorithms is presented in Sec. 5.7.

**Selection by MAE (MAE Model)** In this approach the single model with the **lowest mean absolute error on the neighboring compounds** in the correction set is selected to predict the desired value for the unknown compound. The calculation of the mean absolute error (MAE) on the  $k$  nearest neighbors is based on the distance measure introduced in [4]:

$$\text{MAE}(f_i) = \frac{1}{k} \sum_{j=1}^k |f_i(\mathbf{x}_j) - y_j|. \quad (4.1)$$

Here  $f_i$  refers to one of the  $l$  trained single models. Chemical compounds are represented by descriptor vectors  $\mathbf{x}_j$  and the property to be predicted is written as  $y_j$ . The predicted value  $f^*(\mathbf{x}_t)$  of this ensemble model is given by

$$f^*(\mathbf{x}_t) = f_{\min\text{MAE}}(\mathbf{x}_t) \quad \text{with} \quad f_{\min\text{MAE}} = \underset{f_i \ i=1,\dots,l}{\operatorname{argmin}}(\text{MAE}(f_i)). \quad (4.2)$$

This is the type of algorithm originally proposed by Kühne et al. [118].

**Weighted Model** This model is based on the idea that a **weighted sum of all predictions** of the different single models can result in a greater improvement than selecting the prediction of only one model. The mixing coefficient of each model  $v_{f_i}$  is calculated according to the mean absolute error of the model on the neighboring compounds in the correction set:

$$v_{f_i} = \frac{1}{\text{MAE}(f_i)} \left( \sum_{j=1}^l \frac{1}{\text{MAE}(f_j)} \right)^{-1} \quad \text{such that} \quad \sum_{i=1}^l v_{f_i} = 1 \quad (4.3)$$

and the predicted value of the weighted model can be written as

$$f^*(\mathbf{x}_t) = \sum_{i=1}^l v_{f_i} f_i(\mathbf{x}_t). \quad (4.4)$$

The higher the accuracy of a single model  $f_i$  in the neighborhood of the unknown compound  $t$  the greater the impact of the model  $f_i$  on the prediction of the weighted model.

**Bias Corrected Model** In this approach first one single model is selected with respect to the mean absolute error on the neighboring compounds, identically to the MAE model. From the prediction of the selected model one **subtracts the mean error** on the neighbors in the correction set. To incorporate the distance between the unknown compound  $t$  and its neighbors we define a *distance weight* for each of the  $k$  nearest neighbors as

$$d_j = \frac{1}{\|\mathbf{x}_t - \mathbf{x}_j\|_{red}} \left( \sum_{i=1}^k \frac{1}{\|\mathbf{x}_t - \mathbf{x}_i\|_{red}} \right)^{-1} \quad j = 1, \dots, k. \quad (4.5)$$

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

$$f_{\text{weightedDist}} = \underset{f_i}{\operatorname{argmin}} \left( \frac{1}{k} \sum_{j=1}^k \frac{\|f_i(\mathbf{x}_j) - y_j\|_{red}}{d_j} \right). \quad (4.6)$$

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

$$f^*(\mathbf{x}_t) = f_{\text{weightedDist}}(\mathbf{x}_t) - \frac{1}{k} \sum_{j=1}^k \frac{f_{\text{weightedDist}}(\mathbf{x}_j) - y_j}{d_j}. \quad (4.7)$$

**Average KNN Model and Random Choice Model** These models are baselines to determine the amount of improvement achieved by applying ensemble models. In the Random Choice Model the prediction of one standard model is chosen **randomly** as the predicted value.

Unlike all other models, the Average Model predicts the value for the unknown compound without considering the prediction of the single models. The prediction is the **average of the true values** of the neighboring compounds

$$f^*(\mathbf{x}_t) = \frac{1}{k} \sum_{j=1}^k y_j. \quad (4.8)$$

Results of an evaluation of these algorithms for combining predictions from individual models in the context of the hERG channel blockade effect are described in Sec. 5.7. Details and background information can be found in [4].

### 4.3 EXPLAINING INDIVIDUAL PREDICTIONS USING OBJECTS FROM THE TRAINING SET

In this thesis, two separate methodologies for explaining individual predictions of machine learning models are presented. The method presented in this section *explains* predictions by the means of visualizing relevant objects from the training set of the model. This allows human experts to understand how each prediction comes about. If a prediction conflicts with his intuition, the human expert can easily find out whether the grounds for the models predictions are solid or if trusting his own intuition is the better idea [2].

The method presented in Sec. 4.4 utilizes local gradients of the model’s predictions to explain predictions in terms of the locally most relevant features. This not only teaches the human expert which features are relevant for each individual prediction, but also gives a directional information. Abstractly speaking, one can learn in which direction a data point has to be moved to increase the prediction for the target value [1]. In the context of lead optimization, this means that the human expert can obtain a type of *guidance in compound optimization*. For this reason, the two explaining-related methodologies are presented in separate sections.

#### 4.3.1 Motivation

Let us consider *lead optimization* (see Sec. 2.1). In this application scenario for machine learning models, human experts periodically decide which *small batch of molecules* is going to be synthesized and investigated next. Models can support humans in making such decisions by providing accurate predictions of the relevant properties of the compounds under consideration. However, human experts are not likely to trust a model if its prediction deviates from their own intuition. This deviation might occur for any of these reasons:

1. The models prediction is correct.
  - (a) Its training data includes relevant information that the expert doesn’t know yet.
  - (b) It has generalized from known data in a valid way that is not obvious to the human expert.
  - (c) It has generalized from known data in an invalid way that is not obvious to the human expert, i.e. the correct prediction was a lucky strike.

2. The human experts intuition is correct. The models predictions is wrong because
- (a) The local density of training compounds is not high enough.
  - (b) The local labeling of training compounds is correct, but inconsistent.
  - (c) The local labeling of training compounds is wrong/inaccurate.
  - (d) The features (vectorial descriptors, 2D-graphs ...) given to the learning algorithm do not represent the molecules sufficiently well.
  - (e) The internal representation of compounds in the learning algorithm (e.g. the kernel or covariance function) does not capture the relevant bits of information.
  - (f) Hyperparameter selection (if applicable) led to non-optimal parameters.
  - (g) Learning (model fitting) led to a non-optimal model.

If a model could *visually express* how each prediction was constructed from a small number of most relevant compounds in the training data, the human expert could easily recognize case 1(a). In the following, such visualizations will be called *explanations*. By definition, the human expert cannot distinguish between cases 1(b) and 1(c). To be on the safe side, he should therefore trust his intuition and not believe the models predictions in cases 1(b,c) and 2(a-g). If he learns from the models *explanations* that case 1(a) applies, his intuition and the model prediction will be in sync again, because he has learned some decision-relevant new facts from the model.

#### *Relation to Confidence Estimates*

Individual confidence estimates, e.g. predictive variances from Gaussian Process Models (see Sec. 3.3) can, in principle, help to avoid inaccurate predictions, because they are often made with lower confidence than correct predictions (see Sec. 3.7). Considering the different cases defined above, predictive variances can help to detect cases 2(a) and 2(b). The remaining cases 2(c-g) can, by definition, not be detected by the model itself. External tools, such as local density estimators, outlier indices etc. may use different (internal) representations of the data and may be able to detect potentially unreliable predictions where the model itself predicts confidently. However, the principle difficulties listed in cases 2(c-g) apply to all conceivable external tools in an analog way.

#### 4.3.2 *Prerequisites for Obtaining helpful Explanations*

The key hypothesis introduced in the motivational first part of this section is that sometimes (specifically: case 1(a) in the list above) a human expert’s intuition and a model’s prediction do not match, because the models training data includes relevant information that the expert doesn’t know yet. In this case, a visualization of the relevant part of the training data is useful if the human expert agrees that the compounds (and metadata) presented as *explanations* are indeed relevant to the prediction, because he can then learn something new. Consequently, his intuition and the models prediction finally match again.

Therefore, the model’s *explanations* for a certain prediction will be useful if the following criteria are met:

1. The prediction is correct.
2. The prediction is based on few enough training compounds to allow the human expert to grasp the visualization.
3. The prediction is based on training compounds that are similar to the test compound (according to the perception of the user)
4. At least some of these relevant training compounds (or their labels) are new to the user.

The following paragraphs present how a study can be set up such that all four requirements are satisfied. The discussion starts with the easiest requirements (no. 4 and no. 1) and then proceeds to the more difficult parts (no. 2 and no. 3).

The fourth requirement can, for the purpose of this present investigation, be satisfied by making the assumption that any compound might be unknown to some human expert who might want to use the system.

To satisfy the first requirement, one can generate predictions for held-out test-compounds with known labels and then choose correctly predicted test-compounds for in depth investigation.

The second requirement can be made more specific by consulting the psychological literature. As originally established by Miller [119] and later confirmed by many other researchers, humans can efficiently deal with up to  $7 \pm 2$  items simultaneously. In this study, the  $k$ -nearest-neighbor classifier (KNN) will be used as a baseline method. KNN-predictions are always based on odd numbers of training compounds. Considering that the user will have to simultaneously consider the test compound in question together with all training compounds relevant to this prediction, choosing  $k = 7$  would result

in  $7 + 1 = 8$  items, i.e. demanding above average cognitive skills on the part of the user. To be on the safe side while establishing the usefulness of *explanations*, the constraint  $k \leq 5$  was chosen for this study, resulting in a maximum of  $5 + 1 = 6$  items. Unlike KNN, more complex machine learning methods do not have a parameter that limits how many training points will be taken into account. Furthermore, this number may vary depending on the test points for which predictions are sought. Hence, a subsection of Sec. 4.3.4 is devoted to the question how complex machine learning algorithms can be modified to operate sufficiently local.

The last requirement depends (by definition) on the user. Ideally, one would perform an evaluation with a large number of human experts. However, this type of investigation is out of the scope of this work. The initial evaluation presented in the following sections was performed by the author of this thesis and is considered a starting point for further investigations.



### 4.3.3 Concepts & Definitions

If unlimited computational resources were available, explanations could be generated for any machine learning model using a brute force approach along these lines:

1. Train a model using the full set of training data.
2. Perform leave-one-out cross-validation on the training data and keep all models.
3. Use each model to generate predictions for the test-compound for which *explanations* are sought.
4. Sort the leave-one-out models by how much the prediction for the test compound varies when compared to the prediction from the model that was trained on the full training set. The compounds left out by the models at the top of the list cause the largest changes in prediction when left out and are therefore the most relevant part of the training set.

As discussed in more detail in the next subsection, *explanations* generated in this way can be useful if the machine learning algorithm operates sufficiently local. Applying the same algorithm to global models (like linear regression) would yield the same *explanations* for any conceivable test point. Therefore, they cannot lead the user to new insights by pointing out the training compounds that are most relevant to each individual test compound.

In the special case of the k-nearest-neighbor classifier one can directly use the k nearest neighbors of each test-point to generate explanations.

Kernel methods like support vector machines for classification (SVM) and regression (SVR), Gaussian Process regression (GPR) and kernel ridge regression (KRR) all rely on the following general equation to calculate predictions [120].

$$f(\mathbf{x}) = \sum_{i=1}^n \alpha_i k(\mathbf{x}_i, \mathbf{x}_*) + b \quad (4.9)$$

Depending on the learning method,  $n$  is either the number of support vectors (for sparse methods like SVM, SVR) or the number of all training compounds (in case of GPR, KRR). The weights  $\alpha_i$  are optimized during model fitting and  $k(\mathbf{x}_i, \mathbf{x}_*)$  denotes the kernel function between the respective support vector / training compound  $\mathbf{x}_i$  and the test compound  $\mathbf{x}_*$ .

Predictions made by Gaussian Process Classification (GPC) models include the above expression  $f(\mathbf{x})$ , but additionally take the predictive variance

into account. Let  $\text{var}_f(x_0)$  be the variance of  $f(x_0)$  under the GP posterior for  $f$ . The probability for being of the positive class  $p(x_0)$  predicted by GPC can be shown to be (see Equation 6 in [121])

$$p(x_0) = \frac{1}{2} \text{erfc} \left( \frac{-f(x_0)}{\sqrt{2} * \sqrt{1 + \text{var}_f(x_0)}} \right),$$

with  $\text{erfc}$  being the complementary error function.

For all kernel methods mentioned above, the contribution of each training point can therefore be calculated analytically. The vector of contributions  $\beta_t$  of each training point to the prediction for a specific test point  $x_t$  is hereby defined as follows,

$$\begin{aligned} f(x_t) &= K^* \alpha \\ &= K^* (K + I\sigma)^{-1} y \\ &= \beta_t y_t \end{aligned}$$

with  $K$  symbolizing the kernel matrix of the training data,  $K^*$  as the kernel matrix between the training and the test data,  $\sigma$  as the learned noiselevel and  $y$  symbolizing the vector of training labels.

The normalized contribution of each of the  $n$  training points to the prediction  $f(x_t)$  can then be written as:

$$\hat{\beta}_t := \frac{|K^*(K + I\sigma)^{-1}|}{\sum_{i=1}^n \beta_{t,i}} \quad (4.10)$$

Learning Algorithm	AUC	Error rate
GPC RBF	83.7 %	23.2 %
GPC ISOAK	82.2 %	25.2 %
KNN ( $k = 5$ )	66.8 %	33.7 %

**Table 4.1:** Area under the receiver operating characteristic curve (AUC) and error rate for a k-nearest-neighbor classifier (KNN), Gaussian Process Classification (GPC) utilizing the radial basis function (RBF) kernel and GPC with the ISOAK molecular graph kernel.

#### 4.3.4 Examples & Discussion

##### *Data and Learning Algorithms*

The Ames mutagenicity benchmark dataset introduced in Sec. 5.6 is used to investigate the usefulness of *explanations*. From the viewpoint taken for this particular purpose, the modeling task can be summarized as follows:

- supervised classification task
- dataset: 6512 molecules with labels for Ames toxicity (see Sec. 5.6).
- training set: 1000 randomly chosen compounds, including 203 compounds listed in the world drug index (WDI) [122].
- test set: 4512 randomly chosen compounds, including 1213 compounds listed in the WDI.
- data representations: 904 dimensional vectors (a subset from the Dragon descriptors [123] as chosen previously, see Sec. 5.6) serve as input for GPC with a radial basis function kernel (RBF), 200 out of these 904 feature dimensions are used by KNN<sup>1</sup> and molecular graphs serve as input for the ISOAK molecular graph kernel [106].
- learning algorithms investigated for predicting: See Sec. 5.6 for results obtained using Support Vector Machines, Gaussian Process Classification, Random Forests and  $k$ -Nearest-Neighbor
- learning algorithms investigated for *explaining*: Gaussian Process Classification and  $k$ -Nearest-Neighbor

---

<sup>1</sup>Features were automatically selected using the p-values for the hypothesis of linear correlation of each feature with the label as determined in a permutation test using 1000 random permutations of the labels.

Table 4.1 lists the error rate and area under the receiver operating characteristic curve (AUC) for the models applied in this study. Results obtained using Gaussian Process Classification (GPC) with RBF kernel are on par with previous results (see Sec. 5.6), despite the fact that in the last study a much larger fraction of all compounds was used to train the models.<sup>2</sup> KNN performs significantly worse when considering AUC. One reason may be that KNN does not interpolate as well as GPC. A second reason is that while being a useful single number statistic for classifiers with real valued output (like SVM and GPC), AUC can be misleading when being applied to KNN with small  $k$ .<sup>3</sup> The newly introduced GPC with the ISOAK molecular graph kernel performs almost as well as GPC with RBF kernel when considering both AUC and error rate.

#### *Initial Explanations & Missing Locality of GPC models*

Using these models, initial experiments on generating *explanations* were performed. In line with the previously presented motivation (Sec. 4.3.1) and prerequisites (Sec. 4.3.2), the ten most confidently made predictions for WDI compounds<sup>4</sup> made by each model are identified. Table 4.2, 4.3 and 4.4. list these ten test compounds together with the five most relevant compounds from the respective training set (*explanations*).

Inspecting the test compounds predicted most confidently by the KNN model with  $k = 5$  (Table 4.2), one finds that they are structurally not very similar to the training compounds presented. Therefore, *explanations* generated in this way will not be convincing in the eyes of a human expert utilizing this model.<sup>5</sup>

Next, the ten test compounds predicted most confidently by the Gaussian Process Classification (GPC) model utilizing the ISOAK molecular graph kernel are investigated (Table 4.3). As one would expect when using a graph kernel, the predictions are based on structurally very similar molecules in the

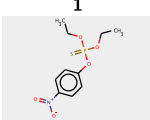
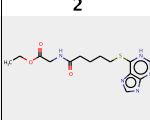
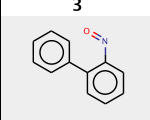
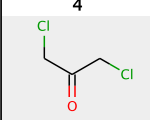
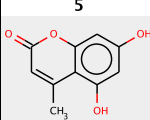
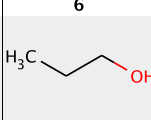
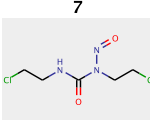
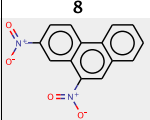
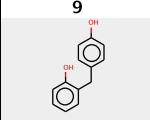
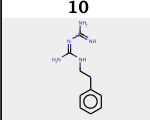
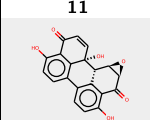
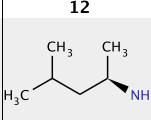
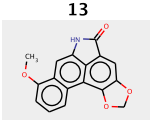
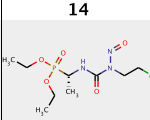
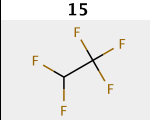
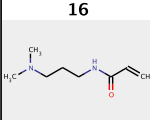
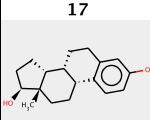
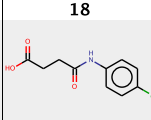
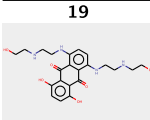
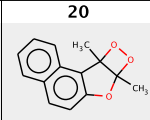
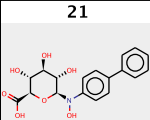
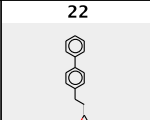
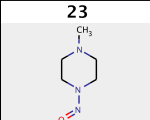
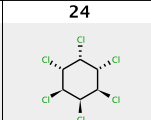
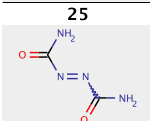
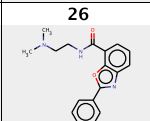
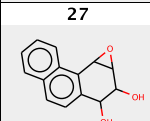
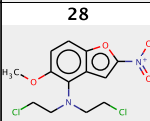
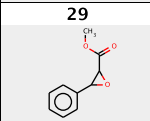
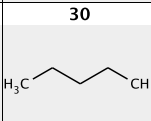
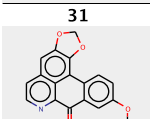
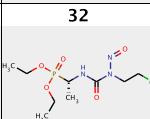
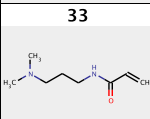
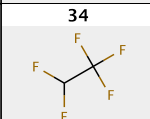
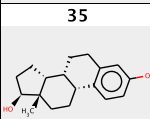
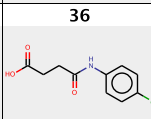
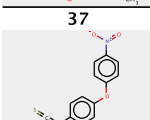
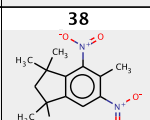
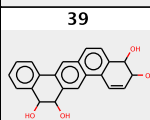
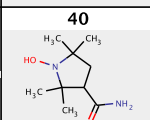
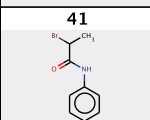
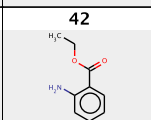
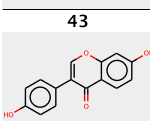
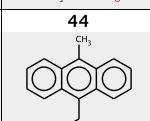
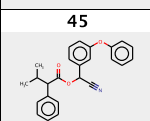
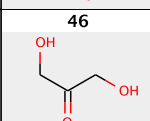
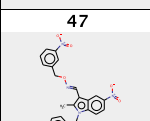
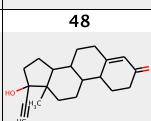
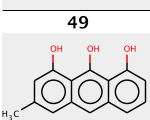
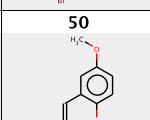
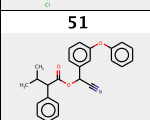
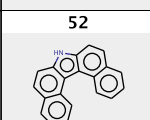
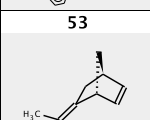
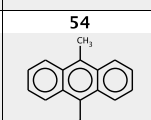
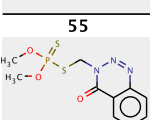
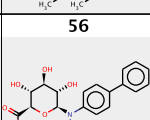
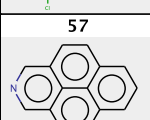
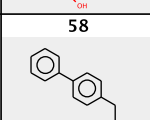
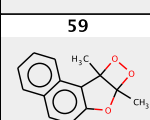
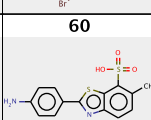
---

<sup>2</sup>Previously, a 5-fold cross-validation was performed with additional compounds in a static training set. Therefore, more than 5000 compounds were used in training each model and only  $\approx 1000$  compounds were used as respective test set.

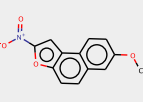
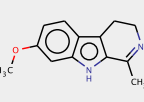
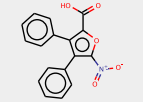
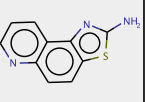
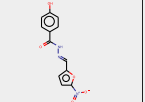
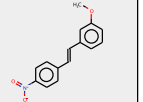
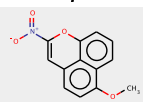
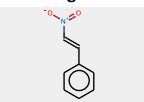

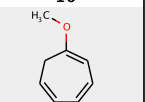
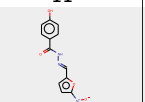
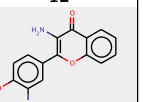
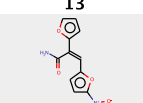
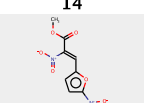
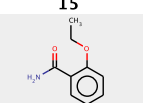
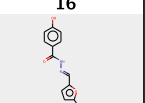
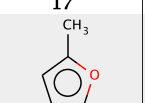
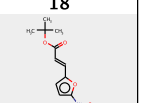
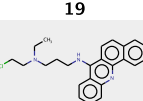
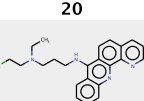
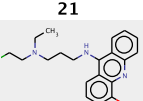
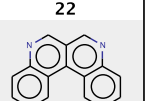
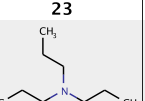
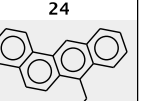
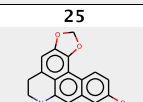
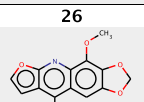
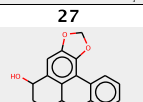
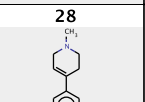
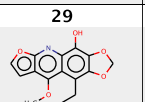
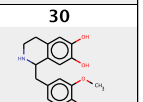
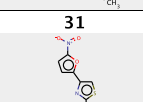
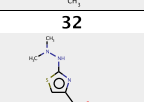
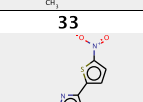
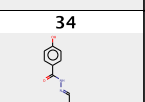
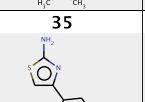
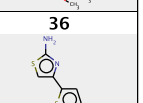
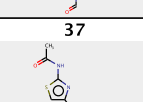
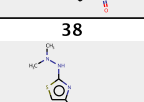
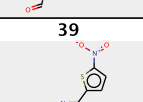
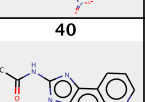
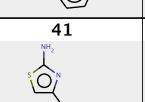
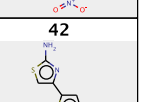
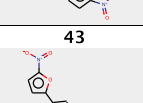
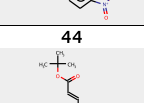
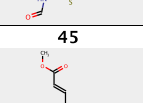
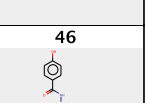
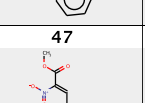
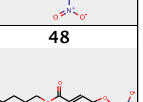
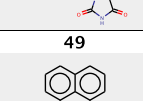
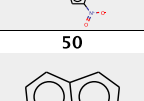
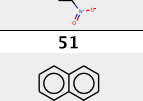
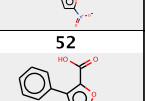
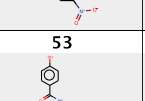
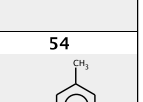
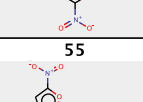
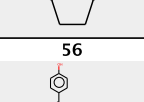
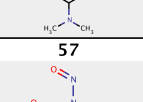
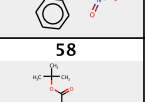
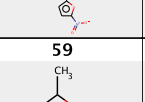
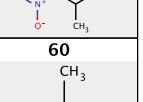
<sup>3</sup>AUC can be misleading when being applied to KNN with small  $k$  because AUC is originally an indicator of ranking performance. KNN generates a real valued output, but only  $2k + 1$  different values (voting outcomes in the range  $[-k \dots k]$ ) are possible. Among test compounds with the same voting outcome, the order is random, which tends to result in low AUCs for small values of  $k$ .

<sup>4</sup>When feasible, WDI listed test compounds are chosen for visual inspection, because they are the most relevant type of compound.

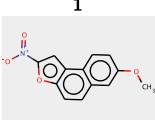
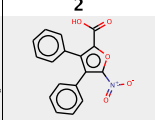
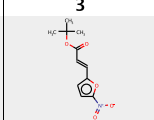
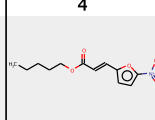
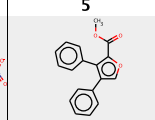
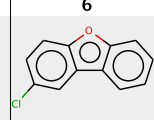
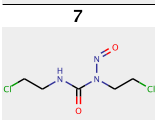
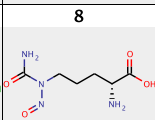
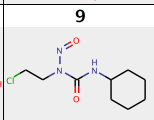
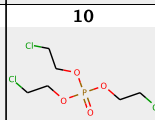
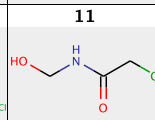
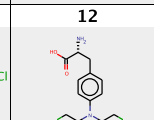
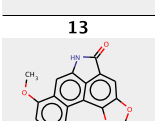
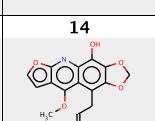
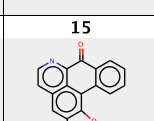
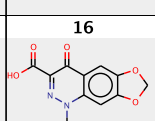
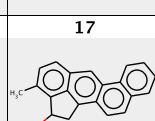
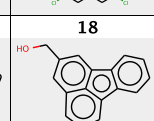
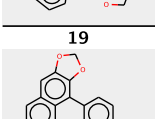
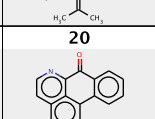
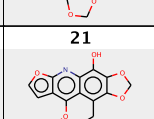
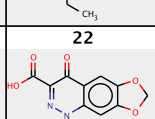
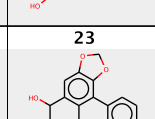
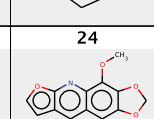
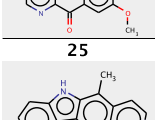
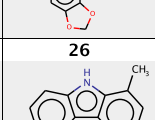
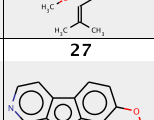
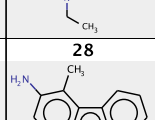
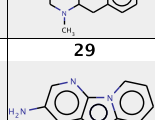
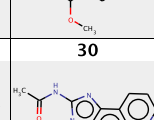
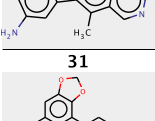
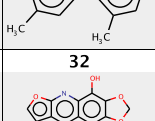
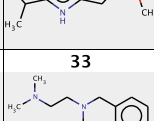
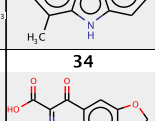
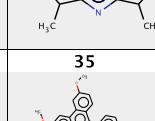
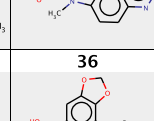
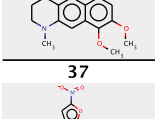
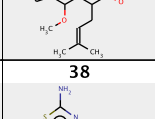
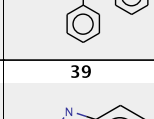
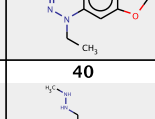
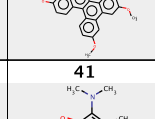
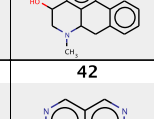
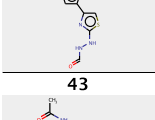
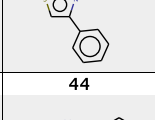
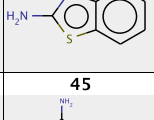
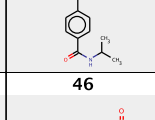
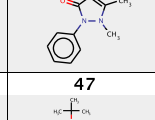
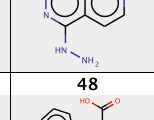
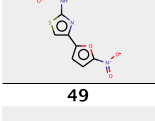
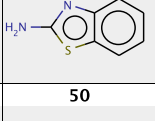
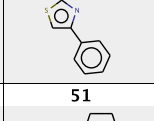
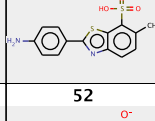
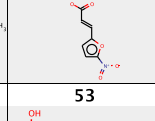
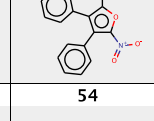
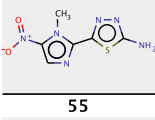
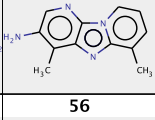
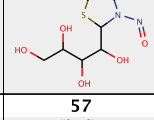
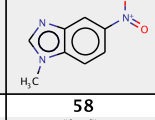
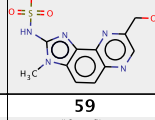
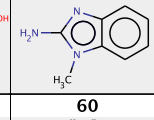
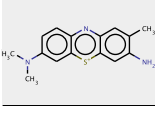
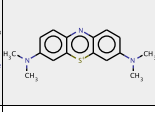
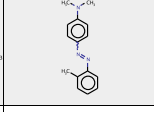
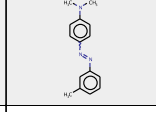
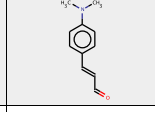
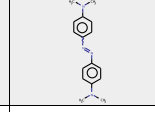
<sup>5</sup>*Explanations* identified by KNN models can potentially be improved by using different distance functions and vectorial data representations.

test mol.   <i>explanations</i> (most relevant molecules in training set)					
					
					
					
					
					
					
					
					
					
					

**Table 4.2:** The left-most column lists ten of the test compounds predicted most confidently by the KNN model with  $k = 5$ . The remaining five columns contain the *explanations*, i.e. in this case the five nearest neighbors from the training set. Structurally these neighbors are not very similar to the test compounds in the leftmost column. Therefore, they will not be convincing in the eyes of a human expert utilizing this model.

test mol.   <i>explanations</i> (most relevant molecules in training set)					
					
					
					
					
					
					
					
					
					
					

**Table 4.3:** The left-most column lists the ten test compounds predicted most confidently by the Gaussian Process Classification (GPC) model utilizing the ISOAK molecular graph kernel. The remaining five columns contain the *explanations*, i.e. the five compounds from the training set that are most relevant for the respective prediction. As one would expect when using a graph kernel, the predictions are based on structurally very similar molecules in the training set and the resulting *explanations* may be perceived as convincing by human experts utilizing this model.

test mol.   <i>explanations</i> (most relevant molecules in training set)					
					
					
					
					
					
					
					
					
					
					
					

**Table 4.4:** The left-most column lists the ten test compounds predicted most confidently by the Gaussian Process Classification (GPC) model utilizing the radial basis function (RBF) kernel. The remaining five columns contain the *explanations*, i.e. the five compounds from the training set that are most relevant for the respective prediction. Despite the fact that structural information about the graph is only implicitly contained in the utilized vectorial descriptors, the predictions are based on structurally very similar molecules in the training set and the resulting *explanations* may be perceived as convincing by human experts utilizing this model.

training set. The resulting *explanations* may be perceived as convincing by human experts utilizing this model.

Lastly, visually inspecting the ten test compounds predicted most confidently by the Gaussian Process Classification (GPC) model utilizing the radial basis function (RBF) kernel, one finds that this model also identifies structurally very similar neighbors. This is somewhat surprising, because structural information about the molecules is only implicitly contained in the vectorial descriptors that enter the RBF kernel as inputs. Consequently, this type of model can also produce *explanations* that may be perceived as convincing by human experts.

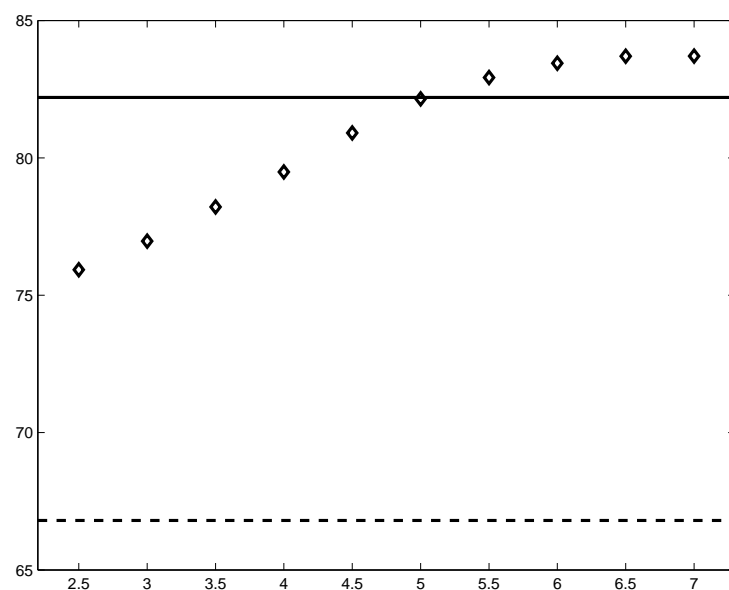
Further considering these initial *explanations*, one important question comes to mind: How local are these models? In case of KNN with  $k = 5$ , the five most relevant compounds together determine, by definition, 100 % of the prediction. When considering GPC RBF and the ten test compounds in Table 4.4, this percentage drops to between 6.0 and 9.3 %. Similarly, for GPC ISAOK (Table 4.3) the five most relevant compounds together determine only 4.1 to 6.3 % of the predictions for the ten test compounds.

### *Increasing Locality of GPC*

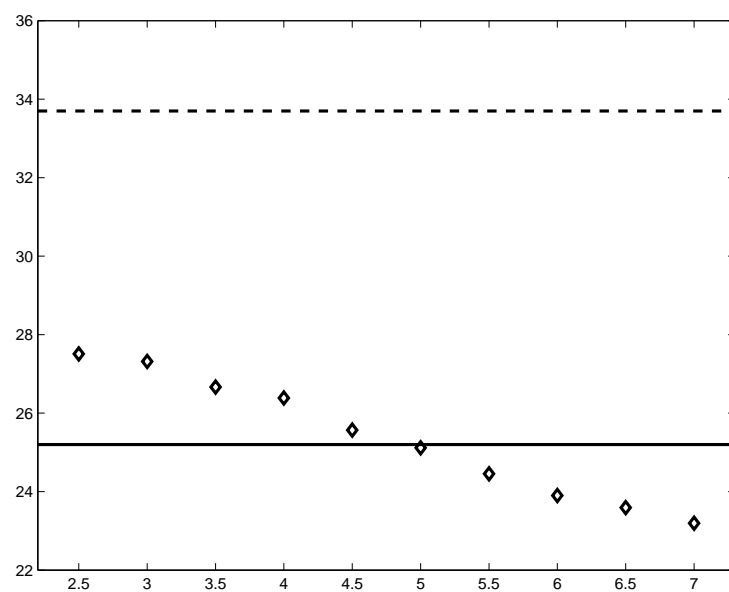
Having learned that the example predictions made by the GPC RBF and GPC ISOAK models are almost completely determined by other compounds than the five compounds most relevant to each respective prediction, the next questions are: Are the respective ten test compounds special or are the models simply not very local? And if they are not local, can we make these models sufficiently local for generating convincing explanations?

In case of the GPC RBF model, the answer is yes: The *kernel width* controls directly how local the model operates. Normally, this parameter is automatically set during model fitting [121]. On this data set, a kernel width of  $\approx 7$  is learned. The impact that manually reducing the kernel width has on the area under the receiver operating characteristic curve (AUC) and error rate achieved by GPC RBF is illustrated by diamonds in Figure 4.1. Results of GPC ISOAK and KNN with  $k = 5$  are indicated by solid and dashed horizontal lines, respectively. One can see that the AUC achieved by GPC RBF decreases as the kernel width is reduced. However, no sudden drop can be observed – AUC decreases in small steps from 84 % down to 76 %. The error rate behaves similarly: It increases in small steps from 23 % up to 28 %. The diamonds used for GPC RBF cross the solid horizontal line indicating the performance achieved using GPC ISOAK, i.e. GPC RBF can perform less good, equal to or better than GPC ISOAK, depending on the kernel





(a) area under curve vs. kernel width



(b) error rate vs. kernel width

**Figure 4.1:** The impact of manually reducing the kernel width of a GPC RBF model is illustrated by diamonds. Performance achieved using the GPC ISOAK model is indicated by solid lines and KNN ( $k = 5$ ) results are represented by dashed horizontal lines, respectively.

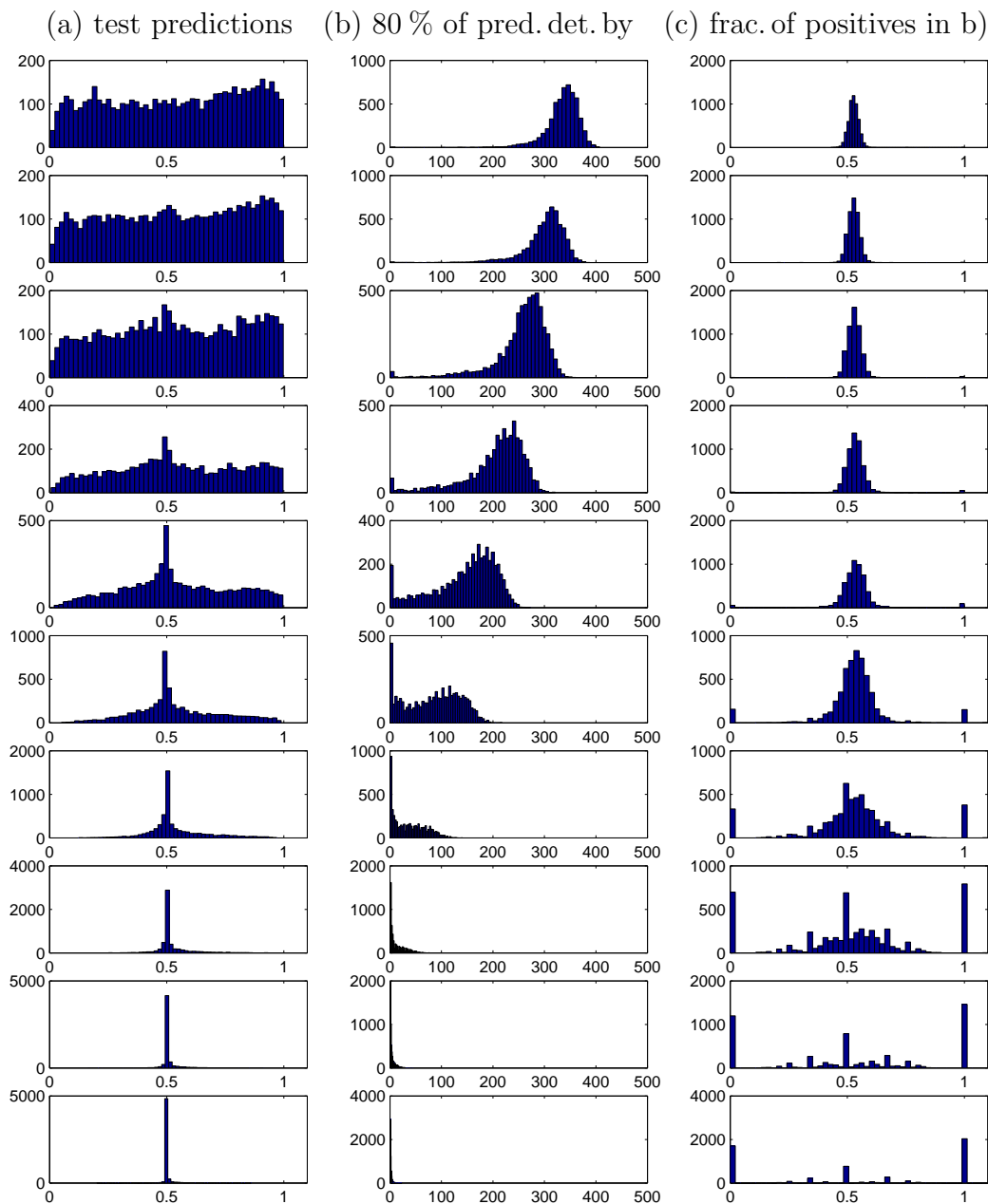
width used, and performs best when using the kernel width automatically determined in the model fitting process. Mind that for the range of kernel widths presented, GPC RBF always outperforms KNN ( $k = 5$ ) by a large margin.

Table 4.5 presents more detailed information about GPC RBF models trained using different kernel widths: The first *row* corresponds to the model with the learned kernel width. In each subsequent row, the kernel width is manually set 0.5 units smaller than in the previous row, thereby spanning the range  $[7 \dots 2.5]$ .

- Column (a) shows histograms of the predictions for the whole test set.
- Column (b) contains histograms indicating for each prediction on the test set, exactly how many training compounds are minimally needed to determine  $\geq 80\%$  of the respective prediction (measured by the  $\hat{\beta}_t$  coefficient introduced in Sec. 4.3, Eq. 4.10).
- Column (c) shows histograms indicating the fraction of positives (class 1 compounds) among the smallest possible set needed to determine 80 % of each test prediction.

In the first row, we can see that predictions span the whole range  $[0 \dots 1]$  (column (a)). For vast majority of test set predictions, 300 to 400 training set compounds are minimally needed to determine 80 % of the respective prediction as indicated by the  $\beta$  coefficient introduced in Sec. 4.3 (histograms in column (b)). Considering our goal of explaining predictions using five training compounds, this model is certainly not local enough. Column (c) presents histograms indicating the fraction of positives (class 1 compounds) among the smallest set needed to determine 80 % of each test prediction. For most predictions from the model with the learned kernel width, this fraction is in the range  $[0.48 \dots 0.58]$ . The large differences in the predictions that are relatively evenly distributed in the whole range  $[0 \dots 1]$  are therefore not the result of a simple KNN-like voting of many equally important training compounds. Instead, large differences in the  $\alpha$  coefficients learned by the model result in very different predictions for different test compounds.

When the kernel width used is manually reduced and kept fixed during the model fitting process, predictions span smaller and smaller ranges. At the smallest kernel width investigated, the range inside which most predictions can be found has shrunk to  $[0.495 \dots 0.505]$ . Keep in mind that Figure 4.1 indicates that, at the same time, performance measured by error rate and AUC decreases only slightly. Therefore, most predictions are still correctly placed below or above the class-separating threshold of 0.5 (as indicated by



**Table 4.5:** The first row corresponds to the GPC RBF model with the learned kernel width ( $\approx 7$ ). In each subsequent row, the kernel width is manually set 0.5 units smaller than in the previous row, thereby covering the range  $[7 \dots 2.5]$ . Column (a) shows histograms of the predictions for the whole test set. Column (b) contains histograms indicating for each prediction on the test set, exactly how many training compounds are minimally needed to determine  $\geq 80\%$  of the respective prediction and column (c) shows histograms indicating the fraction of positives (class 1 compounds) among the smallest possible set needed to determine 80 % of each test prediction.

the still rather low error rate). Also, the numerical value of the predictions still is somehow related to the confidence in each prediction (as indicated by the still relatively high AUC). Predictions do, however, lose the original meaning of being the probability of the respective compound belonging to the positive class (which is exhibited by the model with the learned kernel width presented in the topmost row).

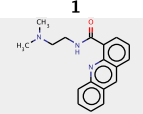
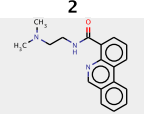
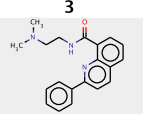
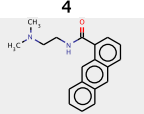
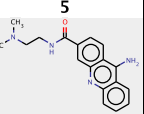
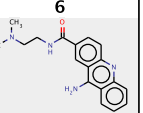
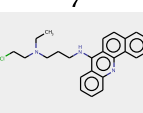
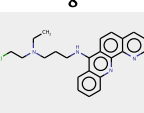
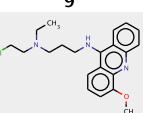
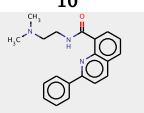
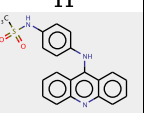
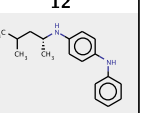
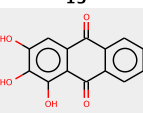
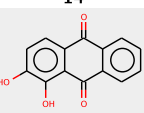
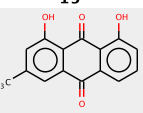
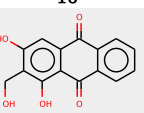
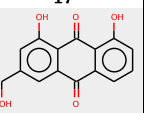
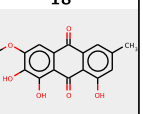
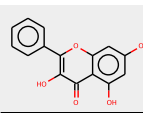
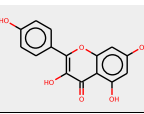
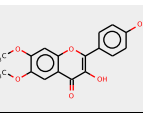
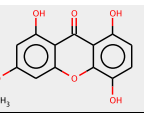
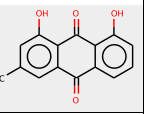
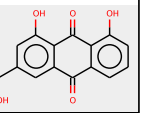
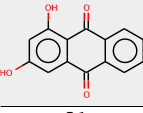
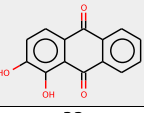
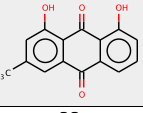
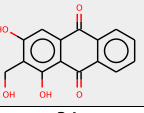
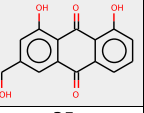
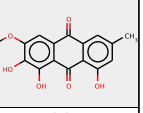
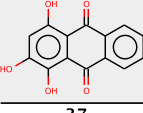
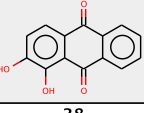
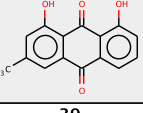
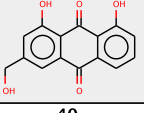
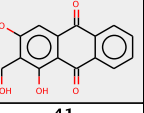
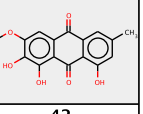
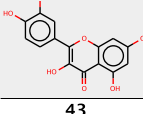
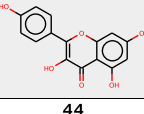
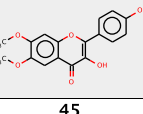
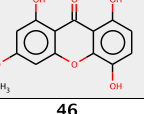
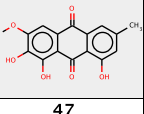
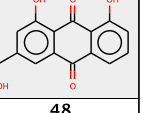
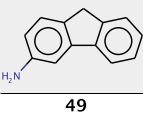
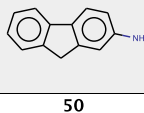
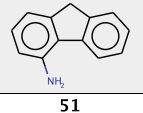
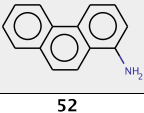
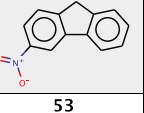
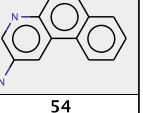
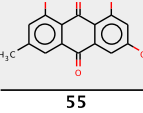
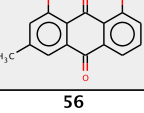
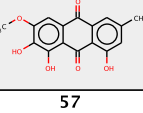
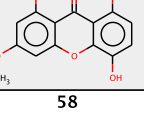
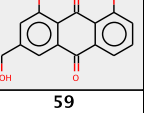
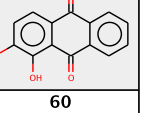
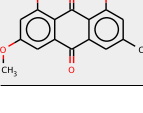
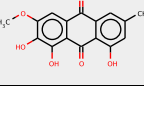
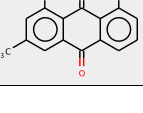
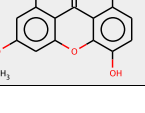
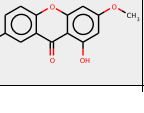
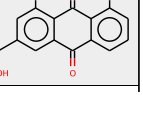
As column (b) indicates, the number of compounds needed to determine  $\geq 80\%$  of each prediction decreases as the kernel width is decreased. For kernel widths  $\leq 3.5$ , almost all test compounds can be predicted using at most five training compounds. Remarkably, even these very local GPC RBF models still outperform KNN (with  $k = 5$ ) by a large margin (as previously established in Figure 4.1).

Finally, column (c) shows that the fraction of positives (class 1 compounds) among the smallest possible set needed to determine 80 % of each test prediction starts out around 0.5, but when the kernel width is reduced, it moves into the extremes (mostly either 0 or 1). This is necessarily so, because, as observed in the last paragraph on column (b), when using small kernel widths, predictions only depend on a very small number of compounds and, consequently, the probability of observing a consistent neighborhood increases.

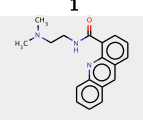
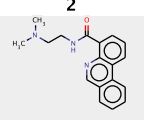
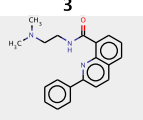
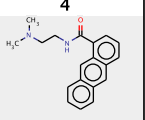
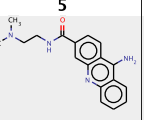
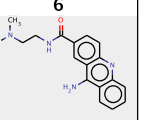
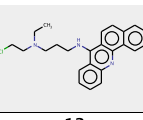
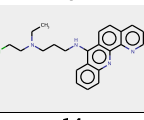
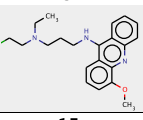
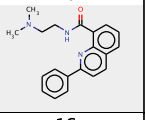
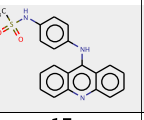
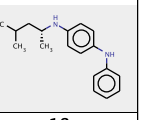
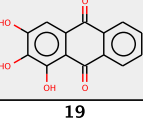
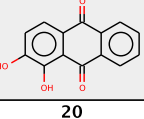
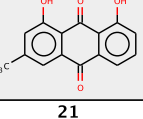
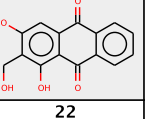
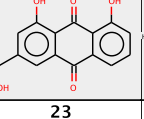
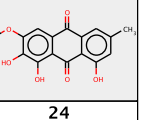
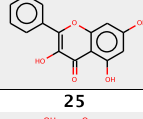
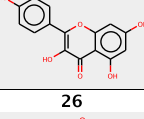
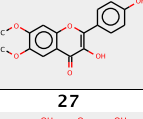
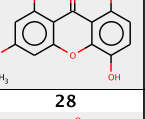
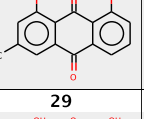
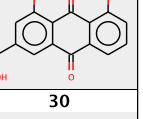
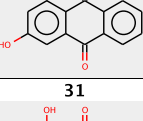
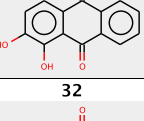
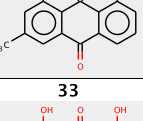
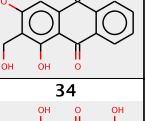
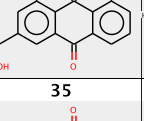
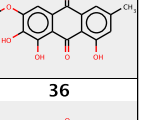
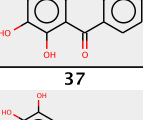
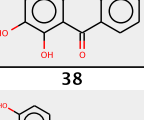
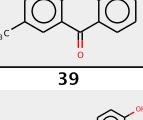
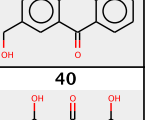
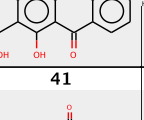
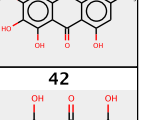
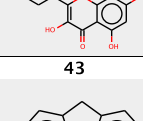
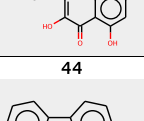
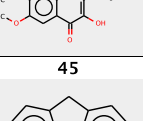
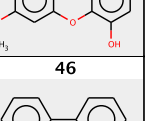
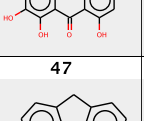
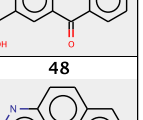
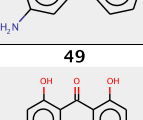
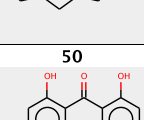
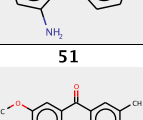
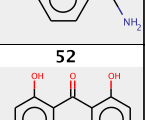
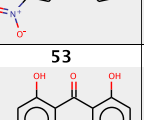
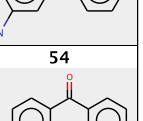
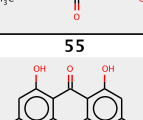
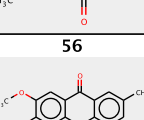
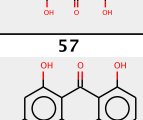
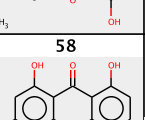
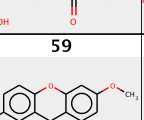
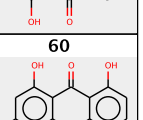
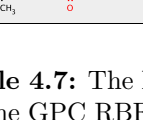
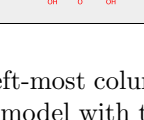
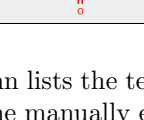
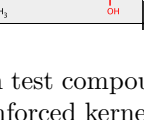
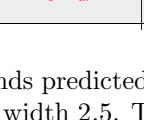
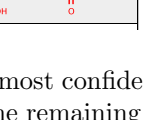
#### *Quality of Explanations Based on Local GPC RBF Models*

As established in Sec. 4.3.4, reducing the kernel width of GPC RBF indeed results in test set predictions being almost completely determined by five or less compounds in the training set. Since the reduced kernel width is imposed throughout the learning process, the weight coefficients  $\alpha$  that are learned by the GPC algorithm can turn out differently when the kernel width is changed. The previously discussed Table 4.4 shows the *explanations* (most relevant training compounds) for the compounds predicted most confidently by the GPC RBF model with the learned kernel width  $\approx 7$ . Table 4.6 presents, for the same ten test compounds as seen in Table 4.4, the *explanations* generated by the GPC RBF model with the smallest kernel width investigated (2.5). Again, the *explanations* are found to be visually convincing and, as expected (Sec. 4.3.4), they almost completely determine the predictions obtained: Percentages are in the range  $[81 \dots 100]\%$  and are 95 % on average. As these are compounds predicted most confidently by a previously introduced model with a larger kernel width, one may ask: How about the predictions made most confidently by the new much more local model? Table 4.7 presents the test compounds predicted most confidently

| test mol. | *explanations* (most relevant molecules in training set) |

**Table 4.6:** For the same ten test compounds as seen in Table 4.4, this table shows the *explanations* generated by the GPC RBF model with the smallest kernel width investigated (2.5). Again, the *explanations* are found to be visually convincing and (due to the reduced kernel width) they almost completely determine the predictions obtained.

test mol.   <i>explanations</i> (most relevant molecules in training set)					
					
					
					
					
					
					
					
					
					
					

**Table 4.7:** The left-most column lists the ten test compounds predicted most confidently by the GPC RBF model with the manually enforced kernel width 2.5. The remaining five columns contain the *explanations*, i.e. the five compounds from the training set that are most relevant for the respective prediction. They are structurally very similar molecules predicted and may be perceived as convincing by human experts utilizing this model.

by the very local GPC RBF model. Again, the *explanations* are found to be convincing. Predictions are almost completely determined by the depicted training compounds ([99...100] %, with the average at 99.7 %).

#### 4.3.5 Conclusion

Models can support human experts (e.g. in the *lead optimization* phase of the drug discovery process) in making decisions by providing accurate predictions of the relevant properties of the compounds under consideration. However, human experts are not likely to trust a model if its prediction deviates from their own intuition. In the previous sections, a method for generating *explanations* of predictions was developed. By consulting these *explanations*, the user may learn that predictions that have previously been perceived as surprising are in fact made by generalizing in an acceptable way from a small<sup>6</sup> set of structurally similar training compounds that he did not know before.

The new patent pending [2] method for generating *explanations* is based on the introduced  $\beta$  coefficient. More specifically,  $\hat{\beta}_t$  is defined as the vector of normalized contribution of each training point to the prediction for a specific test point  $x_t$  and can be calculated analytically for kernel methods such as support vector machines for classification and regression, Gaussian Process regression and classification and kernel ridge regression. *Explanations* generated for Gaussian Processes for Classification (GPC) models of Ames toxicity have been investigated. As a baseline, the k-Nearest-Neighbor algorithm (KNN) was investigated (when using KNN with sufficiently small  $k$ , the nearest neighbors can serve as *explanations*).

The KNN model turned out to present training compounds that are structurally quite different from the respective test compounds. The GPC model utilizing the ISOAK molecular graph kernel succeeded in identifying training compounds that are structurally similar to the respective test compounds. This was expected, since the kernel explicitly uses the molecular structure. Unfortunately, this model is not sufficiently local, i.e. each prediction is based on hundreds of training compounds. GPC RBF models are capable of identifying training compounds that are structurally similar to the respective test compounds. This is a somewhat surprising result, because the vectorial descriptors serving as input to the RBF kernel contain the molecular graph only implicitly. Furthermore, these models can be made to operate sufficiently local. The most local GPC RBF models investigated rely on less than

---

<sup>6</sup>This set has to be small, because (as learned from research in psychology) humans can only consider  $7 \pm 2$  items simultaneously.

five training compounds for making each test prediction and still outperform KNN ( $k = 5$ ) by a large margin.

Interesting directions for future research include investigating *explanations* for other classifiers and regression models. Furthermore, evaluating a large number of *explanations* with an (ideally also large) group of experts is considered the best possible way of *quantifying* how useful such *explanations* can be in practice.

**Note added between grading and publication of Timon Schroeter’s dissertation:** In the initial version of this thesis, the author has suggested evaluating a large number of *explanations* with an (ideally also large) group of experts. David Baehrens (who was initially supervised by the author of this thesis and later by Katja Hansen and Mikio Braun) has conducted this evaluation and has published the results in his diploma thesis [124].

In a forced choice paradigm, human users were presented with conflicting predictions by two models trained on different subsets of the Ames mutagenicity benchmark data set that we published earlier [5]. Forty one (41) students studying pharmaceutical sciences, medicine, chemistry and other subjects each answered forty (40) questions. The effect of the students subject and the effect of *explanations* were quantified in terms of conditional odds ratios. Students studying pharmaceutical sciences were found to choose correct predictions with a two-fold increased probability (relative to the random choice baseline). For students of any subject, *explanations* were shown to result in a 93 % increased probability of choosing correct predictions. The latter result is also statistically significant at a p-value of 0.01.

In conclusion, a large number of *explanations* has been evaluated using a large group of experts. *Explanations* led to a statistically significant ( $p=0.01$ ) strong increase (93 %) in probability of choosing correct predictions, thereby underscoring that human users can strongly profit from *explanations*.



#### 4.4 GUIDING COMPOUND OPTIMIZATION & EXPLAINING DECISIONS USING LOCAL GRADIENTS

In this thesis, two separate methodologies for explaining individual predictions of (possibly non-linear) machine learning models are presented. The method presented in Sec. 4.3 *explains* predictions by the means of visualizing relevant objects from the training set of the model. This allows human experts to understand how each prediction comes about. If a prediction conflicts with his intuition, the human expert can easily find out whether the grounds for the models predictions are solid or if trusting his own intuition is the better idea [2].

The method presented in this section utilizes local gradients of the models predictions to explain predictions in terms of the locally most relevant features. This not only teaches the human expert which features are relevant for each individual prediction, but also gives a directional information. Abstractly speaking, one can learn in which direction a data point has to be moved to increase the prediction for the target value [1]. In the context of lead optimization, this means that the human expert can obtain a type of *guidance in compound optimization*. For this reason, the two explaining-related methodologies are presented in separate sections.

##### 4.4.1 Introduction

Automatic non-linear classification is a common and powerful tool in data analysis. Machine learning research has created methods that are practically useful and that can classify unseen data after being trained on a limited training set of labeled examples. Nevertheless, most of the algorithms do not *explain* their decision. However, in many application scenarios of data analysis it is essential to obtain an instance based explanation, i.e. one would like to gain an understanding what input features made the non-linear machine give its answer for each individual test object and how one could modify this object to optimize its properties.

In the *lead optimization* phase of the drug discovery process (as introduced in Sec. 2.1), variants of the respective lead compounds are synthesized with the goal of finally producing a drug candidate that fulfills all requirements. Reaching this goal typically takes thousands of experiments, each of which is quite expensive. The many decisions which compound or small batch of compounds to synthesize and test next are made by humans, and they are often made on the basis of very little information. A type of *guidance in compound optimization* that facilitates choosing the most promising compounds in each iteration can therefore be of great value.

Typically, explanations are provided jointly for all instances of the training set, for example feature selection methods find out which inputs are salient for a good generalization [125]. While this can give a coarse impression about the global usefulness of each input dimension, it is still an ensemble view and does not provide an answer on an instance basis. In the neural network literature also solely an ensemble view was taken in algorithms like input pruning [126]. The only classification which does provide individual explanations are decision trees [102].

This section proposes a simple framework that provides local explanation vectors applicable to *any* classification method in order to help understanding prediction results for single data instances. The local explanation yields the features being relevant for the prediction at the very points of interest in the data space and is able to spot local peculiarities which are neglected in the global view e.g. due to cancellation effects.

The section is organized as follows: Local explanation vectors are technically *defined* as class probability gradients and an *illustration* for Gaussian Process Classification (GPC) is given. In the following subsection, the new approach is contrasted with *related work*. Some methods output a prediction without a direct probability interpretation. For these, our recent publication [1] proposes a way to *estimate* local explanations.

Results obtained using this new methodology are presented in Sec. 5 and in [1]. More specifically, in section 4 in [1] it is applied to learn distinguishing properties of Iris flowers by estimating explanation vectors for a k-NN classifier applied to the classic Iris dataset. Section 5 in [1] discusses how the approach applied to a SVM classifier allows to explain how digits "two" are distinguished from digits "8" in the USPS dataset. In section 5.8.1 a challenging real world application scenario is presented in which the proposed explanation capabilities prove useful: In the *lead optimization* phase of the drug discovery process, human experts regularly decide how to modify existing lead compounds in order to obtain new compounds with improved properties. Models capable of explaining predictions can help in the process of choosing promising modifications. The automatically generated explanations match with chemical domain knowledge about toxifying functional groups of the compounds in question.

The presentation of results is concluded in the last paragraph of Sec. 4.3. Our recent publication [1] discusses further characteristic properties and limitations of the new methodology. Future directions are two-fold: First it is believed that the new method will find its way into the tool boxes of practitioners who not only want to automatically classify their data but who also would like to understand the learned classifier. The second direction is to generalize this approach to other prediction problems such as regression.

#### 4.4.2 Definitions

This Subsection gives definitions of local explanation vectors in the classification setting. It starts with a theoretical definition for multi-class Bayes classification and then gives a specialized definition being more practical for the binary case.

For the multi-class case, suppose one is given data points  $x_1, \dots, x_n \in \mathbb{R}^d$  with labels  $y_1, \dots, y_n \in \{1, \dots, C\}$  and one intends to learn a function that predicts the labels of unlabeled data points. Assuming that the data could be modelled as being IID-sampled from some unknown joint distribution  $P(X, Y)$ , in theory, one can define the Bayes classifier,

$$g^*(x) = \arg \min_{c \in \{1, \dots, C\}} P(Y \neq c \mid X = x)$$

which is optimal for the 0-1 loss function [see 127].

For the Bayes classifier one defines the *explanation vector* of a data point  $x_0$  to be the derivative with respect to  $x$  at  $x = x_0$  of the conditional probability of  $Y \neq g^*(x_0)$  given  $X = x$ , or formally,

**Definition 4.1.**

$$\zeta(x_0) := \left. \frac{\partial}{\partial x} P(Y \neq g^*(x_0) \mid X = x) \right|_{x=x_0}$$

Note that  $\zeta(x_0)$  is a  $d$ -dimensional vector just like  $x_0$  is. The classifier  $g^*$  partitions the data space  $\mathbb{R}^d$  into up to  $C$  parts on which  $g^*$  is constant. Under the usual assumption that  $P(X = x \mid Y = c)$  is for all  $c$  smooth in  $x$ ,  $\zeta(x_0)$  defines on each of those parts a vector field that characterizes the flow away from the corresponding class. Thus entries in  $\zeta(x_0)$  with large absolute values highlight features that will influence the class label decision of  $x_0$ . A positive sign of such an entry implies that increasing that feature would lower the probability that  $x_0$  is assigned to  $g^*(x_0)$ . Ignoring the orientations of the explanation vectors,  $\zeta$  forms a continuously changing (orientation-less) vector field along which the class labels change. This vector field lets one *locally* understand the Bayes classifier.

For the case of binary classification we directly define local explanation vectors as local gradients of the probability function  $p(x) = P(Y = 1 \mid X = x)$  of the learned model for the positive class.

So for a probability function  $p : \mathbb{R}^d \rightarrow [0, 1]$  of a classification model learned from examples  $\{(x_1, y_1), \dots, (x_n, y_n)\} \in \mathbb{R}^d \times \{-1, +1\}$  the explanation vector for a classified test point  $x_0$  is the local gradient of  $p$  at  $x_0$ :

**Definition 4.2.**

$$\eta_p(x_0) := \nabla p(x)|_{x=x_0}$$

By this definition the explanation  $\eta$  is again a  $d$ -dimensional vector just like the test point  $x_0$  is. Its individual entries point in the direction the prediction would take when the corresponding feature of  $x_0$  is increased locally and their absolute values give the amount of influence in the change in prediction. As a vector  $\eta$  gives the direction of the steepest ascent from the test point to higher probabilities for the positive class. For binary classification the negative version  $-\eta_p(x_0)$  indicates the changes in features needed to increase the probability for the negative class which may be especially useful for  $x_0$  predicted in the positive class.

In the following, definition 4.2 is applied to model predictions learned by Gaussian Process Classification (GPC), see Sec. 3.3. GPC is used here for three reasons. First is the state-of-the-art performance of Gaussian Processes for real world data sets, including challenging chemoinformatics data, see e.g. [7, 8, 11, 12, 90, 98, 121]. It is natural to expect a model with high prediction accuracy on a complex problem to capture relevant structure of the data which is worth explaining and may give domain specific insights in addition to the values predicted. For an evaluation of the explaining capabilities of our approach on a complex problem from chemoinformatics see section 5.8.1. Second GPC does model the class probability function used in Definition 4.2 directly.<sup>7</sup> And third it is possible to calculate the local gradients of the probability function analytically for differentiable kernels as follows.

Let  $f(x) = \sum_{i=1}^n \alpha_i k(x, x_i)$  be a GP model trained on sample points  $x_1, \dots, x_n \in \mathbb{R}^d$  where  $k$  is a kernel function and  $\alpha_i$  are the learned weights of each sample point. For a test point  $x_0 \in \mathbb{R}^d$  let  $\text{var}_f(x_0)$  be the variance of  $f(x_0)$  under the GP posterior for  $f$ . The probability for being of the positive class  $p(x_0)$  predicted by GPC can be shown to be (see Equation 6 in [6])

$$p(x_0) = \frac{1}{2} \text{erfc} \left( \frac{-f(x_0)}{\sqrt{2} * \sqrt{1 + \text{var}_f(x_0)}} \right),$$

with  $\text{erfc}$  being the complementary error function.

Then the local gradient of  $p(x_0)$  is given by

---

<sup>7</sup>For other classification methods such as Support Vector Machines which do not provide a probability function as its output, our recent publication [1] proposes a way to *estimate* local explanations.

$$\begin{aligned}
\nabla p(x)|_{x=x_0} &= \nabla \frac{1}{2} \operatorname{erfc} \left( \frac{-f(x)}{\sqrt{2} * \sqrt{1 + \operatorname{var}_f(x)}} \right) \Big|_{x=x_0} \\
&= \nabla \frac{1}{2} \left( 1 - \operatorname{erf} \left( \frac{-f(x)}{\sqrt{2} * \sqrt{1 + \operatorname{var}_f(x)}} \right) \right) \Big|_{x=x_0} \\
&= -\frac{1}{2} \nabla \operatorname{erf} \left( \frac{-f(x)}{\sqrt{2} * \sqrt{1 + \operatorname{var}_f(x)}} \right) \Big|_{x=x_0} \\
&= -\frac{1}{2} \frac{2}{\sqrt{\pi}} e^{-\left( \frac{-f(x_0)}{\sqrt{2} * \sqrt{1 + \operatorname{var}_f(x_0)}} \right)^2} \nabla \left( \frac{-f(x)}{\sqrt{2} * \sqrt{1 + \operatorname{var}_f(x)}} \right) \Big|_{x=x_0} \\
&= -\frac{1}{\sqrt{\pi}} e^{-\left( \frac{-f(x_0)}{\sqrt{2} * \sqrt{1 + \operatorname{var}_f(x_0)}} \right)^2} \left( -\frac{1}{\sqrt{2}} \nabla \left( \frac{f(x)}{\sqrt{1 + \operatorname{var}_f(x)}} \right) \Big|_{x=x_0} \right) \\
&= \frac{1}{\sqrt{2\pi}} e^{-\left( \frac{-f(x_0)}{\sqrt{2} * \sqrt{1 + \operatorname{var}_f(x_0)}} \right)^2} \left( \frac{\nabla f(x)|_{x=x_0}}{\sqrt{1 + \operatorname{var}_f(x_0)}} \right. \\
&\quad \left. + f(x_0) \left( \nabla \operatorname{var}_f(x)|_{x=x_0} * -\frac{1}{2} (1 + \operatorname{var}_f(x_0))^{-\frac{3}{2}} \right) \right) \\
&= \frac{1}{\sqrt{2\pi}} e^{-\left( \frac{-f(x_0)}{\sqrt{2} * \sqrt{1 + \operatorname{var}_f(x_0)}} \right)^2} \left( \frac{\nabla f(x)|_{x=x_0}}{\sqrt{1 + \operatorname{var}_f(x_0)}} \right. \\
&\quad \left. - \frac{1}{2} \frac{f(x_0)}{(1 + \operatorname{var}_f(x_0))^{\frac{3}{2}}} \nabla \operatorname{var}_f(x)|_{x=x_0} \right).
\end{aligned}$$

As a kernel function choose e.g. the RBF-kernel  $k(x_0, x_1) = e^{-(x_0 - x_1)^2 w}$ , which has the derivative  $\frac{\partial}{\partial x_{0,j}} k(x_0, x_1) = -2we^{-(x_0 - x_1)^2 w}(x_{0,j} - x_{1,j})$  for  $j \in \{1, \dots, d\}$ . Then the elements of the local gradient  $\nabla f(x)|_{x=x_0}$  are

$$\frac{\partial f}{\partial x_{0,j}} = -2w \sum_{i=1}^n \alpha_i e^{-(x_0 - x_i)^2 w} (x_{0,j} - x_{i,j}),$$

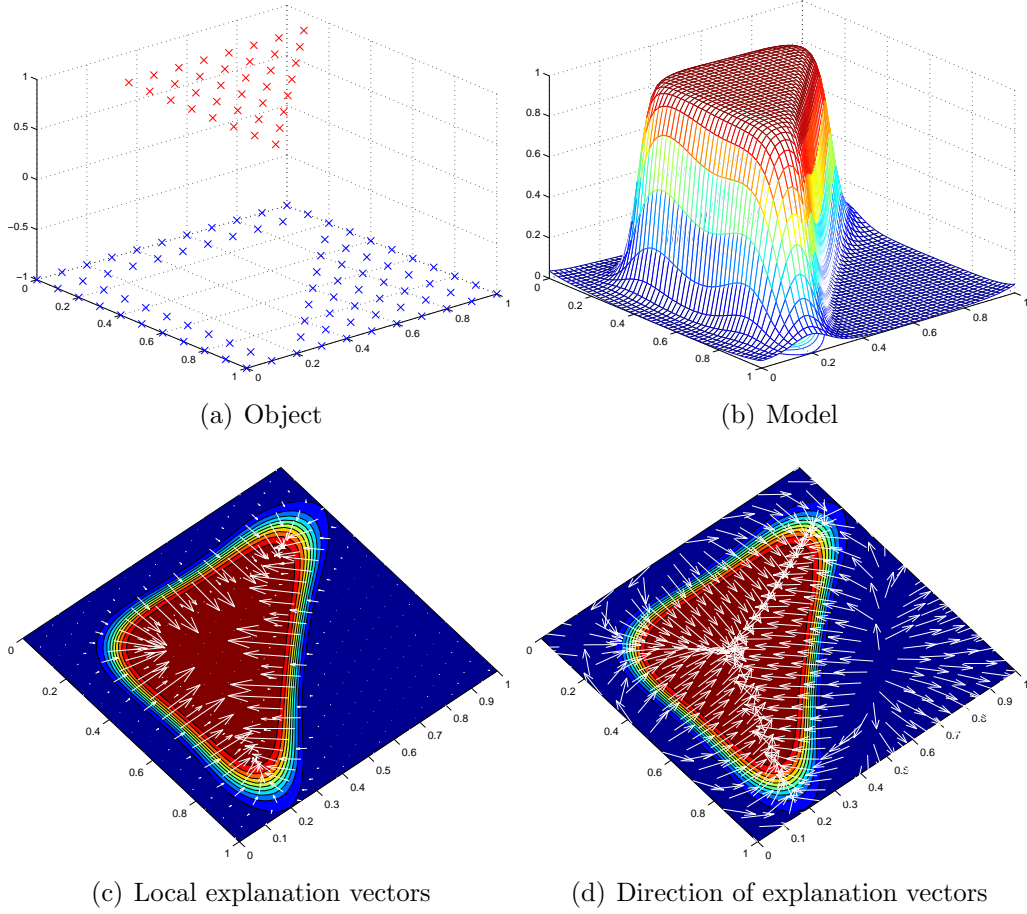
for  $j \in \{1, \dots, d\}$ .

For  $\operatorname{var}_f(x_0) = k(x_0, x_0) - k_*^T (K + \Sigma)^{-1} k_*$  the derivative is given by

$$\begin{aligned}
\nabla \operatorname{var}_f(x)|_{x=x_0} &= \frac{\partial \operatorname{var}_f}{\partial x_{0,j}} \\
&= \frac{\partial}{\partial x_{0,j}} k(x_0, x_0) - 2 * k_*^T (K + \Sigma)^{-1} \frac{\partial}{\partial x_{0,j}} k_*
\end{aligned}$$

for  $j \in \{1, \dots, d\}$ .

Figure 4.2 shows the training data of a simple object classification task (a) and the model learned using GPC (b). The data in (a) is labeled  $-1$  for the blue points and  $+1$  for the red points. As illustrated in (b) the model is a probability function for the positive class which gives every data point



**Figure 4.2:** Explaining simple object classification with Gaussian Processes.

a probability of being in this class. In (c) the probability gradient of the model is shown together with the local gradient explanation vectors. On 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 singles out. 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. As indicated by (d), explanations close to the edges of the plot (especially in the right hand side corner) point away from the positive class. However, as we can learn from (c), their magnitude is very small, and following explanation vectors with large magnitude indeed leads to the positive class.

#### 4.4.3 Related Work

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

The explanation vector proposed here is similar in spirit to sensitivity analysis which is common to various areas of information science. A classical example is the outlier sensitivity in statistics [128]. In this case, the effects of removing single data points on estimated parameters are evaluated by an influence function. If the influence for a data point is significantly large, it is detected as an outlier and should be removed for the following analysis. In regression problems, leverage analysis is a procedure along similar lines. It detects leverage points which have potential to give large impact on the estimate of the regression function. In contrast to the influential points (outliers), removing a leverage sample may not actually change the regressor, 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. Our framework of explanation vectors considers a different view. It describes the influence of *moving* single data points locally and it thus answers the question which directions are locally most influential to the prediction. The explanation vectors are used for extracting sensitive features which are relevant to the prediction results, rather than detecting/eliminating the influential samples.

In recent decades, explanation of results by expert systems have been an important topic in the AI community. Especially, for those 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 [129]. There the influence is evaluated of removing a set of variables (features) from evidences and the explanation is constructed from those variables which affect inference (relevant variables). For example, [130] measures the cost of omitting a single feature  $E_i$  by the cross-entropy

$$H^-(E_i) = H(p(D|E); P(D|E \setminus E_i)) = \sum P(d_j|E) \log \frac{P(d_j|E)}{p(d_j|E \setminus E_i)},$$

where  $E$  denotes evidences and  $D$  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 evidences  $E$ , and is thus local.

Similarly [131] and [132] try to explain the decision of trained kNN-, SVM- and ANN-models 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 [131] 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, [132] propose to calculate the differences between model predictions for every choice of feature subset. The principal differences between our approach and these frameworks are: (i) We consider continuous features and no structure among them is required, while the 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, i.e. any weighted combination of variables, while other approaches only consider the omission of a set of variables.



## CHAPTER 5

# RESULTS

### 5.1 OVERVIEW

This chapter presents the results of six studies on constructing models for various *ADME/Tox*<sup>1</sup> properties, results obtained using a new algorithm for explaining predictions & *eliciting hints for compound optimization* and the results of a *virtual screening* study. The results of each individual work have been previously published. The respective journal publications are referenced in Table 1.1 in Sec. 1.2 and inside each individual section. More specifically, the ADME/Tox properties investigated in this chapter are:

- Partition Coefficients (Sec. 5.2)
- Aqueous Solubility (Sec. 5.3)
- Cytochrome P450 Inhibition (Sec. 5.4)
- Metabolic Stability (Sec. 5.5)
- Ames Mutagenicity (Sec. 5.6)
- hERG Channel Blockade Effect (Sec. 5.7)

Local gradients for explaining individual classification decisions and eliciting compound optimization are validated in Sec. 5.8, using two benchmark sets of data that are well known in the machine learning community (IRIS flowers and USPS digits) and was then applied to Gaussian Process Classification models for Ames mutagenicity.

Furthermore, the author of this thesis participated in a *virtual screening* study that led to the discovery of new PPAR $\gamma$  agonists (Sec. 5.9). Both retrospective and prospective results are discussed.

---

<sup>1</sup>The acronym ADME/Tox stands for properties relating to Absorption, Digestion, Metabolism, Excretion & Toxicity of chemical compounds.

## 5.2 PARTITION COEFFICIENTS

Lipophilicity of drugs is a major factor in both pharmacokinetics and pharmacodynamics. Since a large fraction of drug failures ( $\sim 50\%$ ) [133] results from an unfavorable PC-ADME/T profile (physicochemistry, absorption, distribution, metabolism, excretion, toxicity), predicted octanol water partition coefficients  $\log P$  and  $\log D$  are nowadays considered early on in lead discovery. Due to the confidentiality of in-house data, makers of predictive tools are usually not able to incorporate such data from pharmaceutical companies. Commercial predictive tools are therefore typically constructed using publicly available measurements of relatively small and mostly neutral molecules. Often, their accuracy on the in-house compounds of pharmaceutical companies is relatively low [115].

The following section describes how predictive models for lipophilicity were constructed in collaboration with researchers at Bayer Schering Pharma. The usefulness of individual confidence estimates produced using different algorithms was evaluated; the discussion in this section focuses on Gaussian Process models. A more detailed description of the results of modeling  $\log D_7$  using different machine learning methods and both in-house and public data sets can be found in [7, 12].

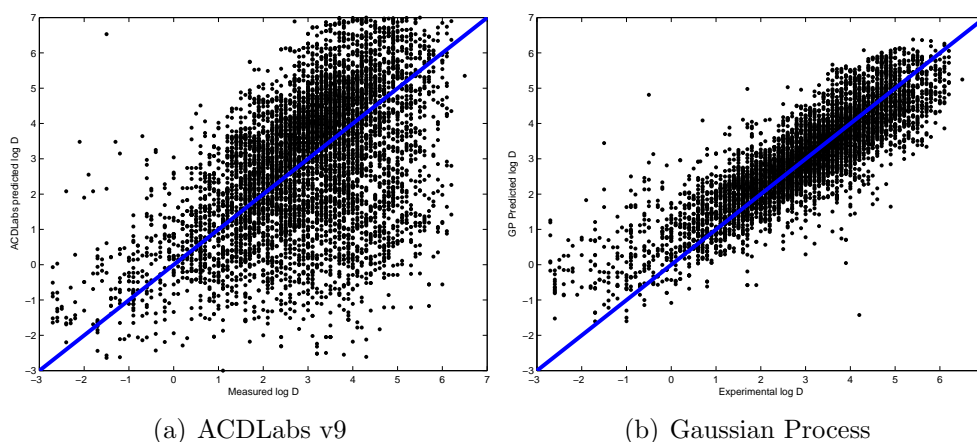
From a machine learning perspective, the modeling task can be summarized as follows:

- supervised regression task
- training data: 14556 compounds, blind<sup>2</sup> test data: 7013 compounds
- data representation: 1664 dimensional vectors [123]
- learning algorithms investigated: Gaussian Processes, support vector machines, random forests, ridge regression

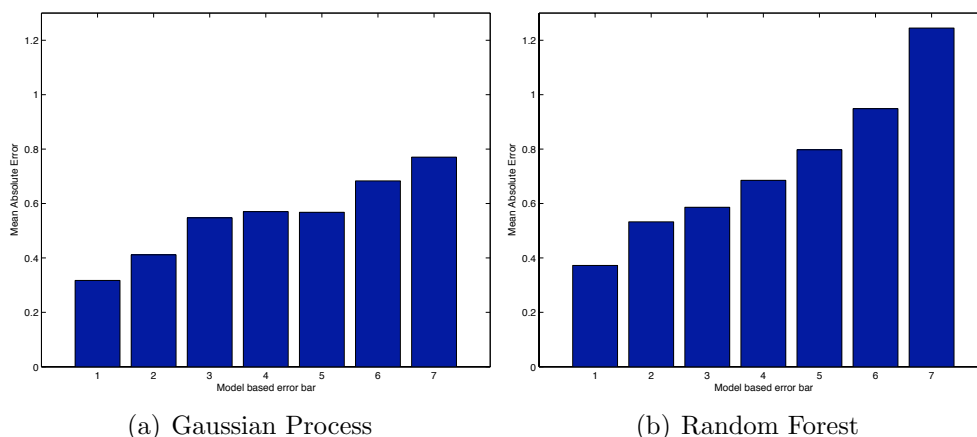
Figure 5.1 illustrates the prediction performance achieved by our Gaussian Process model, compared to the commercial tool ACDLabs v9. Both models were applied to the same set of 7013 drug discovery compounds in a blind test scenario, as described in Sec. 3.4. We chose the mean absolute error, root mean squared error and the percentage of compounds predicted correctly within one log unit as numerical performance indicators. ACDLabs v9 performs as follows: 1.40, 1.79 and 44.2 %. Confirming the impression

---

<sup>2</sup>Initially, performance was estimated in cross-validation on the training data. Later, the final model was evaluated by a group of researchers who were not involved in model building, using a set of new measurements that had become available in the meantime. Hence the term “blind test”, see also Sec. 3.4.



**Figure 5.1:** Scatter-plots from predicting  $\log D_7$  using the commercial tool ACDLabs v. 9.0 [75] and Gaussian Process models for a blind test set of 7013 drug discovery compounds.



**Figure 5.2:** Exploiting model based error bars, the mean absolute error can be reduced significantly. Predictions of  $\log D_7$  for blind test set (see Sec. 5.2) are binned by model based confidence estimates such that each of the seven bins contains 1000 compounds.

from Figure 5.1, our own final Gaussian Process Model produces much better results, namely 0.60, 0.82 and 81.2 %. These results support the promising impression gained from the visual pre-analysis described in Sec. 3.2.

How useful are the individual confidence estimates produced by the Gaussian Process model? One possible way of using these confidence estimates is ranking predictions by their assigned confidence and then checking whether the model actually does make smaller prediction errors when it predicts confidently or, vice versa, large prediction errors where it's not confident. If so, we could either focus on the most confident predictions or discard just the

most inconfidently made predictions.

Kendall's  $\tau$  is a measure of ranking quality [105] and is 1 for 100 % correctly ordered lists and 0 for random lists. For the blind test set, we find the following  $\tau$ s, when considering individual compounds, bins of five and bins of twenty compounds: 0.29, 0.54, 0.72. By definition, the confidence estimates (error bars) produced by Gaussian Processes only have a meaning in a statistical sense, i.e. 68 % of all predictions are within one standard deviation. In this case, however, they carry a lot of information even when considering individual compounds. When first sorting by the confidence estimate and then binning neighboring predictions and averaging both confidence estimates and actual prediction errors observed, we find that  $\tau$  increases significantly.

What do these  $\tau$ s mean for our prediction performance indicators used in the second last paragraph? We order our predictions by the confidence estimates produced by the Gaussian Process model, and then just take the top of the list, where the confidence estimate (error bar) is below 0.3. This part of the list still contains predictions for 2603 compounds and our performance indicators come out at 0.40, 0.55 and 91.3 %. So by focussing on the more confident predictions, we decrease the mean absolute error by one third to just 0.4 log units and we now predict more than 90 % of all log D<sub>7</sub> correct within one log unit. Figure 5.2 illustrates which mean absolute errors can be achieved, not just when focussing on the top of the list, but also by just rejecting the most inconfidently made predictions etc.

### 5.3 AQUEOUS SOLUBILITY

Aqueous solubility is of paramount importance to many areas of chemical research, such as medicinal chemistry, pharmacokinetics, formulation, agrochemistry [134] and environmental applications. In the drug design process, 50% of the failures[133] are due to an unfavorable, PC-ADME/T profile (physicochemistry, absorption, distribution, metabolism, excretion, toxicity), often resulting from poor aqueous solubility. There exists a connection between lipophilicity (Sec. 5.2) and aqueous solubility: The general trend is: The more lipophilic the compounds, the less soluble they are. However, solubility is more complicated than that: It also depends on the stability of the crystals of the compound. Small changes in a compounds structure can have a big impact on the stability of its crystals, consequently making solubility very difficult to predict. A lot of research has thus been devoted to developing in-silico models for aqueous solubility [116, 135–152]. The aqueous solubility of electrolytes at a specific pH is especially hard to predict [151, 152], but many drugs are electrolytes.

The following section describes how predictive models for aqueous solubility were constructed in collaboration with researchers at Bayer Schering Pharma. It discusses how it was noticed that the training and blind test data (described below) exhibit the “covariate shift” phenomenon (Sec. A.1.2) and illustrates how one can use individual confidence estimates to obtain reliable predictions for the compounds included in the respective models domain of applicability (Sec. 3.7). This section focuses on Gaussian Process models and the two in-house sets of data. A more detailed description of the results obtained using different machine learning methods and both in-house and public data sets can be found in [7, 12]. This sections last paragraph treats the a posteriori explanation of predictions that are outliers with respect to the confidence estimates predicted by the respective Gaussian Process model.

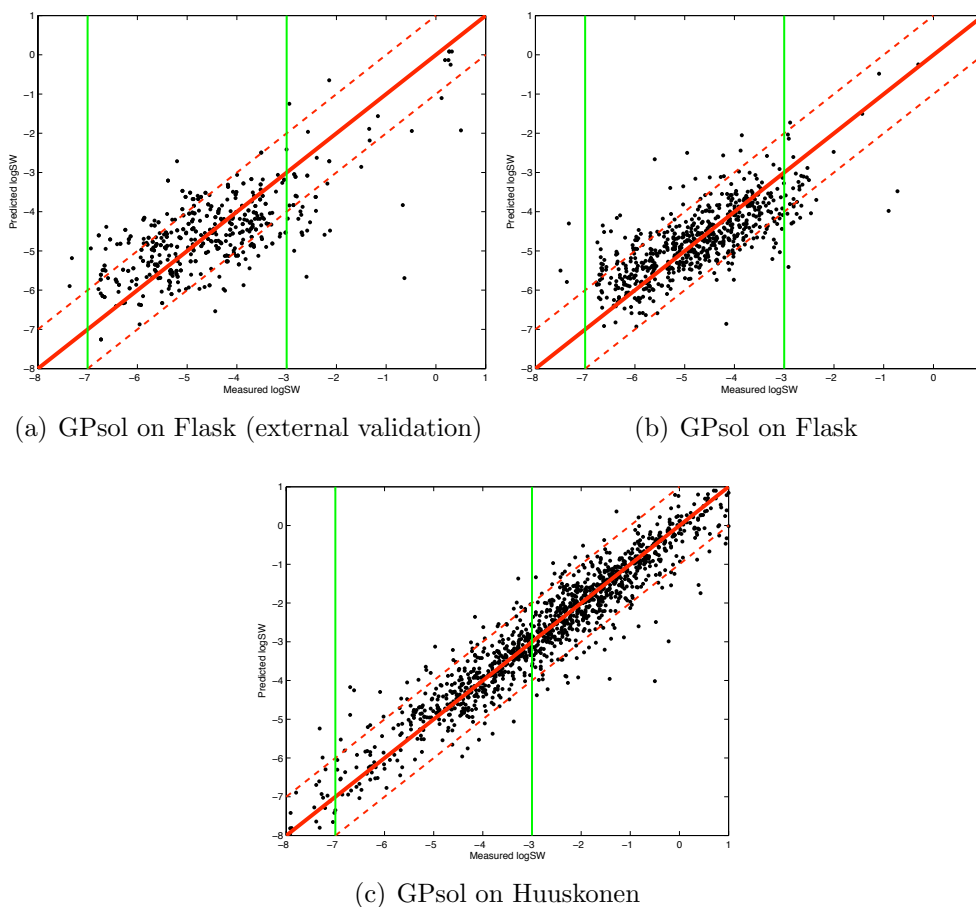
From a machine learning perspective, the modeling task can be summarized as follows:

- supervised regression task
- training data: 626 in-house compounds (the “Flask” dataset) and 3500 public compounds
- blind<sup>3</sup> test data: 536 compounds (the “Flask external” dataset)
- data representation: 1664 dimensional vectors [123]
- learning algorithms investigated: Gaussian Processes, support vector machines, random forests, ridge regression

We chose the mean absolute error, root mean squared error and the percentage of compounds predicted correctly within one log unit as numerical performance indicators. On the “Flask” dataset, ACDLabs v9 performs as follows: 0.90, 1.16 and 64 %. Our own final Gaussian Process Model produces better results, namely 0.60, 0.77 and 82 %. In the blind test scenario, when applying both models to all compounds in the “Flask external” dataset (regardless of the models domain of applicability, as discussed below), one obtains the following performance estimates for ACDLabs v9 : 0.98, 1.24 and 58 % and for our own final Gaussian Process Model: 0.73, 0.93 and 75 %.

---

<sup>3</sup>Initially, performance was estimated in cross-validation on the training data. Later, the final model was evaluated by a group of researchers who were not involved in model building, using a set of new measurements that had become available in the meantime. Hence the term “blind test”, see also Sec. 3.4.



**Figure 5.3:** Scatterplots for the GPsoL models on each of the three datasets described in [8, 11]. The vertical green lines mark the “fit-for-purpose” range [151, 152] to assess the performance of models in the  $\log S_W$  range relevant to drug discovery.

Both the “Flask” and “Flask external” sets of data are “fit-for-purpose” for drug discovery<sup>4</sup>, which also means that they are more difficult to model than

<sup>4</sup>In drug discovery projects, aqueous solubility is typically between 0.1  $\mu\text{g/L}$  and 250  $\mu\text{g/L}$ . For a compound with a molecular weight of 500 g/mol this corresponds roughly to the  $\log S_W$  range from  $-7$  to  $-3.5$ . Delaney [152] observed that a lot of models in the literature are trained on public datasets spanning more than ten orders of magnitude. Compounds with low  $\log S_W$  are usually harder to predict than soluble ones, nevertheless statistics are typically presented for the whole range of  $\log S_W$ . Delaney suggests that studies should be assessed using an element of “fit-for-purpose” (FFP). Johnson et al. [151] picked up the suggestion and evaluated a number of studies, taking into account the performance of the models in the  $\log S_W$  range from  $-7$  to  $-3$ .

Johnson’s FFP range is indicated by vertical green lines in Figure 5.3. Only 37% of the compounds in the “Huuskonen” set are in this range. On the other hand more than 90%

many public benchmark sets, including the Huuskonen set of data [135]. In addition to “just being difficult”, they also exhibit characteristics that can be taken as hints that training and test data are not sampled from the same distribution<sup>5</sup>. In the machine learning community, this phenomenon is called “covariate shift”. One could argue that aqueous solubility additionally possesses characteristics typical of the “multiple modes of action” phenomenon, because relatively similar molecules can adopt quite different crystal structures and the stability of the crystal influences the observed solubility. Both phenomena are discussed in Sec. A.1.2. The observations hinting at covariate shift are:

- In a PCA visualization (see Figure 3.2 in Sec. 3.2), the “Flask external validation” data are projected onto a small subspace of the 2 D plot.
- In Figure 5.3(a) GP<sub>sol</sub> predictions appear to be “vertically compressed”.<sup>6</sup>
- Inspecting Figure 5.3 we find that a lot of points representing predictions for the “Flask” and “Huuskonen” sets are very close the diagonal (i.e., very accurate). The spread is much larger in case of the “Flask external validation” setup.
- In Tab. 2 in [8] we can see clearly that the performance decreases when comparing “Flask external” with the cross-validation results on “Flask”.

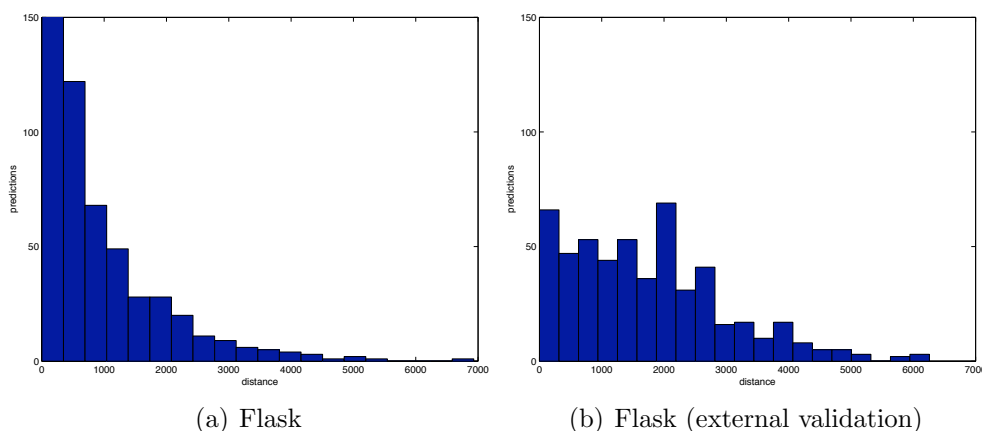
Histograms of Mahalanobis distances from each compound to the closest compound in the respective training set are shown in Figure 5.4. Distances for the cross validated “Flask” setup were calculated for the training/validation-split of one arbitrarily chosen cross validation run. Distances were calculated based on the same set of descriptors that was used to build the models. In the investigated training/validation-split of the “Flask” validation setup, 97% of the compounds have at least one neighbor at a distance smaller than 1500 units. In the “Flask external validation” setup, only 48% of the compounds have neighbors closer than 1500 units. This clearly confirms that “Flask external” is a set of compounds that is, to a large extent, structurally dissimilar

---

of the compounds in the two in-house data sets from Bayer Schering Pharma (compounds from drug discovery projects) are in the FFP range (94% of the “Flask” dataset and 91% of the “Flask external validation” dataset, respectively).

<sup>5</sup>The decrease in performance observed when looking at external validation data could be taken as a hint that the more complex models did over-fit their training data. We did, however, not observe typical symptoms of over-fitting, e.g. a too large number of support vectors in a support vector regression model.

<sup>6</sup>By its construction, predictions from the Gaussian Process model get closer to the mean  $\log S_W$  when new compounds are more dissimilar to those in the training set. At the same time, the size of the predicted error bars increases.



**Figure 5.4:** Histograms of Mahalanobis distances from each compound to the closest compound in the respective training set. Distances for the cross-validated “Flask” setup were calculated for the training/validation-split of one arbitrarily chosen cross-validation run.

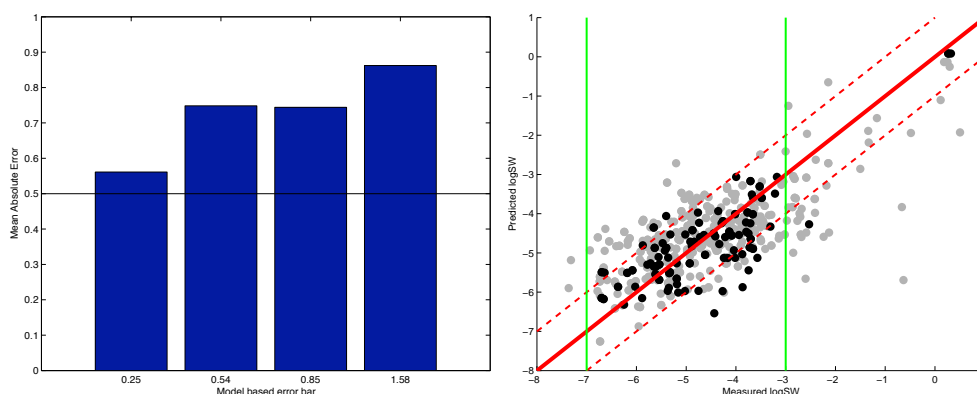
to the training data in “Flask”. Thus, we can assume that the decrease in performance is caused by a large number of compounds being dissimilar to the training set compounds and we are indeed observing the covariate shift phenomenon as discussed in Sec. A.1.2. Many compounds are thus outside of the models’ respective domains of applicability. If this assumption holds, it should be possible to achieve higher performance by rejecting compounds that are outside the domain of applicability:

When new compounds are dissimilar to those in the training set, predictions from the Gaussian Process model get closer to the mean  $\log S_W$ . At the same time, the size of the predicted error bars increases. In Figure 5.5 (right hand side) we present a scatterplot for GPsol on the “Flask external” validation set. Black points represent the confident predictions, whereas grey points represent the less confident predictions with predicted error bars larger than 0.6 log units. Vertical compression can be observed in the cloud of grey points, but is not present in the cloud of black points. Thus, we conclude that the vertical compression observed in Figure 5.3 is indeed caused by compounds that are dissimilar to the training set.

Consequently, performance statistics can be improved by focussing on the most confident predictions. Figure 5.5 (left hand side) shows a staircase plot. We see that performance statistics, in this case the mean absolute error, can indeed be improved. In fact, they can be improved to the level previously estimated using cross-validation on the “Flask” set of data (see Tab. 2 and Tab. 3 in [8]).

Table 3 in [11] shows compounds from the Huuskonen set of data that





(a) GPsol on Flask (external validation) (b) GPsol on Flask (external validation)

**Figure 5.5:** *Left hand side:* Mean absolute error achieved when binning by confidence estimates produced by GPsol. Each column represents one fourth (133 compounds) of the “Flask external” validation set. We observe that the MAE can be reduced significantly by focussing on the most confident predictions (leftmost bin). *Right hand side:* Scatterplot for GPsol on the “Flask external” validation set. Black points represent confident predictions, grey points represent less confident predictions with predicted error bars larger than 0.6. Dashed red lines indicate the range true value  $\pm 1$  log unit. We observe that the black points deviate less from the ideal prediction (red diagonal line) than the grey points, i.e. confidently made predictions are visibly closer to the true value.

are mispredicted by our GP model and where the prediction error is outside of the 99 % confidence interval. It turned out that the two major reasons for these mispredictions are low data quality (e.g., contradictory measurements) and inherent limitations caused by the molecular descriptors (two compounds with different solubility but almost identical descriptors). In some cases of contradictory measurements it was possible to identify the true value and correct the respective entry in the database. See the last paragraph of Sec. 3.2 and [11] for details on how this analysis was carried out.



## 5.4 CYTOCHROME P450 INHIBITION

The five most important members of the Cytochrome P450 superfamily of heme-containing monooxygenases, namely CYP 1A2, 2C19, 2C9, 2D6 and 3A4, are responsible for clearance of more than 90% of all marketed drugs [153–156]. Their inhibition can result in unwanted drug-drug interactions, making it desirable to identify problematic compounds as early as possible in the drug design process. Quantitative Structure Activity Relationships (QSAR), pharmacophore modeling and homology modeling have all been used to generate predictions for molecules binding either as substrates or inhibitors. Also, several groups have combined pharmacophores with homology models [157]. However, many parts of the mechanisms of catalysis, activation and inhibition are still poorly understood [154]. All of the above mentioned approaches aim at understanding CYP inhibition via explicit models of the inhibition process. In contrast, approaches based on statistical learning have been used in these studies [158–160]. In 2005, the study published by Kless et al. [160] was the only publication about application of machine learning methods to data about all of the five most important CYP isoenzymes. The general consensus is that CYP inhibition is a very difficult modeling problem.

The following section describes how predictive models for inhibition of each of five cytochrome P450 enzymes were constructed in collaboration with researchers at Schering. In 2006, CYP 2D6 was first crystallized and its 3D structure determined, following the previous discoveries of the 3D structures of CYP 2C9 and CYP 3A4 [161–163]. This knowledge allows for protein structure based approaches (such as docking) to be used in identification of CYP inhibitors. The study described in the following was started in 2005 and employed a ligand based approach.

From a machine learning perspective, the modeling task can be summarized as follows:

- supervised multiclass classification / ordinal regression task
- training data:  $\sim 800$  compounds with labels for each of 5 CYP subtypes, blind<sup>7</sup> test data:  $\sim 170$  compounds with labels for each of 5 CYP subtypes
- labels: compounds, are labeled as weak, moderate or potent inhibitors of the respective CYP subtype, see Sec. A.2 for detailed definitions.

---

<sup>7</sup>Initially, performance was estimated in cross-validation on the training data. Later, the final model was evaluated by a group of researchers who were not involved in model building, using a set of new measurements that had become available in the meantime. Hence the term “blind test”, see also Sec. 3.4.

- source: most compounds are in-house compounds of Schering, but for some compounds, measurements have been curated from the literature (see Sec. A.2 for the exact composition of each dataset). As the literature compounds span a larger part of the chemical space, it is not clear whether including this data can improve predictions for future in-house compounds.
- data representation in first modelling effort: The following six vectorial representations were used, see Appendix A.3 for a description of each type of representation: Ghose-Crippen [164]<sup>8</sup>, BCUT [166, 167], UNITY fingerprints [168], LogD prediction module from ACD Labs [75], VolSurf [169], GRIND [170]
- data representation in second modelling effort: 1664 dimensional vectors generated using the Dragon descriptor generator [123]
- learning algorithms investigated in first modelling effort: decision trees [171], k-Nearest Neighbor (kNN) [172], Gaussian Processes [43], Support Vector Machines (SVM) [38, 173, 174], Linear Programming Machines (LPM) [175, 176].
- learning algorithms investigated in second modelling effort: Gaussian Processes

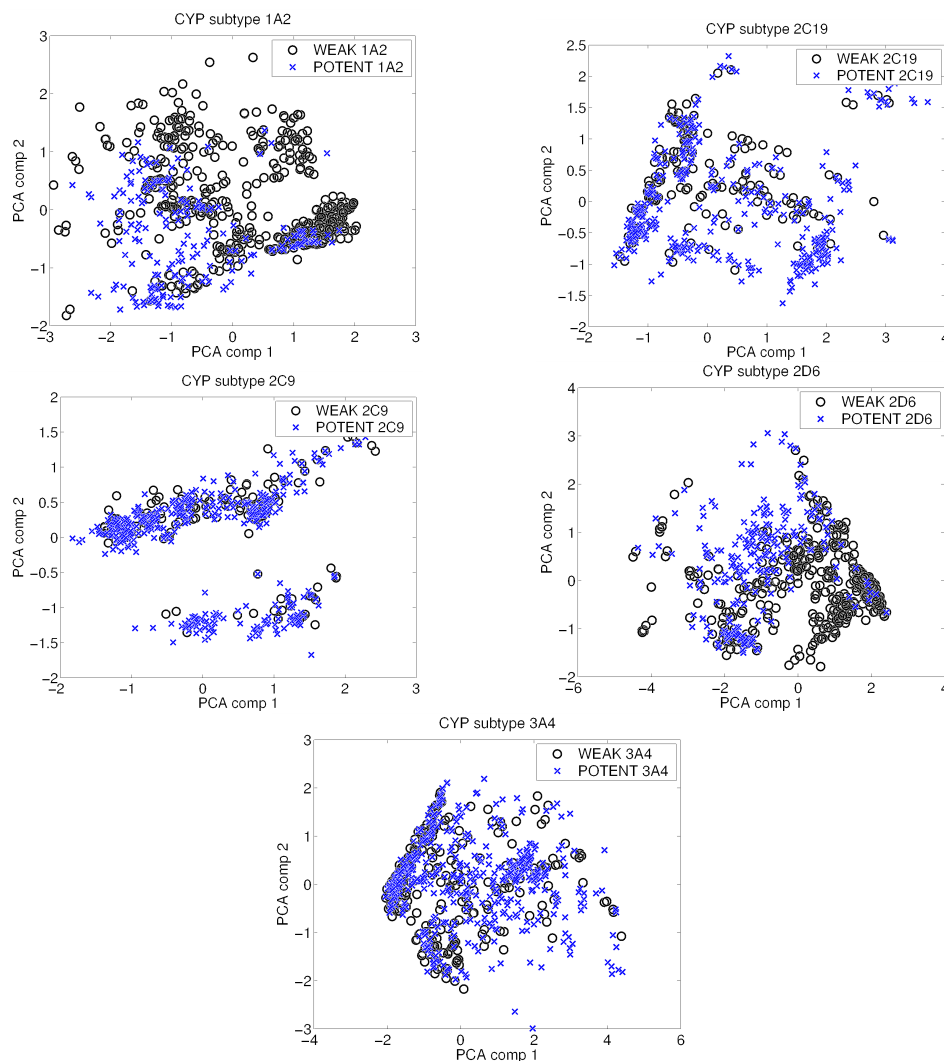
PCA plots using all descriptors together can be found in Figure 5.6. Some structure can be recognized, but in two dimensions even the two most different classes (only weak and potent inhibitors are shown, moderate inhibitors are omitted) do not look separable. Therefore it is unclear whether classification using many dimensions will be possible.

For each of the five CYP isoenzymes, two classifiers are build: The first discriminating between weak inhibitors and non-weak (i.e., moderate and potent) inhibitors (Task 1), the second discriminating between potent and non-potent (moderate and weak) inhibitors (Task 2). For each of these classifiers, it is not clear which descriptor and learning method is best suited for the respective classification task. Thus, classifiers for each combination of descriptors and learning methods were constructed. As a further subdivision, classifiers from the in-house data, and from the union of in-house and literature data were investigated.

As on single sets of descriptors the best results were achieved using support vector machines with RBF kernels, this method was also chosen for

---

<sup>8</sup>It was checked whether using Fisher Scores [165] of the Ghose-Crippen [164] descriptors facilitates modeling. The accuracy achieved was very similar in both cases, therefore the plain Ghose-Crippen descriptors were used in the remaining part of the study.



**Figure 5.6:** PCA visualizations of the CYP inhibition data. For each subtype, we project the data onto the first two PCA components, after removing descriptor dimensions that are uncorrelated with the class label.

evaluating combinations of descriptors. This was done by concatenating the preprocessed features and then proceeding as described for single descriptors. Considering all subtypes, both datasets (in-house only and including external) and both types of classifiers (potent vs. non-potent, weak vs. non-weak) there are  $15 * 5 * 2 * 2 = 300$  pairs and  $20 * 5 * 2 * 2 = 400$  triples of descriptors theoretically to be considered. A greedy approach was employed in choosing combinations to avoid making too many experiments. It was found that simply using all descriptors together works as well as the best combinations of descriptors. However, Volsurf and GRIND descriptors being computationally

	1A2	2C19	2C9	2D6	3A4
Overall	89.16	80.94	NaN	91.92	NaN
Schering compounds	84.35	81.91	77.26	89.16	79.58
non-Schering compounds	91.72	73.41	65.30	84.04	68.73

**Table 5.1:** Performance of classifiers for WEAK compounds in each CYP subtype (task 1). The figures given are the area under the ROC curve in %

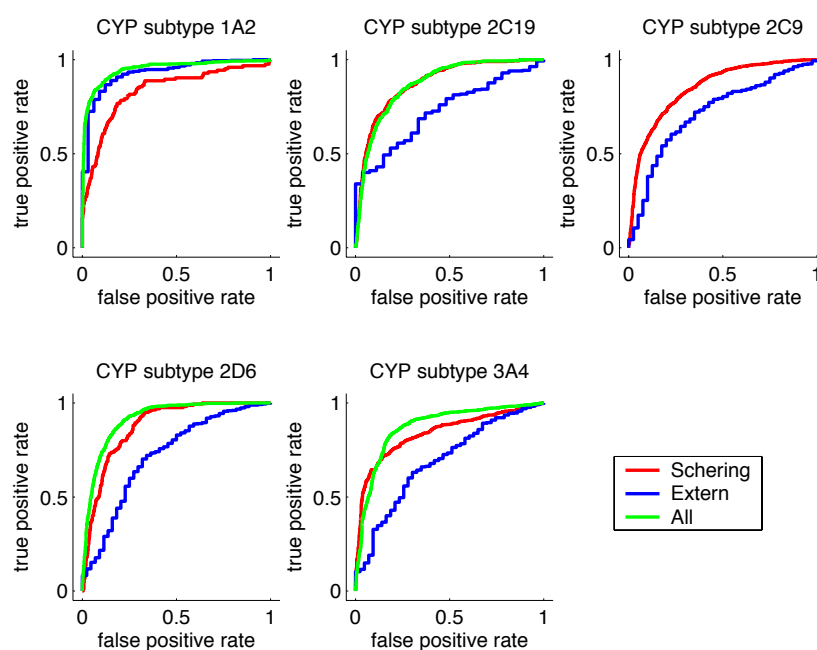
	1A2	2C19	2C9	2D6	3A4
Overall	95.06	87.15	NaN	91.61	87.34
Schering compounds	83.60	87.59	84.96	87.82	83.93
non-Schering compounds	93.07	72.78	71.79	72.01	68.81

**Table 5.2:** Performance of classifiers for POTENT compounds in each CYP subtype (task 2). The figures given are the area under the ROC curve in %

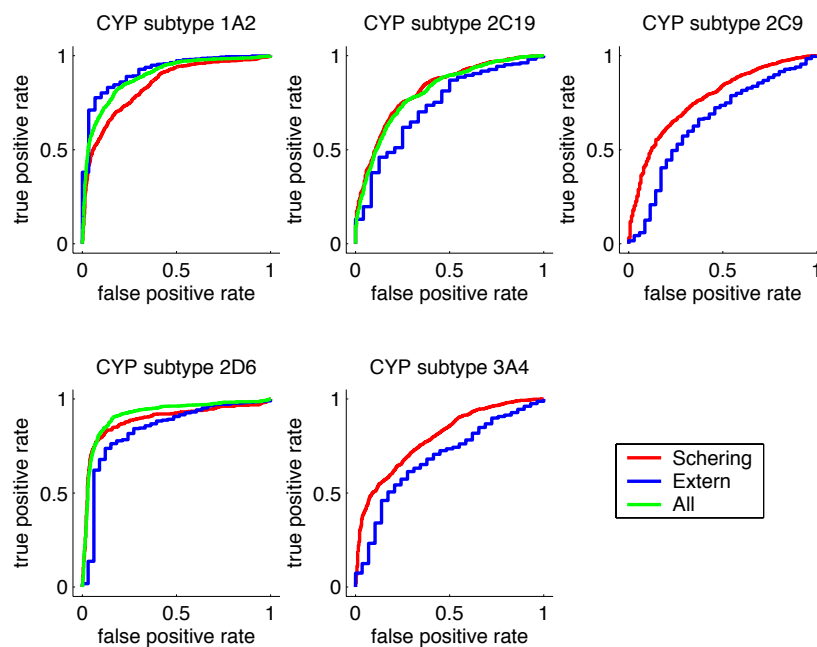
expensive, it was decided to use the combination of Ghose Crippen and ACD descriptors for the (then thought to be final) model. ROC curves for these models are shown in Figure 5.7 and 5.8, the performance is summarized in Table 5.1 and 5.2.

Simulations showed that for most CYP subtypes a global model is superior to separate models for public and in-house data. However, for detecting potent CYP 2C9 inhibitors, using separate models for public and in-house compounds turned out to be more reliable. Likewise, for identifying weak inhibitors, separate models for the subtypes 2C9 and 3A4 are beneficial. For Task 1, an overall performance of about 87% was achieved, whilst classifying the public data was generally more difficult than classifying the in-house compounds. This, however, might be due to the fact that the amount of available external training data was much smaller than the number of compounds available from the in-house subspace. For Task 2, an overall performance of 87% to 95% was achieved. Here, as well, classifying the external data was more difficult.

A blind test (Sec. 3.4) was conducted using  $\sim 170$  compounds from recent projects. Despite the promising impression that the training data are difficult, but modeling is feasible, the results of this test were not satisfying and it was decided to evaluate the Dragon software for descriptor generation, because it provides a very large ( $> 1600$ ) and diverse set of descriptors. Different ways of preprocessing the respective Dragon blocks were investigated and several ways of selecting individual descriptors, whole blocks of descriptors and combinations thereof were applied. Finally, a Gaussian Process model based on certain 10 blocks of Dragon descriptors was constructed and evaluated in a second blind test. This new model can provide a moderate



**Figure 5.7:** ROC curves for predicting POTENT CYP inhibitors in cross-validation on the training data.



**Figure 5.8:** ROC curves for predicting WEAK CYP inhibitors in cross-validation on the training data.

enrichment when ranking large libraries of compounds, but is not yet useful for interactive use by bench chemists in lead optimization: The individual predictions are often incorrect and the confidence estimates from the Gaussian Process model are not strongly enough correlated with these deviations to be able to compensate. This unsatisfying performance probably results from the four facts that

- Cytochrome P 450 inhibition is an inherently difficult modeling problem,
- the amount of training data available was relatively small,
- some of the enzymes have large flexible active sites, allowing for multiple binding modes (see Sec. A.1.3),
- test data was sampled from newly explored regions in chemical space that were not covered well by the training data (covariate shift, see Sec. A.1.2)

and the confidence estimates of the Gaussian Process models were not accurate enough to identify reliable predictions.



## 5.5 METABOLIC STABILITY

In the drug development process, 50% of the failures[133] in late development stages are due to an unfavorable ADMET profile (Absorption, Distribution, Metabolism, Excretion & Toxicity). Groups across the world have invested a lot of research effort in obtaining *in silico* predictions for properties that are closely related to the ADMET profile. See references in the sections on partition coefficients (Sec. 5.2), aqueous solubility (Sec. 5.3), cytochrome P450 inhibition (Sec. 5.4), toxicity (Sec. 5.6) and the hERG channel blockade effect (Sec. 5.7). Furthermore, commercial tools are available for most of the above mentioned properties, excluding only the hERG channel blockade effect and metabolic stability. If a compound is metabolically unstable, more than half of the drug molecules are lost during the first half hour in the human body. In order to be able to reach and maintain efficacious concentrations of the compound in the tissue(s) where the target receptors are located, one would (theoretically) have to give very large doses of the drug. Large doses will, on the other hand, increase the risk of unwanted side effects. Building general-purpose models that are accurate over a large number of structural classes is virtually impossible, since a plethora of not fully understood mechanisms is involved in metabolizing a chemical compound in the human liver. Furthermore, experimental protocols and assays can vary widely, such that tool predictions and actual experimental outcome may exhibit large differences. Only when the classes of compounds are limited, and experimental techniques are very homogeneous one can hope to establish Quantitative Structure Property/Activity Relationship (QSPR) models that reliably predict a property like metabolic stability. Preceding the study summarized in this section, there was only little published work about such approaches[177, 178], despite development efforts by various pharmaceutical companies.

The following section describes how predictive models for metabolic stability were constructed in collaboration with researchers at Bayer Schering Pharma. Separate paragraphs discuss how extreme noise in the real valued labels was dealt with and how prior knowledge was used to identify outliers in labels, issues leading to a limited domain of applicability of the model and lastly how confidence estimates produced using Gaussian Process models are effective in identifying compounds for which reliable predictions can be made. A more detailed description of the results of modeling metabolic stability using different machine learning methods and both in-house and public data sets can be found in [6].

From a machine learning perspective, the modeling task can be summarized as follows:

- supervised regression task
- extremely noisy labels (see next paragraph for a discussion)
- training data: from 900 up to 1900 compounds for each of four separate species<sup>9</sup>, blind<sup>10</sup> test data: from 190 up to 700 compounds for each species
- data representation: 1664 dimensional vectors generated using the Dragon descriptor generator [123]
- learning algorithms investigated: Gaussian Processes for classification and regression, support vector machines for classification and regression, ridge regression

The metabolic stability of compounds treated in this study was assessed by measuring the percentage of each compound remaining after incubation with liver microsomes of humans, rats and mice, respectively, for 30 minutes. The procedure is described in detail in [6]. For the moment let us just note that measurements should, theoretically, span the range between 0 % and 100 %. In practice, this is not the case: Not only are these measurements very noisy (errors  $\pm 20$  % are in order), but there are also issues like too slow dissolution: In these cases, the compounds are only partially dissolved when the incubation is started and continue to dissolve during the incubation period. This can result in measurement values exceeding 150 %. It was decided to filter out these most extreme measurements and otherwise treat metabolic stability as a classification problem. Later, the performance of regression algorithms was also investigated and it was found that the ranking performance<sup>11</sup> increases only insignificantly.

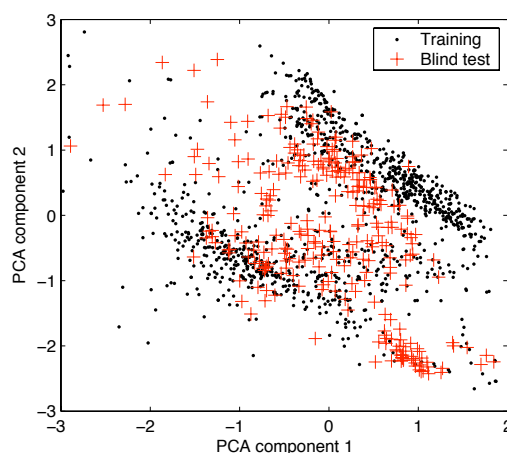
Figure 5.9 illustrates the differences between training and blind test data via a principal component analysis (PCA) plot. PCA was computed on the descriptors for the training data, afterwards the blind test data were projected into the same PCA coordinate system. The plot shows that training and blind test data contain compounds that are structurally different. There are many different enzymes present in the microsomal preparation

---

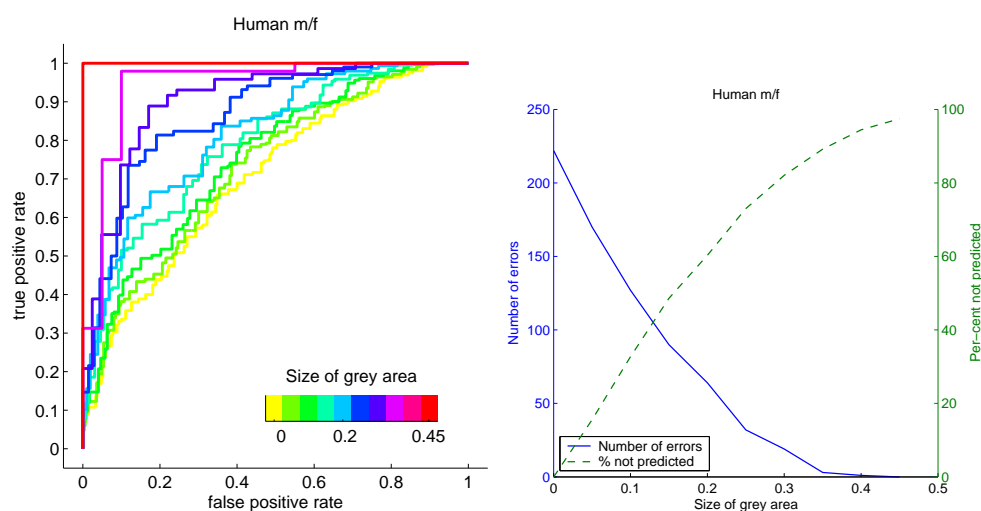
<sup>9</sup>Measurements of metabolic stability were available for the following four species: humans, male mice, female mice and male rats.

<sup>10</sup>Initially, performance was estimated in cross-validation on the training data. Later, the final model was evaluated by a group of researchers who were not involved in model building, using a set of new measurements that had become available in the meantime. Hence the term “blind test”, see also Sec. 3.4.

<sup>11</sup>Both Gaussian Process Classification models and Support Vector Machines for classification produce real valued output that allows to rank compounds.



**Figure 5.9:** Visualization of training and blind test data for the assay “mouse female” by principal component analysis (PCA). The blind test data covers recent projects, and thus follows a distribution that is different from the training data



**Figure 5.10:** The output of a Gaussian Process Classification model is close to 0.5 if predictions are made inconfidently. ROC curves resulting from rejecting predictions falling inside “grey areas”, i.e. intervals  $[0.5 - q \dots 0.5 + q]$  are presented in the left hand side plot. At the same time, the number of rejected compounds increases (right hand side plot). We can see that the shape of the ROC-curves (left hand side) improves continuously as  $q$  is increased, finally reaching almost 100 % correct classification (red curve at the top).

used for the measurements and some of the enzymes even have different active sites. Therefore the models are simultaneously facing both a sampling issue (Sec. A.1.2) and multiple mechanisms. The next paragraph explains how reliable predictions can be identified.

The output of a Gaussian Process Classification model is the probability that the compound belongs to class 1. It becomes closer to 0.5 as the distance to the training set increases and actually is 0.5 for compounds that are far from the training set. Therefore a natural way to reject compounds that are outside of the domain of applicability is to define some desired level of confidence and then reject all predictions in the interval  $[0.5 - q \dots 0.5 + q]$ . Receiver operating curves for different values of  $q$  in the range  $[0 \dots 0.45]$  are shown in Figure 5.10 (left). At the same time, the number of rejected compounds increases, see Figure 5.10 (right). We can see that the shape of the ROC-curves improves continuously as  $q$  is increased, finally reaching almost 100 % correct classification (red curve at the top).

In conclusion, one can use prior knowledge to identify outliers in the training data, the domain of applicability of models is limited due to sampling issues and multiple mechanisms, but using a Gaussian Process Classification model one can predict the metabolic stability of compounds in the form of a probability that the respective compound is stable. By leaving out unsure predictions where the predicted probability is close to 50 %, performance can be improved to the point of being almost perfect.

## 5.6 AMES MUTAGENICITY

The bacterial reverse mutation assay (Ames test [179]) to detect mutagenicity in vitro is of crucial importance in drug discovery and development as an early alerting system for potential carcinogenicity and/or teratogenicity. In the Ames test, frame-shift mutations or base-pair substitutions may be detected by exposure of histidine-dependent strains of *Salmonella typhimurium* to a test compound. When these strains are exposed to a mutagen, reverse mutations to the wild-type histidine-independent form enable bacterial colony growth on a medium deficient in histidine ("revertants"). Since many chemicals interact with genetic material only after metabolic activation by enzyme systems not available in the bacterial cell, the test compounds are in many cases additionally examined in the presence of a mammalian metabolizing system, which contains liver microsomes (with S9 mix, see [5]). Existing commercial tools suitable for predicting the outcome of the Ames test, such as DEREK and MultiCASE, provide promising results on several evaluation data sets and the possibility to derive structure-activity and/or even mechanistic information from the predictions. Still, these commercial tools are limited in terms of statistical performance, technical accessibility for bench chemists and adaptability to a company's chemical space. In the public literature, several approaches have been followed to predict Ames mutagenicity, generally yielding good specificity and sensitivity values (prediction accuracy of up to 85%). Depending on the descriptors and the statistical methods used, some of the models offer structure-activity information, such as Helma et al. [180] or Kazius et al. [181], some are however harder to interpret due to their choice of chemical descriptors, such as Feng et al. [182]. Moreover, different data sets have been used in the respective studies [180–182] without disclosing the splits (training set / test set) used for model evaluation.

The following section describes how a large unique benchmark set was collected (and published) and predictive models for Ames mutagenicity were constructed in collaboration with researchers at Bayer Schering Pharma. An evaluation of several machine learning algorithms and three commercial tools is presented. The benchmark data set was used for developing two new frameworks for interpreting results and increasing the acceptance of models by bench chemists. Sec. 4.3 *explains* predictions in terms of the most relevant compounds in the training set,. In contrast, Sec. 4.4 explains predictions in terms of the locally most relevant features. These are identified using local gradients which, at the same time, provide hints that can help in *compound optimization*.

From a machine learning perspective, the modeling task can be summarized as follows:

- supervised classification task
- evaluation strategy: Five fold cross-validation, each time including a *static* training set (see below).
- data representation: 904 dimensional vectors (blocks 1, 2, 6, 9, 12, 15, 16, 17, 18 and 20 of DRAGON-X version 1.2 [123] based on a 3D structure generated by CORINA version 3.4 [123])

To allow for a reasonable comparison of different methods, we assembled a new benchmark set of 6512 compounds together with their Ames test results from public sources. As described in [5], we make this large unique benchmark set - including well-defined random splits - publicly available to facilitate future comparisons with prediction methods of other researchers.

Using the benchmark data set, four machine learning techniques and three commercial prediction tools were evaluated. For the non-commercial machine learning methods we considered a support vector machine, a Gaussian Process, a random forest and a k-nearest neighbor model. For the machine learning algorithms, short descriptions and references to the literature can be found in Sec. 3.3). The commercial tools are described in the following:

- Pipeline Pilot’s Bayesian categorization model[183] provides supervised Bayesian learning for large data collections. In our example, we have combined this Bayesian classifier with Pipeline Pilot’s ECFP chemical fingerprint technology (ECFP\_4 fingerprints).
- DEREK (Version 10.0.2 Service Pack 3, Knowledge Base Release DfW 10.0.0\_25\_07\_2007, Lhasa Ltd., UK) is an expert system providing known structure-activity relationships (SARs), each relating to a toxicological endpoint. The mutagenicity prediction by DEREK was considered positive if a structure triggered at least one mutagenicity alert, and negative if it did not raise any mutagenicity alerts. Compounds contained as examples in the knowledge base of DEREK were excluded from the evaluation[184].
- MultiCASE (Multicase Inc., USA) is a correlative tool predicting toxicity on the basis of structural fragments statistically correlated with activity (QSAR). For mutagenicity prediction, the commercially available AZ2 mutagenicity module was used. Compounds contained in the training set of the AZ2 module were excluded from the evaluation[185].

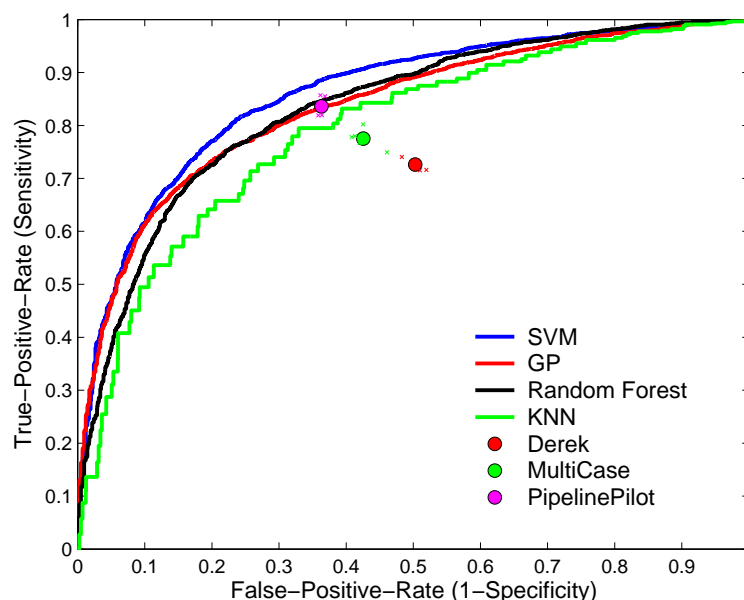
All models were evaluated in a 5-fold cross validation setting. Firstly all compounds which were verifiably known to DEREK or MultiCASE were

Model	AUC
SVM	$0.86 \pm 0.01$
GP	$0.84 \pm 0.01$
Random Forest	$0.83 \pm 0.01$
k-Nearest Neighbor	$0.79 \pm 0.01$

**Table 5.3:** Cross validation results for parametric classifiers.

pooled together in a static training set. The remaining data set was divided into five cross validation splits. Within each step of the cross validation all models were trained on a set of four cross validation splits together with the static training set (at least 5525 compounds). The fifth split forms the validation set. To select the parameters for the machine learning algorithms an inner loop of cross validation was performed on the training settings.

We measured the quality of the resulting models using the Receiver Operating Characteristic (ROC, see Figure 5.11). In an ROC graph the false positive rate ( $1 - \text{specificity}$ ) is plotted against the true positive rate (sensitivity). The point (0,1) in this diagram marks a perfect classifier; at (0,0) all samples are classified as negative and in (1,1) all samples are assigned to the positive class.



**Figure 5.11:** Receiver operating characteristic (ROC) curve / point for each model.

Cross validation results are presented in Figure 5.11 and Table 5.3. For a presentation and discussion of sensitivity & specificity see [5]. PipelinePi-

lot, trained with the developed data set, shows the best results of the three commercial tools followed by MultiCASE. The expert system DEREK shows the lowest sensitivity and specificity of all considered models. MultiCASE and DEREK cannot take advantage of the rich information provided by the training data. They are based on a fixed set of mainly 2D descriptors (MultiCASE) or a static system of rules derived from a largely unknown data set and expert knowledge (DEREK). Moreover, the rules contained in DEREK may in part be too generic to reflect the influence of the chemical neighborhood of a functional group on their mutagenic activity. It can be assumed that there are unknown structure activity relationships which are not contained in the DEREK knowledge data base. The employed AZ2 model of MultiCASE cannot be adapted to a specific chemical space and therefore yields a lower prediction accuracy. Nevertheless, DEREK and MultiCASE are still essential for drug discovery and development as they provide structure-activity and/or mechanistic information essential for structure optimization and regulatory acceptance.

The machine learning algorithms in contrast exclusively derive their knowledge from the training data. The fact that none of the other tools could outperform one of the machine learning models (5.11) indicates the power of the latter approaches and the higher information content of the provided benchmark data set. The rather good performance of the simple k-Nearest Neighbor model indicates a strong influence of small local molecular changes on Ames mutagenicity. However the application of more sophisticated machine learning methods results in a significant performance gain especially for the support vector machine.

In conclusion, all five evaluated machine learning methods (SVM, Gaussian Process, Random Forest, k-Nearest Neighbors and the commercial Pipeline Pilot) yield good results on the benchmark data set. The future evaluation of additional prediction methods on the published benchmark data set represents a promising strategy for further optimization of Ames mutagenicity prediction. Furthermore, scientists interested in method development may benefit from the present work as all modeling and evaluation results obtained using the new data set allow for a direct comparison of different methods. In fact, work in this direction has already begun. A joint publication summarizing the results of a challenge on this benchmark set is currently being written by members of several research groups. Future development will strive for improving the accuracy of machine learning based prediction tools that yield interpretable results. First steps in this direction are presented in Sec. 4.3 and 4.4.



## 5.7 hERG CHANNEL BLOCKADE EFFECT

In recent years, avoiding drug induced cardiac arrhythmia has become an important optimization parameter during the discovery and development of new drugs [186, 187]. One of the most common issues is the prolongation of the QT interval<sup>12</sup> by blocking the human ether-a-go-go related gene-encoded potassium channel (hERG channel) expressed in cardiac muscle cells [188, 189]. QT prolongation enhances the risk of potentially fatal torsades de pointes. A number of drugs that were withdrawn from the market due to QT prolongation such as terfenadine or cisapride were shown to cause an unwanted blockade of the hERG channel. Following the “fail fast – fail cheap” paradigm of drug discovery it is highly desirable to identify compounds which exhibit hERG inhibition early in the discovery process [190]. In this context, in-silico methods have been developed and established to either cope with limited capacities for in-vitro testing or to assess virtual compounds. For a survey on computational efforts towards a model for hERG blockade, comprising homology models, pharmacophore approaches and QSAR models, the reader is referred to recent reviews [191–196]. Various modern machine learning methods which relate molecular descriptors with biological activities are available, and techniques like support vector machines (SVMs [38]), artificial neural networks [126], or, more recently, Gaussian Processes [43] (GPs), have been applied to address drug absorption, distribution, metabolism, excretion, or toxicity and hERG inhibition [121, 197–201]. Regression models based on public domain hERG data sets in general exhibit predictive powers between  $r^2 = 0.5$  and  $r^2 = 0.7$  (estimated from cross-validation experiments or predictions for independent test set molecules).

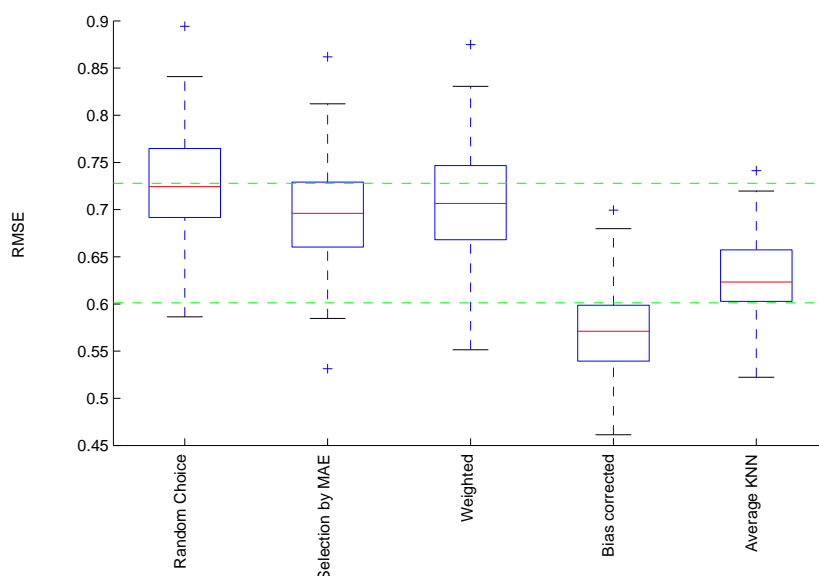
The following section describes how predictive models for the hERG channel blockade effect were constructed in collaboration with researchers at Boehringer Ingelheim Pharma GmbH. It focuses on the aspect of using the hERG prediction problem as a test bed for evaluating several techniques for fusing predictions from multiple models. Ensemble algorithms are introduced in Sec. 4.2. Different visualizations of the data generated in a pre-analysis are shown in Sec. 3.2. A more detailed description of the results of modeling can be found in [4].

From a machine learning perspective, the modeling task can be summarized as follows:

- supervised regression task
- dataset: 660 compounds, investigated in clustered cross-validation [4]

---

<sup>12</sup>The time between the start of the Q wave and the end of the T wave in the heart’s electrical cycle is called QT interval.

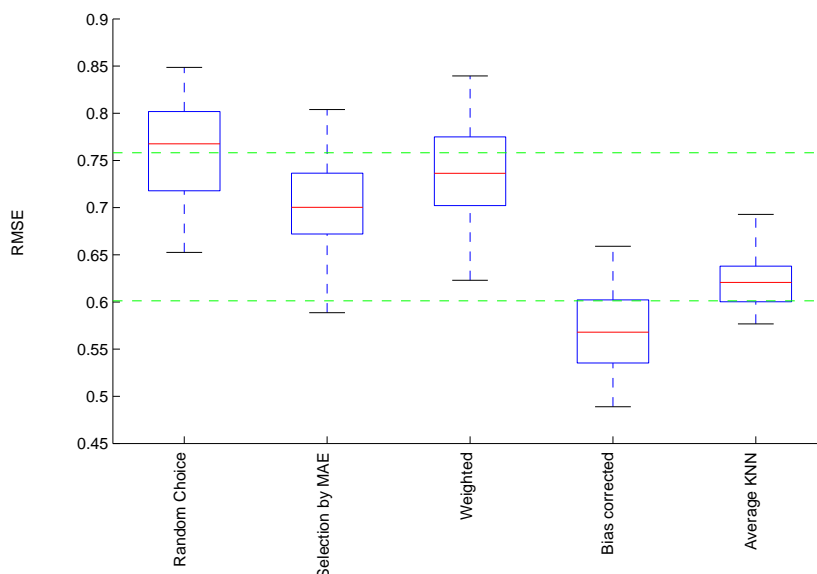


**Figure 5.12:** Combination of a Random Forest, a Gaussian Process and a SVR Model trained on equal sets: Box-plot depiction of the root mean squared error (RMSE) of the different ensemble methods in the clustered cross-validation setting over 50 repetitions. The upper dashed line refers to the RMSE of the underlying single Random Forest model, the lower dashed line marks the RMSE of a Random Forest model trained in leave-one-out cross validation. The ensemble methods are evaluated with respect to ten nearest neighbors of each compound. 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 the interquartile range. Points outside this range are marked as outliers.

- data representation: QSAR descriptors available in MOE (MOE 2007.09, Chemical Computing Group, Montreal, Canada), ChemAxon pharmacophoric fingerprints (ChemAxon Kft, Budapest, Hungary), CATS descriptors [202], VolSurf (vsplus 0.4.5a, Molecular Discovery Ltd, UK), using four standard chemical probes (water, hydrophobic probe, carbonyl oxygen, and amide nitrogen [203, 204]).
- learning algorithms investigated: Gaussian Processes regression, support vector regression, random forests, ridge regression
- ensemble algorithms investigated: “selection by MAE”, “weighted”, “bias corrected”, “average KNN”, “random choice”, see Sec. 4.2 for definitions.

#### *Combination of different single models trained on equal training sets*

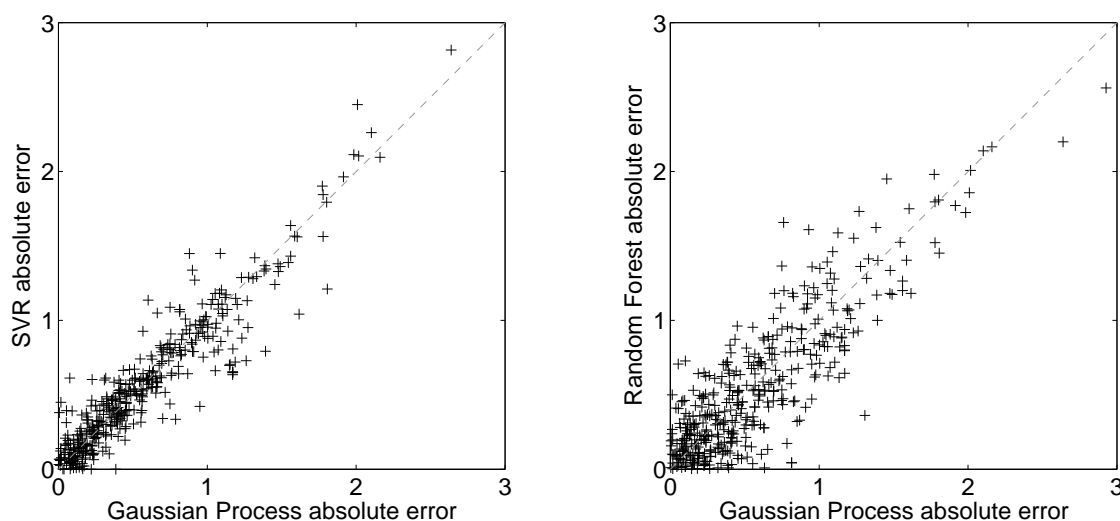
Figure 5.12 visualizes the distribution of the RMSE over 50 repetitions for the different ensemble algorithms described in Sec. 4.2 when training the



**Figure 5.13:** Combination of 20 classical bagging Random Forest Models trained on different sets: Box-plot depiction of the root mean squared error (RMSE) of the different ensemble methods in the clustered cross-validation (see Sec. 3.4 setting over 50 repetitions). The upper dashed line refers to the RMSE of the average underlying single Random Forest model, the lower dashed line marks the RMSE of a Random Forest model trained in leave-one-out cross validation. The ensemble methods are evaluated with respect to ten nearest neighbors of each compound. 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 the interquartile range. Points outside this range are marked as outliers.

three single models on identical data. The RMSE of the *Random Choice* model equals the RMSE of the single Random Forest Model. This was to be expected, because the single models (GP, SVR and Random Forest) perform about equally well. The *Weighted* as well as the *Selection by MAE* approach introduced by Kühne et al. [118] do not improve the performance significantly compared to the single Random Forest model (upper dashed line). The reason for this observation is illustrated in 5.14: The prediction errors of each individual model for the compounds are highly correlated. If one single model results in an inaccurate prediction the other two single models show equally large prediction errors and a compensation of prediction errors through a combination of the single models is hardly possible.

The local bias correction (in the *Average KNN* and the *Bias corrected*) model shows a comparably large improvement. For the latter one the mean RMSE is even smaller than the RMSE of a Random Forest model evaluated in leave-one-out cross validation on the whole correction set while also utilizing



**Figure 5.14:** 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).

the training set (lower dashed line). This result indicates that a local bias correction is even more important than the choice of the prediction method. Considering the fact that in the *Bias corrected* consensus model the single models are only trained on two thirds of the data set and not on nearly the whole data set like in the leave-one-out model this seems surprising. Interestingly, incorporating additional information about just the ten nearest compounds allows to reach this small RMSE.<sup>13</sup>

#### *Combination of equal single models trained on different training sets*

This subsection discusses the performance of ensemble models which combine the predictions of 20 Random Forests trained on different parts of the training set. Using a bagging approach, a different training set is constructed for each Random Forest and they are combined using the same ensemble algorithms and cross-validation setting as described in the previous section.

The main results of this evaluation are summarized in 5.13. The underlying single models are now trained on data sets with more variety. Due to the different training sets, the differences between each group of single models now occur on different clusters of compounds and some errors of the single

<sup>13</sup>Using ten nearest neighbors, saturation is reached for all algorithms employed. See [4] for details.

models can be compensated by choosing the best performing model included in the ensemble model. However, the distribution of the RMSE shows similar tendencies as in the previous setting: For the *Random Choice* model we observe a worse performance than a single model (upper dashed line) and the *Weighted* Model again only achieves a small improvement. In contrast to the previous observation, the *Selection by MAE* Model introduced by Kühne et al. [118] now achieves a somewhat larger improvement with respect to the single Model (RMSE reduced from 0.76 to 0.70). The *Bias corrected* Model again achieves the largest improvement of all ensemble methods (RMSE reduced from 0.76 to 0.57).

In conclusion, local bias correction may be a good way to cope with the strong locality of the hERG inhibition problem while still profiting from global trends in the data. Furthermore the experiments indicate practical hints for the use of machine learning models in drug design: When new measurements are added to an existing training set, a simple local bias correction can substitute retraining a whole SVR, Gaussian Process or Random Forest model on the expanded data set. Further investigation is necessary to evaluate in which cases the bias corrected approach is adequate and in which cases retraining should be preferred.



## 5.8 LOCAL GRADIENTS FOR EXPLAINING INDIVIDUAL CLASSIFICATION DECISIONS & GUIDING COMPOUND OPTIMIZATION

### *Overview*

In this thesis, two separate methodologies for explaining individual predictions of (possibly non-linear) machine learning models are presented. The method presented in Sec. 4.3 *explains* predictions by the means of visualizing relevant objects from the training set of the model. This allows human experts to understand how each prediction comes about. If a prediction conflicts with his intuition, the human expert can easily find out whether the grounds for the models predictions are solid or if trusting his own intuition is the better idea.

Sec. 4.4 proposes a method that sheds light into the black boxes of non-linear classifiers. In other words, it introduces a method that can explain the local decisions taken by arbitrary (possibly) non-linear classification algorithms. In a nutshell, 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, a probabilistic approximate mimic of the learning machine to be explained is employed.

To validate the new gradient based methodology, [1] shows how it can be used to draw new conclusions on how the various Iris flowers in Fisher’s famous dataset are different from each other (section 4 in [1]) and how to identify the features with which certain types of digits 2 and 8 in the USPS dataset can be distinguished (section 5 in [1]). In the following Subsection, the method is applied to a challenging drug discovery problem, namely the prediction of Ames mutagenicity (Sec. 5.6). Results fully agree with existing domain knowledge, which was not available to the method. Even local peculiarities in chemical space (the extraordinary behavior of steroids) was discovered using the local explanations given by the new approach.

### *5.8.1 Explaining Mutagenicity Classification by Gaussian Processes*

In the following section we describe an application of our local gradient explanation methodology to a complex real world data set. Our aim 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 in the high-dimensional feature space used to determine its prediction. We investigate the task of predicting Ames mutagenic activity of chemical compounds. Not being mutagenic (i.e. not able to cause mutations in the DNA) is an impor-

tant requirement for compounds under investigation in drug discovery and design. The Ames test [179] is a standard experimental setup for measuring mutagenicity. The following experiments are based on a set of Ames test results for 6512 chemical compounds that we published previously.<sup>14</sup>

GPC was applied as detailed in the following:

- Class 0 consists of non-mutagenic compounds
- Class 1 consists of mutagenic compounds
- Randomly split 6512 data points into 2000 training and 4512 test examples such that:
  - The training set consists of equally many class 0 and class 1 examples.
  - For the steroid compound class the balance in the train and test set is enforced.
- 10 additional random splits were investigated individually. This confirmed the results presented below.
- Each example (chemical compound) is represented by a vector of counts of 143 molecular substructures calculated using the DRAGON software [205].
- Normalize training and test set using the mean and variance of the training set.
- Apply GPC model with RBF kernel
- Performance (84 % area under curve) confirms our previous results [5]. Error rates can be obtained from Figure 5.15.

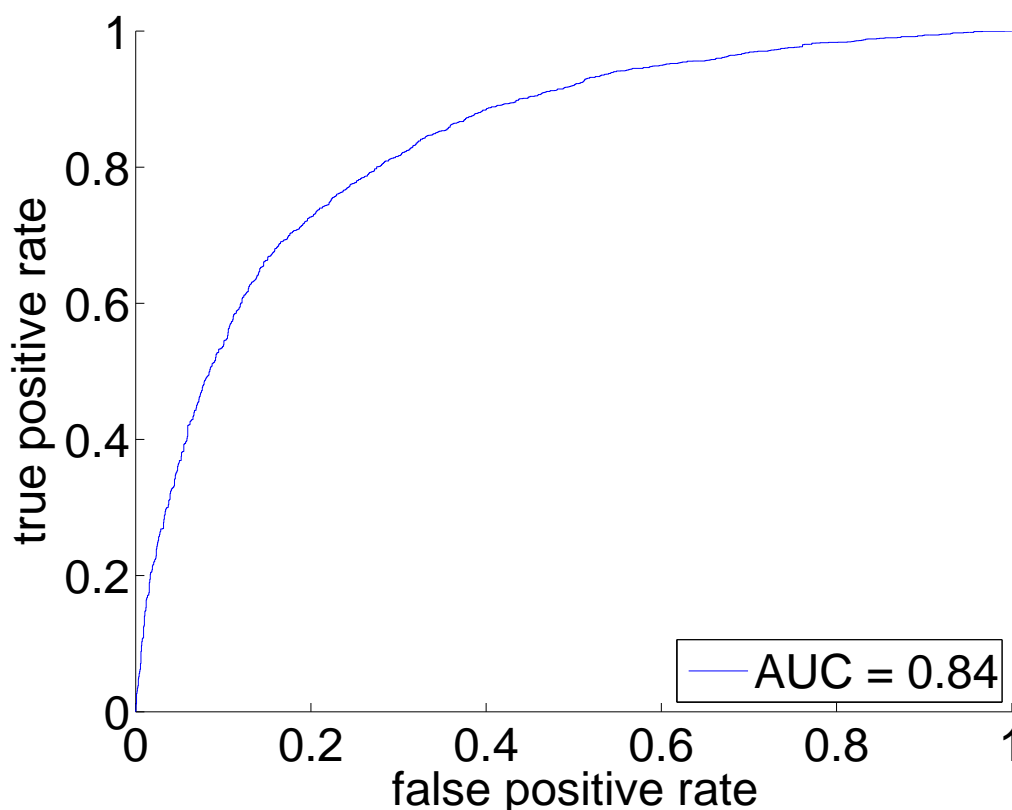
Together with the prediction we calculated the explanation vector (as introduced in section 4.4.2 with Definition 4.2) for each test point. The remainder of this section is an evaluation of these local explanations.

In Figures 5.16 and 5.17 we show the distribution of the local importance of selected features across the test set: For each input feature we generate a histogram of local importance values, as indicated by its corresponding entry in the explanation vector of each of the 4512 test compounds. As a common visual reference, the normal distribution with a standard deviation of 0.02

---

<sup>14</sup>See [5] for results of modeling this set using different machine learning methods. The data itself is available online at <http://ml.cs.tu-berlin.de/toxbenchmark>





**Figure 5.15:** Receiver operating curve of GPC model for mutagenicity prediction

(arbitrary choice) is included in each histogram. Each subfigure contains two measures of (dis-)similarity for each pair of distributions. The p-value of the Kolmogorov-Smirnoff test (KS) gives the probability of error when rejecting the hypothesis that both relative frequencies are drawn from the same underlying distribution. The symmetrized Kullback-Leibler divergence (KLD) gives a metric of the distance between the two distributions.<sup>15</sup>

The features examined in Figure 5.16 are counts of substructures known to cause mutagenicity. We show all approved “specific toxicophores” introduced by [207] that are also represented in the DRAGON set of features.

The features shown in Figure 5.17 are known to detoxify certain toxicophores [again see 207]. With the exception of (e) the toxicophores also have a toxifying influence according to our GPC prediction model. Feature (e) seems to be mostly irrelevant for the prediction of the GPC model on the

<sup>15</sup>Symmetry is achieved by averaging the two Kullback-Leibler divergences:  $\frac{KL(P1,P2)+KL(P2,P1)}{2}$ , cf. [206]. To prevent zero-values in the histograms which would lead to infinite KL distances, an  $\varepsilon > 0$  has been added to each bin count.

test points. In contrast the detoxicophores show overall negative influence on the prediction outcome of the GPC model. Modifying the test compounds by adding toxicophores will increase the probability of being mutagenic as predicted by the GPC model while adding detoxicophores will decrease this predicted probability.

So we have seen that the conclusions drawn from our explanation vectors agree with established knowledge about toxicophores and detoxicophores. While this is reassuring, such a sanity check required existing knowledge about which compounds are toxicophores and detoxicophores and which are not. Thus it is interesting to ask, whether we also could have *discovered* that knowledge from the explanation vectors. To answer this question we ranked all 143 features by the means of their local gradients<sup>16</sup>. Clear trends result: 9 out of 10 known toxicophores can be found close the top of the list (mean rank of 19). The only exception (rank 81) is the aromatic nitrosamine feature.<sup>17</sup> This trend is even stronger for the detoxicophores: The mean rank of these five features is 138 (out of 143), i.e. they consistently exhibit the largest negative local gradients. Consequently, the established knowledge about toxicophores and detoxicophores could indeed have been *discovered* using our methodology.

In the following paragraph we will discuss steroids<sup>18</sup> 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.

Figure 5.18 displays the difference in relevance of epoxide (a) and aliphatic nitrosamine (c) substructures for the predicted mutagenicity of steroids and non-steroids. For comparison we also show the distributions for compounds chosen at random from the test set (b,d). Again the p-value of the KS test and the symmetric KL divergence are used to measure the difference and distance in each pair of distributions. While containing epoxides generally

---

<sup>16</sup>Tables resulting from this ranking are made available as a supplement to [1] and can be downloaded from the journals website.

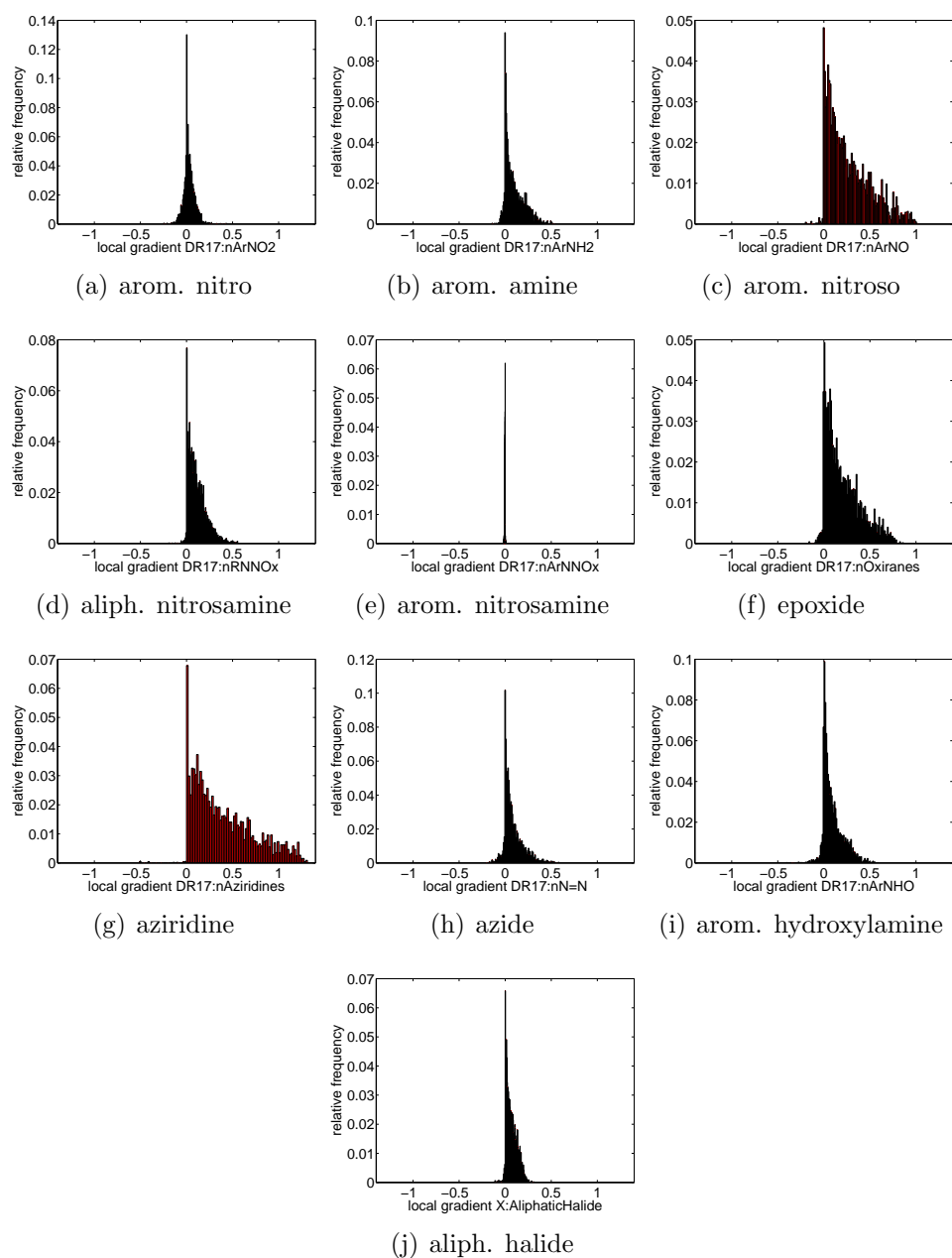
<sup>17</sup>This finding agrees with the result obtained by visually inspecting Figure 5.16(e). We found that only very few compounds with this feature are present in the dataset. Consequently, detection of this feature is only possible if enough of these few compounds are included in the training data. This was not the case in the random split used to produce the results presented above.

<sup>18</sup>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.

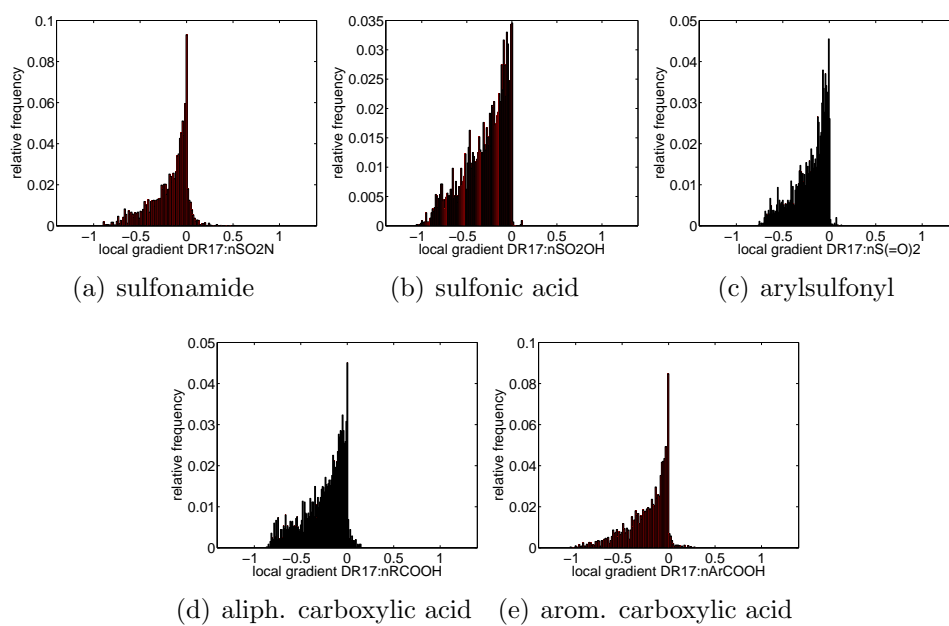
tends to make molecules mutagenic (see above), we do not observe this effect for steroids. “Immunity” of steroids to the epoxide toxicophore is an established fact and has first been discussed by [208]. This peculiarity in chemical space is clearly exhibited by the local explanation given by our approach. 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. To our knowledge, this phenomenon has not yet been discussed in the pharmaceutical literature.

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

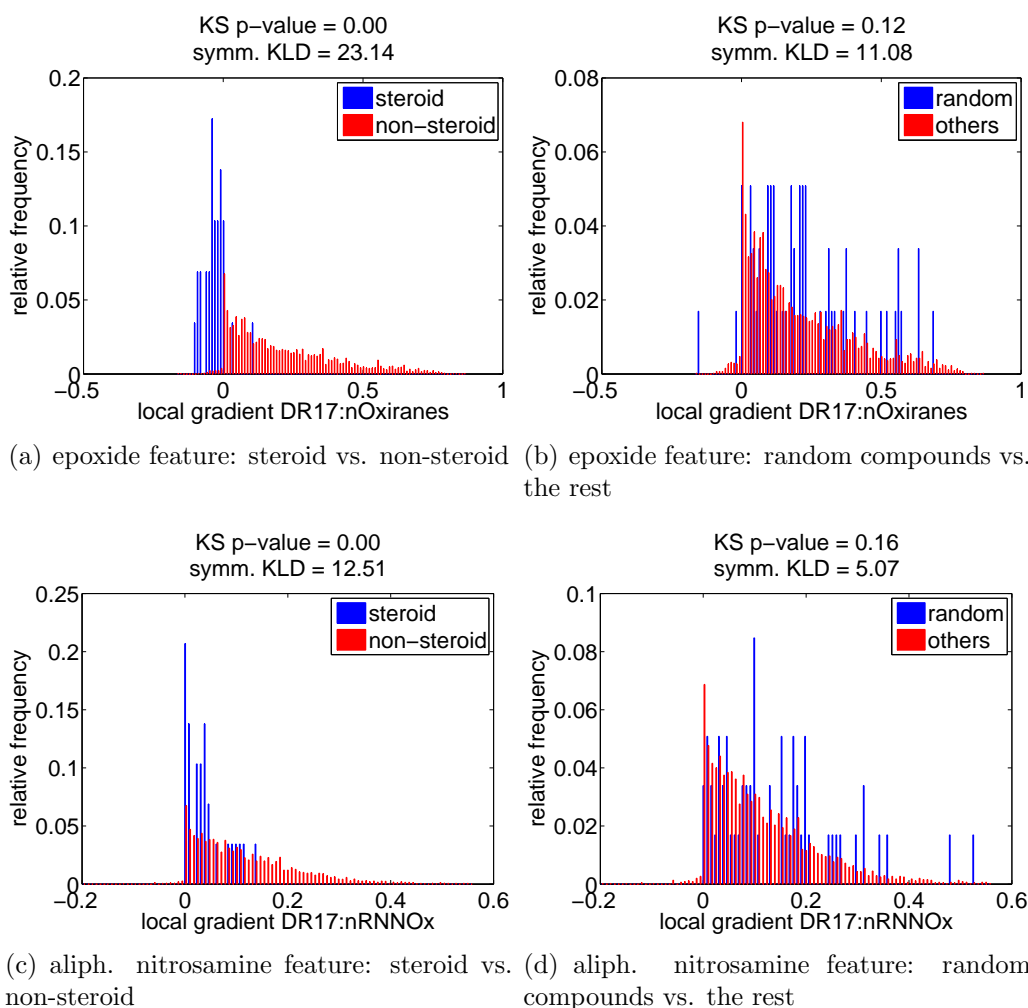
- toxicophores tend to make compounds mutagenic (class 1)
- detoxicophores tend to make compounds non-mutagenic (class 0)
- steroids are immune to the presence of some toxicophores (epoxide, aliphatic nitrosamine)



**Figure 5.16:** Distribution of local importance of selected features across the test set of 4512 compounds. Nine out of ten known toxicophores [207] indeed exhibit positive local gradients.



**Figure 5.17:** Distribution of local importance of selected features across the test set of 4512 compounds. All five known detoxicophores exhibit negative local gradients



**Figure 5.18:** The local distribution of feature importance to steroids and random non-steroid compounds significantly differs for two known toxicophores. 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 indeed exhibit positive local gradients.

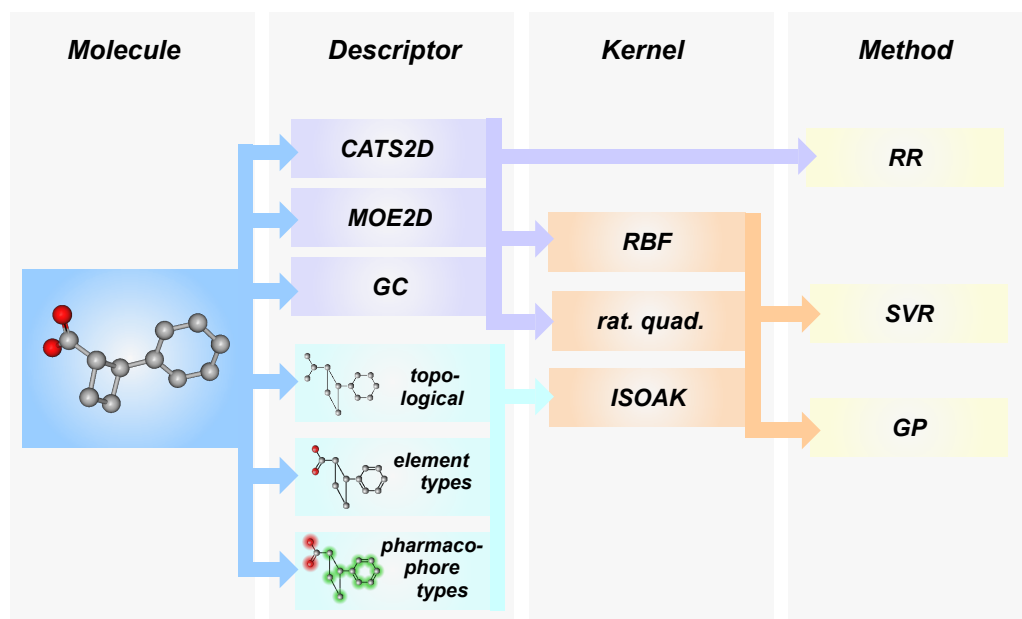
### 5.9 VIRTUAL SCREENING FOR PPAR $\gamma$ AGONISTS

The peroxisome-proliferator activated receptor family (PPAR) are nuclear receptors acting as transcription factors. They are involved in lipid metabolism and inflammatory response regulation and act as corresponding drug targets. The following section describes how predictive models for PPAR $\gamma$  were constructed in collaboration with researchers at the University of Frankfurt. It discusses how the training data was used in a clustered cross-validation scheme to evaluate the generalization capability of different combinations of data representations and learning algorithms. The last two paragraphs present how the best performing models were applied in screening two large vendor libraries: New PPAR $\gamma$  agonists were discovered, including a PPAR $\gamma$  selective compound with a new scaffold. This new compound is especially interesting, because it is almost identical to a natural product in plants that exhibit anti-diabetic and anti-inflammatory effects. The newly discovered PPAR $\gamma$  agonist represents a first hint to the cause of these effects. A more detailed discussion of the results of this virtual screening study can be found in [3].

From a machine learning perspective, the modeling task can be summarized as follows:

- ranking based on real valued labels, implemented as follows:
  - supervised regression task
  - ranking of unlabeled library by predicted target value & confidence
  - experimental investigation of 16 compounds from hitlist
- training data: 144 compounds with real valued labels (binding affinity expressed as  $pK_i$ ) [209], “leave 50 % of all clusters out” cross-validation
- unlabeled screening library:  $360 \cdot 10^3$  compounds from the Asinex[210] Gold and Platinum libraries<sup>19</sup>
- data representation: molecular graphs, CATS2D[211] topological pharmacophore descriptor (210-dimensional autocorrelation vector), MOE[212] 2D descriptors (184 diverse 2d descriptors), Ghose-Crippen fragment descriptors[164, 213, 214] (109 substructure counts)
- learning algorithms investigated: Gaussian Processes, support vector machines, ridge regression

<sup>19</sup>Asinex[210] Gold ( $233 \cdot 10^3$  compounds) and Platinum ( $129 \cdot 10^3$  compounds) libraries were combined, resulting in  $360 \cdot 10^3$  compounds after removal of duplicates and compounds not processable by intermediate other software.



**Figure 5.19:** Employed combinations of molecular descriptors, kernels (where applicable) and machine learning methods.

- kernel functions: ISOAK molecular graph kernel [106], radial basis function (RBF), rational quadratic (RQ) and combinations thereof (multiple kernel learning, see Sec. 3.6)

A flowchart illustrating the employed combinations of molecular descriptors, kernels (where applicable) and machine learning methods can be found in Figure 5.19. The retrospective part of the study was conducted as follows: All models were evaluated in 10 runs of “leave 50 % of all clusters out” cross-validation (see Sec. 3.4 for a discussion of evaluation strategies). The exact same splits into training and test data were used by all algorithms. Performance indicators for each algorithm were averaged over the 10 runs and are listed in tables 5.4, 5.5 and 5.6, together with the standard deviation over the 10 runs. The standard regression performance indicators were found to be highly correlated with each other ( $r > 0.95$ ), therefore we only list the mean absolute error alongside our newly defined  $FI_{20}$  performance indicator.<sup>20</sup>

Table 5.4 lists results achieved with three different learning algorithms each combined with three different vectorial descriptor sets. The non-linear Support Vector Regression and Gaussian Process models perform better than

<sup>20</sup> $FI_{20}$  is the fraction of inactive compounds in the top 20 of the ranking obtained with the respective model, see Sec. 3.5 for a discussion of performance indicators and loss functions)



Learning Algorithm	Descriptors	MAE	$FI_{20}$
SVR rbf	ACATS	<b>0.68</b> $\pm$ 0.06	0.33 $\pm$ 0.08
	GC	0.86 $\pm$ 0.12	0.41 $\pm$ 0.09
	MOE	0.69 $\pm$ 0.08	0.29 $\pm$ 0.14
Linear RR	ACATS	1.7 $\pm$ 0.14	0.80 $\pm$ 0.08
	GC	1.7 $\pm$ 0.08	0.79 $\pm$ 0.04
	MOE	1.45 $\pm$ 0.04	0.78 $\pm$ 0.05
GP RBF & RQ	ACATS	<b>0.66</b> $\pm$ 0.09	<b>0.27</b> $\pm$ 0.14
	GC	0.86 $\pm$ 0.07	0.33 $\pm$ 0.12
	MOE	0.76 $\pm$ 0.06	<b>0.25</b> $\pm$ 0.12

**Table 5.4:** For Support Vector Regression, Linear Ridge Regression and Gaussian Process regression based on ACATS, Ghose Crippen & MOE2D descriptors we list the Mean Absolute Error and the Fraction of Inactives in the top 20

Learning Algorithm	Node & Edge	MAE	$FI_{20}$
GP ISOAK	none	<b>0.68</b> $\pm$ 0.06	0.33 $\pm$ 0.15
GP ISOAK	Type	0.74 $\pm$ 0.06	<b>0.32</b> $\pm$ 0.14
GP ISOAK	PPP	0.70 $\pm$ 0.06	0.38 $\pm$ 0.09

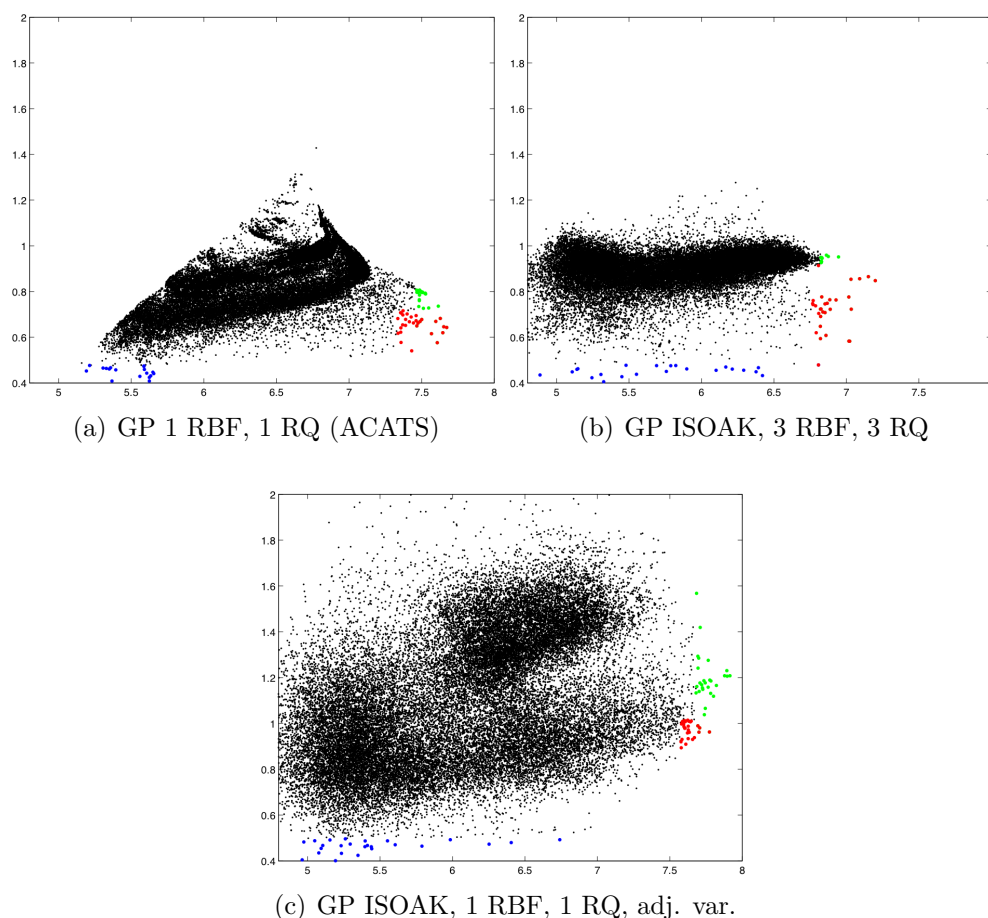
**Table 5.5:** Using atom & bond types (Type) or pharmacophore points (PPP) in addition to the plain graph structure does not improve the performance of ISOAK based Gaussian Process models. See Table 5.4 for abbreviations.

the linear Ridge Regression, indicating that the relationship between activity and the chosen descriptors is indeed non-linear. Non-linear models with the richer MOE2D and ACATS descriptor sets outperform models based on the rather simple Ghose Crippen fragment descriptors.

Table 5.5 lists performance indicators obtained when applying variants of the ISOAK graph kernel. Interestingly, comparing molecules based on the graph structure alone results in more accurate predictions than comparing molecules based on both their graph structure and either atom & bond types or pharmacophore points.

Learning Algorithm & Descriptors	MAE	$FI_{20}$
GP ISOAK, 1 RBF, 1 RQ	0,67 $\pm$ 0,08	0,31 $\pm$ 0,14
GP ISOAK, 1 RBF, 1 RQ, adj. var.	<b>0,66</b> $\pm$ 0,07	0,32 $\pm$ 0,15
GP ISOAK, 3 RBF, 3 RQ	0,70 $\pm$ 0,11	<b>0,21</b> $\pm$ 0,09
GP ISOAK, 3 RBF, 3 RQ, adj. var.	0,71 $\pm$ 0,12	0,26 $\pm$ 0,12

**Table 5.6:** The Gaussian Process Models based on both ISOAK graph kernels and RBF & RQ kernels with vectorial descriptors yield the lowest MAE and  $FI_{20}$  values observed so far. See Table 5.4 for abbreviations.



**Figure 5.20:** Plots of predictive variance vs. predicted binding affinity, obtained by applying the three chosen models to the screening database. Both values were fused into a single list by subtracting the predictive standard deviation from each prediction and sorting by the result, so that compounds with highly confident high predictions can be found at the top of the list[215]. The top 30 compounds suggested by each model are marked with red dots in each plot. They differ from the compounds with highest confidence in the prediction (blue dots) and the compounds with the highest predicted binding affinity (green dots).

Statistics for the most complex models can be found in Table 5.6: These Gaussian Process models were built using both the molecular structure and vectorial descriptors at the same time, by combining ISOAK graph kernels and RBF & RQ kernels in a multiple kernel learning (MKL) setting. Giving the compounds with higher activity more weight in hyperparameter optimization does not have a visible effect on the performance. Combining all vectorial features into one RBF & RQ kernel results in slightly reduced mean

absolute errors. However, using individual RBF & RQ kernels for the three individual vectorial descriptor sets we obtained the lowest values for the  $FI_{20}$  performance indicator. When grouping models based on their performance, three groups can be identified:

- All linear models were found to be very inaccurate ( $MAE > 1.45$ ,  $FI_{20} > 0.78$ ).
- Both non-linear models based on GC descriptors reach the same MAE of  $0.86 \pm 0.10$ . The SVR model exhibits an  $FI_{20}$  of  $0.41 \pm 0.09$ . Incorporating the confidence estimates of the Gaussian process model results in a slightly reduced  $FI_{20}$  of  $0.33 \pm 0.12$ , despite equal regression performance.
- Non-linear models based on the vectorial MOE or ACATS descriptors, molecular structures or combinations thereof all achieve mean MAEs between 0.66 and 0.74. All eleven models reach  $FI_{20}$  below 0.40. Five models even reach  $FI_{20}$  between 0.2 and 0.3.

While the differences between the groups are large, differences within the best performing group are as small as the standard deviations across the ten runs. Instead of choosing a single final model for the prospective evaluation based on these results alone, we first chose a group of three very well performing models with different characteristics:

- The GP model with an ISOAK graph kernel and 3 individual RBF & RQ kernels for each respective vectorial descriptor set was chosen because it exhibits the lowest  $FI_{20}$  score ( $0.21 \pm 0.09$ ) observed in all experiments, combined with a rather low MAE of  $0.70 \pm 0.11$ . At the same time it includes all available information about each compound.
- A GP model with an ISOAK graph kernel and one single RBF & RQ kernel for all vectorial descriptor sets taken together was chosen because it exhibits the lowest MAE ( $0.66 \pm 0.07$ ) observed in all experiments, combined with a reasonable  $FI_{20}$  score of  $0.32 \pm 0.15$ . It also includes all available information about each compound and gives the active compounds higher weight in the parameter optimization.
- From the pool of still well performing but less complex models we chose the GP model employing just one RBF & RQ kernel on a single vectorial descriptor set (ACATS), achieving MAE  $0.66 \pm 0.09$  and  $FI_{20}$   $0.27 \pm 0.14$ .

The ASINEX database was then ranked with each of the three models listed above. The predictive standard deviation was subtracted from each prediction, so that compounds with highly confident high predictions can be found at the top of the list [215]. The top 30 compounds are marked with red dots in the confidence vs. target value plots for each model in Figure 5.20. From these three lists, 16 compounds were selected by a panel of human experts and experimentally tested, resulting in 8 novel PPAR $\alpha$  and PPAR $\gamma$  agonists with EC<sub>50</sub> values as low as 9  $\mu mol$ . Interestingly, 10 out of 16 cherrypicked compounds were suggested by this model: “GP with ISOAK and 3 RBF and RQ kernels on the vectorial descriptors” (chosen for best  $FI_{20}$  score), and all 8 compounds exhibiting activity stem from this list.

The most potent compound is a known natural product in plants: Bermuda grass (*cynodon dactylon*) is known to be anti-inflammatory, for which the PPAR $\gamma$  activity of our compound provides a first possible explanation. A detailed discussion of this finding can be found in [3].

## CHAPTER 6

# CONCLUSION

This thesis presents a collection of seven studies about constructing predictive models for application in drug discovery & drug design. Gaussian Process models were *introduced* into the field of chemoinformatics and for the first time, individual confidence estimates are provided based on a solid theoretical foundation. Furthermore, *new algorithms* were developed to cope with the specific requirements of lead optimization, the most challenging part of the drug discovery process. The first new algorithm can *improve* the *accuracy* of models in the early stages of lead optimization projects. The second new algorithm can *explain individual predictions* made by complex models to human experts and the third new algorithm generates *hints for compound optimization*. Models developed in the above mentioned studies have been deployed for use by human experts in pharmaceutical companies. A virtual screening study has not only led to new agonists of PPAR $\gamma$ , but also helped to understand the anti-diabetic and anti-inflammatory effects of bermuda grass (cynodon dactylon).

From the point of view of machine learning, chemoinformatics is a very challenging field of endeavor:

Learning algorithms are always based on some representation of objects (in this case: molecules). Molecules are dynamical three dimensional objects, exhibiting many different types of flexibility. Available representations of molecules for machine learning either completely ignore this fact by considering only features derived from the two dimensional graph of the molecule, or they consider a small arbitrarily chosen number of 3D structures that may or may not be relevant for the task at hand. Consequently, the accuracy that can be achieved by machine learning models based on these representations is limited.

Machine learning algorithms rely on the assumption that training data and future test data are sampled ideally identically distributed (i.i.d.) from the same underlying probability density, and further assume that the conditional distribution of labels (measurements) given the input features (descriptors) is the same in both training and test data. Lastly, it is generally assumed that similar molecules exhibit similar activity.

In the machine learning community, violation of the first assumption is called *covariate shift* or *dataset shift*. In the lead optimization application scenario, one question that regularly arises is how to best use the first new measurements for compounds belonging to a newly explored compound class, i.e. a new part of the chemical space. New different model selection and bias correction algorithms based on utilizing the labels of the nearest neighbors in the recently measured set of compounds are introduced. An evaluation of these algorithms in the context of the hERG Channel Blockade effect reveals that a locally performed bias correction significantly improves models. Interestingly, previously trained random forests profit so strongly from the bias correction, that they even outperform k-nearest-neighbor models trained in leave-one-out cross validation on all training and test data simultaneously.

Four out of seven modeling studies conducted in this thesis deal with properties that have *multiple underlying mechanisms* that are relevant in different parts of the chemical space. These properties are Metabolic Stability, Aqueous Solubility, Cytochrome P450 Inhibition and Ames Mutagenicity. Training data are typically scarce (e.g. several hundred training compounds described by about equally many descriptor dimensions (features)). In new projects, new compound classes are explored, i.e. test data are sampled from a different part of the chemical space than the training data. Because of the different underlying mechanisms, this leads to simultaneously *violating* both the *assumption of i.i.d. sampling* and the *assumption of equal conditionals*. Furthermore, all properties concerned with molecular recognition can exhibit sudden extreme changes (*activity cliffs*). This necessitates complex non-linear models and makes generalizing difficult. The six out of seven studies conducted in this thesis where activity cliffs need to be considered are: Metabolic Stability, Cytochrome P450 Inhibition, Ames Mutagenicity, Aqueous Solubility, PPAR $\gamma$  binding and the hERG Channel Blockade Effect.

As indicated in the last paragraph, one has to expect that many predictions for test compounds may be incorrect. It is therefore very desirable to obtain some kind of confidence estimate that allows to identify compounds for which reliable predictions can be made. To this end, Gaussian Process (GP) models have been *introduced* into the field of chemoinformatics. Their predictive variances can directly serve as individual confidence estimates. The practical usefulness of predictive variances is investigated in detail for models predicting Partition Coefficients, Aqueous Solubility and Metabolic Stability. In each case it is shown that focussing on confidently predicted compounds indeed significantly increases the accuracy of predictions. In a virtual screening for compounds binding to the PPAR $\gamma$  receptor, predictive variances of GPs were considered to select compounds with high values of predicted binding affinity that are, at the same time, deemed to be very

reliable. And indeed, new potent agonists of PPAR $\gamma$  were identified.

In addition to the above mentioned *fundamental* aspects that make chemoinformatics challenging from the point of view of machine learning, there are many more points to be considered when not only treating well curated benchmark datasets, but really starting the process of model building from raw datasets that have just been exported out of corporate databases or gathered from (many) separate publications. Systematical measurement errors, heteroschedastic noise, compounds with outlying labels or features or even missing values in some features necessitate carefully performing various pre-processing steps before training machine learning models. Since different raw sets of data can vary tremendously, the process of pre-processing is not easily formalized. The first three sections of this thesis describe insights gained when transforming raw data into well-behaved training data for machine learning models.

Two separate algorithms for explaining individual predictions of (possibly non-linear) machine learning models are presented. The first method *explains* predictions by the means of visualizing relevant objects from the training set of the model. This allows human experts to understand how each prediction comes about. If a prediction conflicts with his intuition, the human expert can easily find out whether the grounds for the models predictions are solid or if trusting his own intuition is the better idea. Theoretically, a brute force algorithm can be applied to gather this information about any retrainable model, regardless of the underlying learning algorithm. As shown in this thesis, a much more elegant solution exists for kernel based learning methods. When the representer theorem can be applied (i.e. in the case of support vector machines for classification (SVM) and regression (SVR), Gaussian Processes for classification (GPC) and regression (GPR) and kernel ridge regression (KRR)) one can calculate the normalized contribution of each training data point *analytically*, and identify the most important points to visualize. These are called *explanations*. GPC models trained using standard procedures were found to tend to rely on more training compounds than one can use for producing an intuitively understandable visualization. By reducing the width-parameter of radial basis function kernels, it is possible to obtain models where predictions for test compounds are almost completely determined by very few training compounds, so that intuitively understandable visualizations can indeed be produced. Furthermore, these *explanations* were found to be convincing from a chemists point of view.

In a somewhat similar spirit, a second method utilizes local gradients of the model's predictions to explain predictions in terms of the *locally most relevant features*. This not only teaches the human expert which features are relevant for each individual prediction, but also gives a directional infor-

mation. Abstractly speaking, one can learn in which direction a data point has to be moved to increase the prediction for the label (measurement). In the context of lead optimization, this means that the human expert can obtain *hints for compound optimization*. An estimation framework allows to obtain local gradients from any type of model, including k-nearest neighbor classifiers, support vector machines, decision trees or neural networks. However, in case of Gaussian Process models, local gradients can be calculated analytically. The new gradient based methodology was validated using two benchmark sets of data that are well known in the machine learning community (IRIS flowers and USPS digits) and was then applied to Gaussian Process Classification models for Ames mutagenicity. The new approach correctly identified toxicophores and detoxicophores. Furthermore, even local peculiarities in chemical space (the extraordinary behavior of steroids) was discovered based on the local gradients of the predictions.

#### FUTURE DIRECTIONS

Today, machine learning based models are firmly established in the early stages of the drug discovery process, i.e. in library design. In these early stages, large numbers of compounds are ranked or filtered with respect to different properties and any model that can make predictions that are significantly better than random choice has the chance to exert a positive effect, even if the effect is not very large. In later stages of the drug discovery process, more specifically in the lead optimization phase, requirements are much higher: Human experts (often bench chemists) regularly decide which small batch of molecules to synthesize and investigate next. They tend to be very sceptical towards model predictions unless they understand exactly how each prediction is generated. This behavior also makes sense, because it is the researchers themselves who have to make the decision and bear the responsibility. Algorithms capable of *explaining predictions* will be the *key to* reaching the goal of much increased *acceptance* of models. Once models are accepted by the relevant decision makers in lead optimization, technologies for *guiding compound optimization* have the chance to become valuable.

While the lead optimization scenario served as the original motivation and testbed for developing algorithms for *explaining* individual predictions, both new methods can be applied to a wide range of modeling tasks. Wherever human experts are to be supported in making decisions, explanations of predictions will be valuable. Therefore, future research will strive for developing and refining algorithms for *explaining predictions* by machine learning models and demonstrating their practical usefulness.



## APPENDIX A

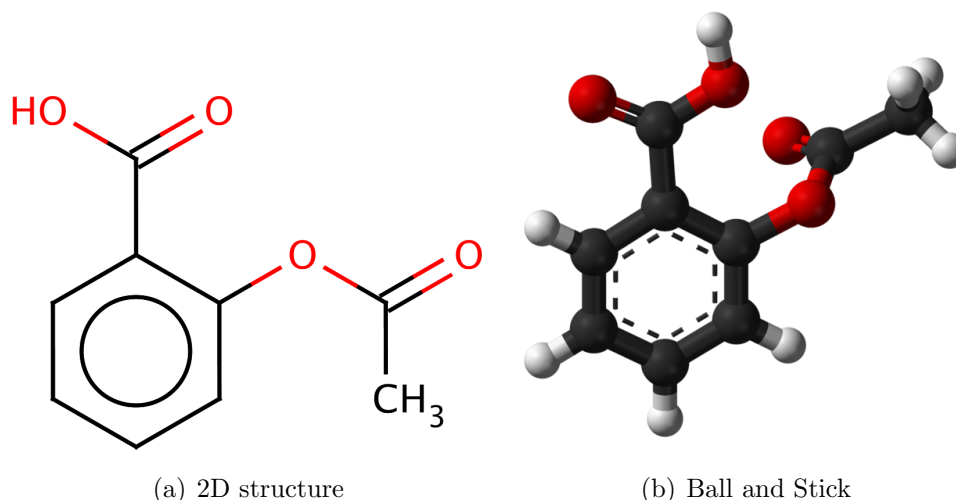
### MISC

#### A.1 CHALLENGING ASPECTS OF MACHINE LEARNING IN DRUG DISCOVERY

##### A.1.1 *Molecular Representations*

The by far most critical aspect to consider when modeling the properties of chemical compounds is the fact that state of the art representations of molecules are seriously flawed. Chemical compounds are commonly characterized by the 2D structure (graph) of their molecules. As an example, the 2D-structure and a ball and stick model of acetylsalicylic acid (also known as Aspirin) can be found in Figure A.1. This representation has been developed for use by human experts and is very useful in tasks like developing synthesis paths, because 2D structures can easily be drawn by hand and printed in books and they can be used to express chemical reactions or whole series of reactions that one can learn to read, write, memorize and apply in the laboratory. In the 2D-structure Figure A.1(a), each type of edge indicates a different type of chemical bond: A single line symbolizes a single bond, two parallel lines indicate a double bond and the circle drawn inside the six membered ring indicates aromatic bonds. Note how the ensemble of atoms symbolized in the right hand side of subfigure (a) appears in a different position in subfigure (b). The position of this group of atoms (the acetyl group) with respect to the remaining part of the molecule is *arbitrary*, because it is connected through a single bond, and single bonds *rotate freely*. Furthermore, bonds can *vibrate* in different ways, including stretching and bending. When dissolved in water, the acid group depicted in the top left hand corner of Figure A.1(a) can *dissociate* its proton ( $H^+$  Ion) and later *associate* with any proton nearby. In more complex molecules, more complex issues arise: Protons can bind to different functional groups within the same molecule, resulting in shift of the positions of alternatingly placed double and single bonds. This often observed phenomenon is called *tautomerism*. In summary, molecules can be flexible both in a physical sense (rotation and vibration) and a chemical sense (dissociation & association of protons, tautomerism).

As a result of this flexibility, molecules adopt many different states when

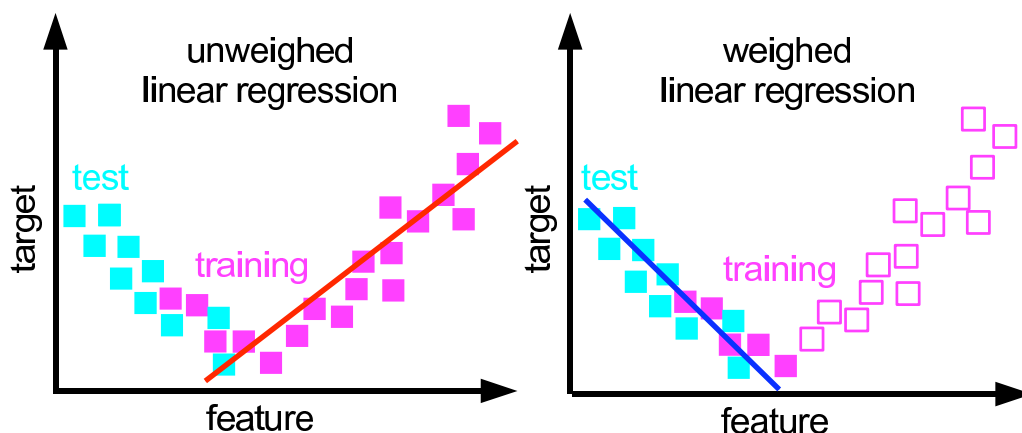


**Figure A.1:** 2D-structure and a ball and stick model of acetylsalicylic acid, also known as Aspirin.

observed in (aqueous) solution. Some states are more stable than others and therefore contribute more to the time averaged distribution across all possible states. Therefore, one could think of developing a molecular representation based on this whole distribution of possible states and their relative probability of occurrence. However, when binding to proteins, molecules can adapt to the shape of the respective protein. In the most stable complex of a protein and a molecule, the molecule's state may be one that would not have been stable in solution and is unique to this very complex.

The most popular way of representing molecules for the purpose of machine learning is choosing one out of many available tools to calculate a vector of so called chemical descriptors from the 2D structure of the molecule, and then apply standard learning algorithms to the resulting vectorial descriptors (features). In this way, the flexible structure of molecules is completely ignored. In a similar spirit, Kernel based machine learning methods like support vector machines can directly use the 2D structure representation of molecules if so called graph kernels are applied [106]. Again, the flexible structure of molecules is ignored.

Some descriptor calculation tools also take the 3D structure of molecules into account. As established above, the choice of any finite numbers of 3D structures is arbitrary and one always runs the risk of overlooking the most relevant ones for the respective modeling task. This may be the reason for the fact that models based on 3D representations of molecules do not necessarily outperform models based on 2D representations.



**Figure A.2:** In both plots, the training and test data have different but overlapping distributions. This scenario is known as “covariate shift” and can lead to models that will not generalize well (left plot). If the density of the test data can be estimated, the sample bias can be reduced or even eliminated using different weighing and sampling schemes [216]. This leads to models that perform better on the test data (right plot).

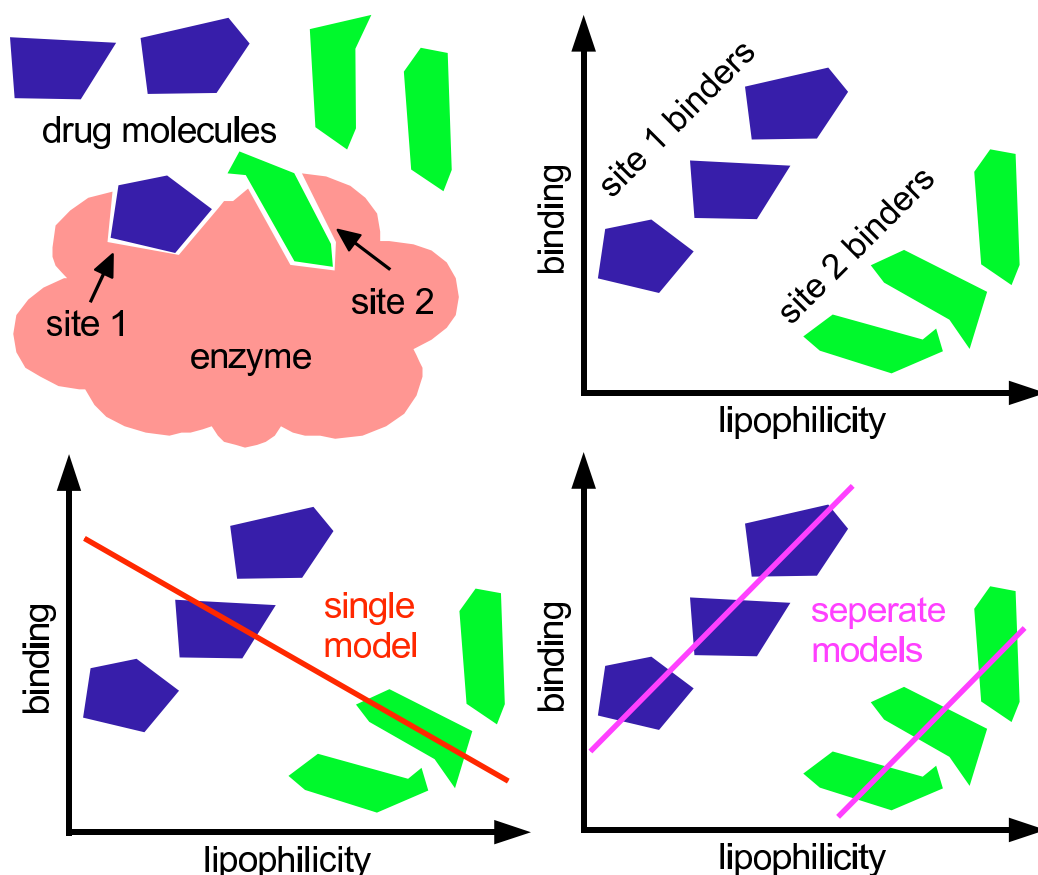
### A.1.2 Covariate Shift

This section introduces the phenomena “covariate shift” and “multiple modes of action”, discusses ways of handling these situations and points to parts of this thesis where either of the phenomena were observed.

Almost all known machine learning algorithms rely on the assumption that both training data and future test data are sampled from the same underlying probability density. If this assumption is not satisfied, we extrapolate from the training data. (Mild) extrapolation is feasible if the conditional distribution of target values given the input features (descriptors) is the same in both test and training data. In the machine learning community, this scenario is known as *covariate shift* [216–220]. One of the symptoms is a bias in the model selection and fitting phase. This bias can result in models that do not perform well on the test data. If the density of the test data can be estimated at the point in time when the model is trained, for some types of models this bias can be reduced or even eliminated using different weighing and sampling schemes. This is illustrated in Figure A.2, using a one dimensional example.<sup>1</sup> For a proof that unbiased estimators can indeed be obtained, see [216]. For a detailed discussion, reading [220] and references therein is suggested.

In drug discovery and development, covariate shift is omnipresent, due

<sup>1</sup>Keep in mind that the above mentioned scheme reduces the bias of a not complex enough model by effectively discarding training data.



**Figure A.3:** Top left: The enzyme displayed has two separate binding sites for small molecules. Top right: There exists a general trend for proteins to bind molecules more strongly as they get more lipophilic. Bottom left: Building a single simple model based on the valuable descriptor “lipophilicity” results in misleading models. Bottom right: If sufficient previous knowledge exists about both test and training data, one can construct separate models for the site 1 and site 2 binding molecules. Alternatively one can use a more rich feature set with a complex enough learning algorithm.

to the small training sets resulting from expensive experiments, but huge libraries for which predictions are sought. The term “covariate shift”, however, is typically not used. It is not only relatively new and originated in the machine learning community [216], but also because the situation is, in fact, often even worse: As explained above, covariate shift can help select better models in case training data and future test data are *not* sampled from the same underlying probability density, provided that the conditional distribution of target values given the input features (descriptors) is the same in both test and training data. As explained in the next subsection, the latter assumption is unfortunately often not satisfied in drug discovery

applications.

### A.1.3 Multiple Mechanisms

Figure A.3 illustrates this fact for the problem of protein binding. In this case, the protein under consideration is an enzyme, to which small molecules (including drugs) can bind. Our enzyme has two geometrically quite different active sites. Therefore, quite different molecules will bind to these different sites (Figure A.3, top left). Proteins are, generally speaking, quite lipophilic. Consequently, the more lipophilic a molecule, the stronger it binds to a given protein. In Figure A.3 (top right) a plot of binding strength vs. lipophilicity is shown. Inside each group (site 1 binders and site 2 binders) the binding strength indeed increases as the lipophilicity increases. If we now train a simple linear regression model using all data together (Figure A.3, bottom left), we obtain a bad fitting model that points in the completely wrong direction. This phenomenon is not restricted to enzymes with multiple binding sites. Some enzymes only have one active site, but if this single site is large and flexible, different types of molecules can dock into this site in different orientations or binding modes. In the chemoinformatics community, this problem is referred to as *multiple mechanisms*. Possible ways of obtaining useful models are:

1. construct individual models for each site or mode of action, as illustrated in Figure A.3 (bottom right)
2. use a rich feature set and advanced learning algorithms to implicitly construct local models for each mode of action by building a single sufficiently complex non-linear model

The first option not only requires the knowledge which training molecule binds into which site etc., but also leads to a number of different models. Ideally, one would have to know the relevant binding site or mode of action for each molecule in the test set to apply the adequate model from this set of models. This type of knowledge is usually not available.

Building a single model as illustrated in Figure A.3 (bottom left) only fails because this model has a one dimensional feature set and it is not sufficiently complex. Imagine a rich feature set, corresponding to a very high dimensional space. If any of the features are relevant for the problem at hand, the molecules with different modes of action will automatically end up in different regions of this space. If one now uses a learning algorithm that can take local phenomena into account (e.g. a Support Vector Machine or Gaussian process with a sufficiently local kernel function) one can succeed in

training a single model that will perform well despite the different modes of action within the same set of training (and test) data. Naturally, this model will not be able to make reasonable predictions for the whole vast space of feasible molecules. It is therefore important that models can provide some sort of confidence estimate, that allows assessing whether a given molecule is inside the domain of applicability, see Sec. 3.7.

#### A.1.4 Activity Cliffs

Machine learning in drug discovery and design relies on the assumption that similar molecules exhibit similar activity. In other words, one hopes that the “activity landscape” is somewhat smooth. However, very similar molecules may in some cases possess very different activities leading to what can be called *activity cliffs*. Maggiora [221] defines these by the ratio of the difference in activity of two compounds to their distance of separation in a given chemical space. The existence of such activity cliffs is not entirely surprising since molecular recognition plays a crucial role in determining activity.

Figure A.3 shows a simplified example where only the shape of molecules is relevant. In reality, a small number of functional groups (e.g. hydrogen donor or acceptor groups) in the molecule intensively interacts with their respective counterpart in specific spots in the binding pocket. The number functional groups in the molecule may be large, but only very few of them ([1...10]) participate in very important interactions. A slight change in the molecular structure may, in general, only cause a small difference in activity, but when one of the key interactions is involved, activity may change completely [221, 222].

Isoenzyme	weak	moderate	potent	reported	total
CYP 1A2	577	89	28	104	802
CYP 2C19	225	298	170	56	772
CYP 2C9	179	286	236	86	828
CYP 2D6	497	88	52	105	784
CYP 3A4	318	281	107	122	887

**Table A.1:** Number of compounds that are weak, moderate or potent inhibitors of each CYP isoenzyme.

Isoenzyme	Schering in-house	[223]	GT	[224]	other
CYP 1A2	720	124	67	7	41
CYP 2C19	718	72	55	13	38
CYP 2C9	719	122	97	16	83
CYP 2D6	718	152	119	19	129
CYP 3A4	711	188	129	25	98
total	3586	658	467	80	389

**Table A.2:** Number of CYP inhibition data that could be retrieved from the individual data sources, [223, 224] and in-house data.

## A.2 DATA USED IN MODELING CYTOCHROME P450 INHIBITION

We consider a total 5180 data points on CYP inhibition of 1130 different compounds. The data stems from two different sets:

1. A set of data collected from various literature sources [223, 224].
2. A large homogeneous set of quantitative data produced under standardized experimental conditions (in-house data at Schering).

Compounds in the in-house set of data were classified as weak ( $IC_{50} > 10\mu M$ ), moderate ( $1\mu M < IC_{50} < 10\mu M$ ) or potent inhibitors ( $IC_{50} < 1\mu M$ ) for each of the five CYP isoenzymes. If no quantitative data was available, but the structure was reported in the literature as an inhibitor of the specific isoenzyme, this structure is classified as “reported”. It is likely that structures with the label “reported” are moderate or potent inhibitors of the respective CYP isoenzyme. These qualitative data stem from various literature sources and have a higher diversity of CYP inhibitor structural classes.

### A.3 DESCRIPTORS USED IN MODELLING CYTOCHROME P450 INHIBITION

Cytochrome P450 Inhibition is a highly complex process, it is thus not clear which chemical descriptor can best capture the properties of a chemical compound that are indicative of its inhibition behavior. Therefore, we consider a total of six different sets of descriptors. These range from conceptually simple and computationally cheap pseudo atom counts (Ghose-Crippen descriptor) to complex and computationally more demanding 3 D descriptors (VOLSURF). In the following section, each descriptor is explained shortly and references to the original publications are provided.

The Ghose-Crippen descriptor comprises the numbers of occurrence of 120 different pseudo-atoms. It was originally developed to predict the octanol-water partition coefficient [213].  $P(\text{octanol/water})$  is the ratio of the concentration of a chemical in octanol and in water at equilibrium and at a specified temperature. Octanol is an organic solvent that is used as a surrogate for natural organic matter. G. & C. suggest that  $\log P$  is useful for treating ligand-receptor interaction. G. & C. also used these descriptors to predict molecular refractivity [214]. The set of descriptors was later extended to allow prediction of  $\log P$  for a wider range of compounds [164]. They argue that the dispersive force between molecules (or a molecule and a macro-molecule like a receptor) is related to their polarizability, which is proportional to their molar refractivity and suggest that this property is therefore useful for treating ligand-receptor interaction.

The BCUT descriptors are the highest and lowest eigenvalues of atomic connectivity matrices, where the diagonal elements can be set to the atoms mass, van der Waals volume, electronegativity or polarizability to reflect the atoms' ability to form bonds or otherwise interact [167]. They are an extension of the Burden eigenvalues [166]. In our application, we consider the 16 eigenvalues of largest magnitude and 16 eigenvalues of lowest magnitude.

UNITY fingerprints [168] (generated from the default screen2d definition file shipped with the software) are represented as string of 988 bits. They were developed to speed up database searches and work similar to hash functions. 928 bits are set to indicate the presence of paths of lengths 2 to 6 through the molecular graph. The remaining 60 bits indicate the presence of certain predefined fragments (e.g. rare atoms, or different ring systems).

The  $\log D$  prediction module from ACD Labs [75] is used to predict  $\log D$  for the ten pH values between 2 and 11 and the number of rotatable bonds. These eleven numbers are then used as descriptors.

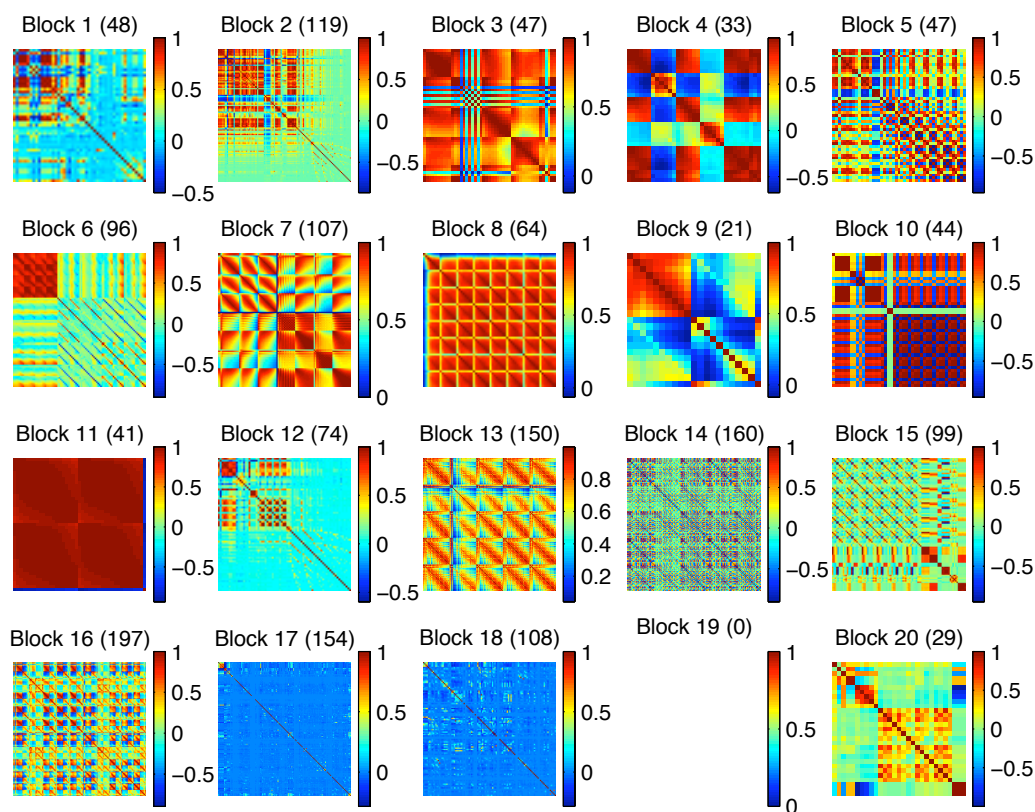
VolSurf descriptors [169] are generated in a three step process: A 3D



structure of the molecule is calculated using Concord [225]. Then different 3D molecular interaction fields [226] are created by virtually moving different hydrophilic or lipophilic probes around the molecule and recording distances of equal interaction energy. In our cases the H<sub>2</sub>O, DRY and O probes were used. These interaction fields are then used to calculate the actual VolSurf descriptors, including volume, surface, globularity, different descriptors of hydrophilic and lipophilic regions, interaction energy moment and others.

GRIND descriptors [170] are generated in a four step process. A 3D structure of the molecule is generated using Concord [225] and used to calculate a molecular interaction field (see above). In our case, the DRY, O, N1 and shape probes were used. A fixed number of nodes (e.g., 100) is chosen from the grid to maximize the value of a scoring function that depends on the sum of field intensities at all chosen points and the sum of distances between all chosen points. The ensemble of these regions for all relevant probes is called the virtual reaction site and forms the starting point for calculating the actual GRIND descriptors: For each possible pair of nodes the product of their field intensities is calculated. A discrete number of categories, each representing a small range of distances, is considered. For each category, the highest product of field intensities is stored. The parameters can be visualized by plotting the highest products vs. the distance. If pairs of nodes from the same molecular interaction field are considered, this is called auto-correlogram, if the pairs of nodes are from two different molecular interaction fields, this is called a cross-correlogram.

## A.4 MISCELLANEOUS PLOTS



**Figure A.4:** Blockwise correlation (Sec. 3.2) between Dragon descriptor dimensions of compounds used in modeling aqueous solubility (Sec. 5.3). A lot of descriptor dimensions are highly correlated, illustrated by the substructure inside each block.

## A.5 HOW TO CITE THIS THESIS

Please cite this document as follows:

Timon Schroeter. Machine Learning in Drug Discovery and Drug Design. PhD thesis, Machine Learning Dept., University of Technology Berlin, 2009.  
URL <http://ml.cs.tu-berlin.de/~timon>

Users of L<sup>A</sup>T<sub>E</sub>X are invited to use the following BibTeX record:

```
@PhdThesis{Schroeter_2009,  
author = {Timon Schroeter},  
title  = {Machine Learning in Drug Discovery and Drug Design},  
school = {Machine Learning Dept.,  
          University of Technology Berlin},  
year   = {2009},  
url    = {http://ml.cs.tu-berlin.de/~timon}  
}
```



## BIBLIOGRAPHY

- [1] 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*, *submitted*, 2009.
- [2] Timon Schroeter, Klaus-Robert Müller, and Katja Hansen. Method for explaining the predictions of a mathematical model to its users. *Patent Application at German Patent Office DPMA*, 2008.
- [3] Matthias Rupp, Timon Schroeter, Ramona Steri, Ewgenij Proschak, Katja Hansen, Oliver Rau, Manfred Schubert-Zsilavecz, Klaus-Robert Müller, and Gisbert Schneider. From virtual screening to natural products: New ppar $\gamma$  agonists. *ChemMedChem*, *submitted*, 2009.
- [4] Katja Hansen, Fabian Rathke, Timon Schroeter, Jan M. Kriegl, and Sebastian Mika. Bias-correction of regression models: A case study on herg inhibition. *Journal of Chemical Information and Modelling*, *in press*, 49(6):1486–1496, 2009. URL <http://dx.doi.org/10.1021/ci9000794>.
- [5] Katja Hansen, Sebastian Mika, Timon Schroeter, Andreas Sutter, Antonius Ter Laak, Thomas Steger-Hartmann, Nikolaus Heinrich, and Klaus-Robert Müller. A benchmark data set for in silico prediction of ames mutagenicity. *Journal of Chemical Information and Modelling*, *accepted*, 49(9):2077–2081, 2009. URL <http://dx.doi.org/10.1021/ci900161g>.
- [6] 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 Modeling*, 48:785–796, 2008. URL <http://dx.doi.org/10.1021/ci700142c>.
- [7] Timon Schroeter, Anton Schwaighofer, Sebastian Mika, Antonius Ter Laak, Detlev Suelzle, Ursula Ganzer, Nikolaus Heinrich, and Klaus-Robert Müller. Machine learning models for lipophilicity and their

- domain of applicability. *Mol. Pharm.*, 4(4):524–538, 2007. URL <http://dx.doi.org/10.1021/mp0700413>.
- [8] Timon Schroeter, Anton Schwaighofer, Sebastian Mika, Antonius Ter Laak, Detlev Suelzle, Ursula Ganzer, Nikolaus Heinrich, and Klaus-Robert Müller. Estimating the domain of applicability for machine learning qsar rmodels: A study on aqueous solubility of drug discovery molecules. *Journal of Computer Aided Molecular Design - special issue on "ADME and Physical Properties"*, 21(9):485–498, 2007. URL <http://dx.doi.org/10.1007/s10822-007-9125-z>.
- [9] Timon Schroeter, Anton Schwaighofer, Sebastian Mika, Antonius Ter Laak, Detlev Suelzle, Ursula Ganzer, Nikolaus Heinrich, and Klaus-Robert Müller. Estimating the domain of applicability for machine learning qsar rmodels: A study on aqueous solubility of drug discovery molecules. *Journal of Computer Aided Molecular Design - regular issue*, 21(12):651–664, 2007. URL <http://dx.doi.org/10.1007/s10822-007-9160-9>.
- [10] Anton Schwaighofer, Timon Schroeter, Sebastian Mika, and Gilles Blanchard. How wrong can we get? a review of machine learning approaches and error bars. *Combinatorial Chemistry & High Throughput Screening*, 12(5):453–468, 2009. URL <http://dx.doi.org/10.2174/138620709788489064>.
- [11] Anton Schwaighofer, Timon Schroeter, Sebastian Mika, Julian Laub, Antonius ter Laak, Detlev Sülzle, Ursula Ganzer, Nikolaus Heinrich, and Klaus-Robert Müller. Accurate solubility prediction with error bars for electrolytes: A machine learning approach. *Journal of Chemical Information and Modelling*, 47(2):407–424, 2007. URL <http://dx.doi.org/10.1021/ci600205g>.
- [12] Timon Schroeter, Anton Schwaighofer, Sebastian Mika, Antonius ter Laak, Detlev Sülzle, Ursula Ganzer, Nikolaus Heinrich, and Klaus-Robert Müller. Predicting lipophilicity of drug discovery molecules using gaussian process models. *ChemMedChem*, 2(9):1265–1267, 2007. URL <http://dx.doi.org/10.1002/cmdc.200700041>.
- [13] Peter Kroll, Timon Schröter, and Martina Peters. Prediction of novel phases of tantalum(v) nitride and tungsten(vi) nitride that can be synthesized under high pressure and high temperature. *Angew. Chem. Int. Ed.*, 44(27):4249–4254, 2005. URL <http://dx.doi.org/10.1002/anie.200462980>.

- [14] Peter Kroll, Timon Schröter, and Martina Peters. Synthesen bei hohem druck und hoher temperatur führen zu neuen phasen von tantal(v)-nitrid und wolfram(vi)-nitrid. *Angewandte Chemie*, 117(27):4321–4326, 2005. URL <http://doi.wiley.com/10.1002/ange.200462980>.
- [15] Timon Schroeter. Book review: Pathway analysis for drug discovery. *ChemMedChem*, 4(6):1020, 2009. URL <http://dx.doi.org/10.1002/cmdc.200900107>.
- [16] Novartis AG. Drug discovery and development process <http://www.novartis.com/research/drug-discovery.shtml>, 2009.
- [17] Gisbert Schneider and Karl-Heinz Baringhaus. *Molecular Design - Concepts and Applications*. Wiley-Vch, Weinheim, 2008.
- [18] Johann Gasteiger and Thomas Engel, editors. *Chemoinformatics: A Textbook*. Wiley-VCH, 2003.
- [19] Asa Ben-Hur, Cheng Soon Ong, Sören Sonnenburg, Bernhard Schölkopf, and Gunnar Rätsch. Support vector machines and kernels for computational biology. *PLoS Computational Biology*, 4, 2008. URL <http://svmcompbio.tuebingen.mpg.de>. (accepted).
- [20] G. Rätsch, S. Sonnenburg, and B. Schölkopf. RASE: recognition of alternatively spliced exons in c. elegans. *Bioinformatics*, 21:i369–i377, June 2005.
- [21] A. Zien, G. Rätsch, S. Mika, B. Schölkopf, T. Lengauer, and K.-R. Müller. Engineering support vector machine kernels that recognize translation initiation sites in DNA. *Bioinformatics*, 16(9):799–807, Sep 2000.
- [22] K. Grabowski, K.-H. Baringhaus, and G. Schneider. Scaffold diversity of natural-products: Inspiration for combinatorial library design. *Nat. Prod. Rep.*, *accepted.*, 2008.
- [23] R. Glen and G. (Eds) Schneider. Special journal issue: Challenges in virtual screening. *QSAR & Combinatorial Science*, 25(12), 2006.
- [24] Thomas Lengauer, Christian Lemmenb, Matthias Rareyc, and Marc Zimmermann. Novel technologies for virtual screening. *Drug Discovery Today*, 9(1):27–34, 2004. URL [http://dx.doi.org/10.1016/S1359-6446\(04\)02939-3](http://dx.doi.org/10.1016/S1359-6446(04)02939-3).

- [25] Martin Weisel, Ewgenij Proschak, Jan M. Kriegl, and Gisbert Schneider. Form follows function: Shape analysis of protein cavities for receptor-based drug design. *Proteomics*, 9(2):451–459, 2009. URL <http://dx.doi.org/10.1002/pmic.200800092>.
- [26] Gisbert Schneider and Uli Fechner. Computer-based de novo design of drug-like molecules. *Nat Rev Drug Discov*, 4(8):1474–1776, 2005. URL <http://dx.doi.org/10.1038/nrd1799>.
- [27] Markus Hartenfeller, Ewgenij Proschak, Andreas Schüller, and Gisbert Schneider. Concept of combinatorial de novo design of drug-like molecules by particle swarm optimization. *Chemical Biology & Drug Design*, 72(1):16–26, 2008. URL <http://dx.doi.org/10.1111/j.1747-0285.2008.00672.x>.
- [28] Andreas Schüller, Marcel Suhartono, Uli Fechner, Yusuf Tanrikulu, Sven Breitung, Ute Scheffer, Michael W. Göbel, and Gisbert Schneider. The concept of template-based de novo design from drug-derived molecular fragments and its application to tar rna. *J Comput Aided Mol Des*, 2008. URL <http://dx.doi.org/10.1007/s10822-007-9157-4>.
- [29] Gisbert Schneider, Markus Hartenfeller, Michael Reutlinger, Yusuf Tanrikulu, Ewgenij Proschak, and Petra Schneider. Voyages to the (un)known: adaptive design of bioactive compounds. *Trends in Biotechnology*, 27(1):18–26, 2009. URL <http://dx.doi.org/10.1016/j.tibtech.2008.09.005>.
- [30] Volker Hähnke, Bettina Hofmann, Tomislav Grgat, Ewgenij Proschak, Dieter Steinhilber, and Gisbert Schneider. Phast: Pharmacophore alignment search tool. *J. Comput. Chem.*, 30(5):761–771, 2008. URL <http://dx.doi.org/10.1002/jcc.21095>.
- [31] Steffen Renner, Mirko Hechenberger, Tobias Noeske, Alexander Bäcker, Claudia Jatzke, Michael Schmuker, Christopher G. Parsons, Tanja Weil, and Gisbert Schneider. Searching for drug scaffolds with 3d pharmacophores and neural network ensembles. *Angew. Chem. Int. Ed.*, 46: 5336–5339, 2007.
- [32] K. Grabowski, E. Proschak, K.-H. Baringhaus, O. Rau, M. Schubert-Zsilavecz, and G. Schneider. Bioisosteric replacement of molecular scaffolds: From natural products to synthetic compounds. *Nat. Prod. Commun.*, 3:1355–1360, 2008.



- [33] B. Hofmann, L. Franke, E. Proschak, Y. Tanrikulu, P. Schneider, D. Steinhilber, and G. Schneider. Scaffold-hopping cascade yields potent inhibitors of -lipoxygenase. *ChemMedChem*, *accepted.*, 2009.
- [34] B. Krüger, A. Dietrich, K.-H. Baringhaus, and G. Schneider. Scaffold-hopping potential of fragment-based de novo design: The chances and limits of variation. *Comb. Chem. High-Throughput Screen.*, *accepted.*, 2009.
- [35] Y. Tanrikulu and G. Schneider. Pseudoreceptor models in drug design: Bridging ligand- and receptor-based virtual screening. *Nature Rev. Drug Discov.*, 7:668–677, 2008.
- [36] A. Schüller and G. Schneider. Identification of hits and lead structure candidates with limited resources by adaptive optimization. *J. Chem. Inf. Model.*, *accepted.*, 2008.
- [37] Alexander J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004. URL <http://dx.doi.org/10.1023/B:STC0.0000035301.49549.88>.
- [38] 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.
- [39] Vojtech Franc and Sören Sonnenburg. OCAS optimized cutting plane algorithm for support vector machines. In *Proceedings of the 25nd International Machine Learning Conference*. ACM Press, 2008. URL <http://ida.first.fraunhofer.de/~franc/ocas/html/index.html>.
- [40] Sebastian Mika, Christin Schäfer, Pavel Laskov, David Tax, and Klaus-Robert Müller. Support vector machines. In James E. Gentle, Wolfgang Härdle, and Yasuo Mori, editors, *Handbook of Computational Statistics*. Springer, Berlin, 2004.
- [41] Masashi Sugiyama, Motoaki Kawanabe, and Klaus-Robert Müller. Trading variance reduction with unbiasedness: The regularized subspace information criterion for robust model selection in kernel regression. *Neural Computation*, 16(5):1077–1104, 2004.
- [42] P. Laskov, C. Gehl, S. Krüger, and K. R. Müller. Incremental support vector learning: Analysis, implementation and applications. *Journal of Machine Learning Research*, 7:1909–1936, September 2006.

- [43] Carl Edward Rasmussen and Christopher K.I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2005.
- [44] Kai Yu, Volker Tresp, and Anton Schwaighofer. Learning Gaussian processes from multiple tasks. In Luc De Raedt and Stefan Wrobel, editors, *Machine Learning: Proceedings of the 22nd International Conference (ICML 2005)*. Morgan Kaufman, 2005.
- [45] Gilles Blanchard, Christin Schäfer, Yves Rozenholc, and Klaus-Robert Müller. Optimal dyadic decision trees. *Machine Learning*, 66(2–3):209–241, 2007. URL <http://dx.doi.org/10.1007/s10994-007-0717-6>.
- [46] Klaus Pawelzik, Klaus-Robert Müller, and Jens Kohlmorgen. Prediction of mixtures. In *Artificial Neural Networks – ICANN ’96*, pages 127–132. Springer, 1996.
- [47] K.-R. Müller, N. Murata, A. Ziehe, and S.-I. Amari. *On-line learning in Switching and Drifting environments with application to blind source separation*, pages 93–110. On-line learning in neural networks. Cambridge University Press, 1998.
- [48] Klaus-Robert Müller, Gunnar Rätsch, Jens Kohlmorgen, Alexander Smola, Bernhard Schölkopf, and Vapnik Vladimir. Time series prediction using support vector regression and neural networks. In T. Higuchi and Y. Takizawa, editors, *2nd International Symposium on Frontiers of Time Series Modelling: Nonparametric Approach to Knowledge Discovery*. Institute of mathematical statistics publication, 2000.
- [49] Roberto Todeschini and Viviana Consonni. *Handbook of Molecular Descriptors*. John Wiley & Sons, Ltd., 2000.
- [50] Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009. URL <http://pages.cs.wisc.edu/~bsettles/active-learning/>.
- [51] Pavel Laskov, Konrad Rieck, Christin Schäfer, and Klaus-Robert Müller. Visualization of anomaly detection using prediction sensitivity. In *Sicherheit 2005 (Sicherheit-Schutz und Verlässlichkeit)*, pages 197–208, 2005.
- [52] P. Laskov, C. Schäfer, I. Kotenko, and K.-R. Müller. Intrusion detection in unlabeled data with quarter-sphere support vector machines (extended version). *Praxis der Informationsverarbeitung und Kommunikation*, 27:228–236, 2004.

- [53] Patrick Düssel, Christian Gehl, Pavel Laskov, and Konrad Rieck. Incorporation of application layer protocol syntax into anomaly detection. In *Proc. of International Conference on Information Systems Security (ICISS)*, pages 188–202, 2008.
- [54] Stefan Wahl, Konrad Rieck, Pavel Laskov, Peter Domschitz, and Klaus-Robert Müller. Securing IMS against novel threats. *Bell Labs Technical Journal*, 2009. to appear.
- [55] Tammo Krueger, Christian Gehl, Konrad Rieck, and Pavel Laskov. An architecture for inline anomaly detection. In *Proc. of European Conference on Computer Network Defense (EC2ND)*, pages 11–18, 2008.
- [56] Konrad Rieck and Pavel Laskov. Linear-time computation of similarity measures for sequential data. *Journal of Machine Learning Research*, 9(Jan):23–48, 2008.
- [57] Pavel Laskov, Konrad Rieck, and Klaus-Robert Müller. Machine learning for intrusion detection. In *Mining Massive Data Sets for Security*. IOS press, 2008. (in press).
- [58] Matthias Krauledat. *Analysis of Nonstationarities in EEG signals for improving Brain-Computer Interface performance*. PhD thesis, Technische Universität Berlin, Fakultät IV – Elektrotechnik und Informatik, 2008.
- [59] Klaus-Robert Müller, Michael Tangermann, Guido Dornhege, Matthias Krauledat, Gabriel Curio, and Benjamin Blankertz. Machine learning for real-time single-trial EEG-analysis: From brain-computer interfacing to mental state monitoring. *J Neurosci Methods*, 167(1):82–90, 2008. URL <http://dx.doi.org/10.1016/j.jneumeth.2007.09.022>.
- [60] Carmen Vidaurre, Alois Schlögl, Benjamin Blankertz, Motoaki Kawanabe, and Klaus-Robert Müller. Unsupervised adaptation of the LDA classifier for Brain-Computer Interfaces. In *Proceedings of the 4th International Brain-Computer Interface Workshop and Training Course 2008*. Verlag der Technischen Universität Graz, 2008. to appear.
- [61] Anton Nijholt, Desnay Tan, Gert Pfurtscheller, Clemens Brunner, José Millán, Brendan Allison, Bernhard Grainmann, Florin Popescu, Benjamin Blankertz, and Klaus-Robert Müller. Brain-computer interfacing for intelligent systems. *IEEE Intelligent Systems*, 23(3):72–79, 2008. URL <http://dx.doi.org/10.1109/MIS.2008.41>.

- [62] Matthias Krauledat, Michael Tangermann, Benjamin Blankertz, and Klaus-Robert Müller. Towards zero training for brain-computer interfacing. *PLoS ONE*, 3(8):e2967, Aug 2008. doi: 10.1371/journal.pone.0002967. URL <http://dx.doi.org/10.1371%2Fjournal.pone.0002967>.
- [63] Michael Tangermann, Matthias Krauledat, Konrad Grzeska, Max Sagebaum, Benjamin Blankertz, and Klaus-Robert Müller. Playing pinball with non-invasive BCI. In *Advances in Neural Information Processing Systems 21*. MIT Press, Cambridge, MA, 2009. in press.
- [64] Benjamin Blankertz, Florian Losch, Matthias Krauledat, Guido Dornhege, Gabriel Curio, and Klaus-Robert Müller. The Berlin Brain-Computer Interface: Accurate performance from first-session in BCI-naïve subjects. *IEEE Trans Biomed Eng*, 55(10):2452–2462, 2008. URL <http://dx.doi.org/10.1109/TBME.2008.923152>.
- [65] Mikio L. Braun, Joachim Buhmann, and Klaus-Robert Müller. On relevant dimensions in kernel feature spaces. *Journal of Machine Learning Research*, 9:1875–1908, Aug 2008.
- [66] Mikio L. Braun. Accurate bounds for the eigenvalues of the kernel matrix. *Journal of Machine Learning Research*, 7:2303–2328, Nov 2006.
- [67] Mikio L. Braun, Tilman Lange, and Joachim Buhmann. Model selection in kernel methods based on a spectral analysis of label information. In *Proc. DAGM*, volume 4174 of *LNCS*, pages 344–353, 2006.
- [68] Mikio L. Braun, Joachim Buhmann, and Klaus-Robert Müller. Denoising and dimension reduction in feature space. In *Advances in Neural Inf. Proc. Systems (NIPS 20)*, 2007. accepted.
- [69] S. Harmeling, A. Ziehe, M. Kawanabe, B. Blankertz, and K.-R. Müller. Nonlinear blind source separation using kernel feature spaces. In T.-W. Lee, editor, *Proc. Int. Workshop on Independent Component Analysis and Blind Signal Separation (ICA2001)*, pages 102–107, 2001.
- [70] Frank C. Meinecke, Andreas Ziehe, Motoaki Kawanabe, and Klaus-Robert Müller. Assessing reliability of ICA projections – a resampling approach. In T.-W. Lee, editor, *Proc. Int. Workshop on Independent Component Analysis and Blind Signal Separation (ICA2001)*, 2001.
- [71] S. Harmeling, A. Ziehe, M. Kawanabe, and K.-R. Müller. Kernel feature spaces and nonlinear blind source separation. In T.G. Dietterich,

- S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2002.
- [72] S. Harmeling, A. Ziehe, M. Kawanabe, and K.-R. Müller. Kernel-based nonlinear blind source separation. *Neural Computation*, 15:1089–1124, 2003.
- [73] Andreas Ziehe, Motoaki Kawanabe, Stefan Harmeling, and Klaus-Robert Müller. Blind separation of post-nonlinear mixtures using linearizing transformations and temporal decorrelation. *Journal of Machine Learning Research*, 4:1319–1338, Dec 2003.
- [74] Sören Sonnenburg, Mikio Braun, Cheng Soon Ong, Samy Bengio, Leon Bottou, Geoffrey Holmes, Yann LeCun, Klaus-Robert Müller, Fernando Pereira, Carl Edward Rasmussen, Gunnar Rätsch, Bernhard Schölkopf, Alexander Smola, Pascal Vincent, Jason Weston, and Robert Williamson. The Need for Open Source Software in Machine Learning. *Journal of Machine Learning Research*, 8:2443–2466, September 2007.
- [75] *ACD/Solubility Batch v9.0*. Toronto, Canada, 2006.
- [76] S. Mika, B. Schölkopf, A.J. Smola, K.-R. Müller, M. Scholz, and G. Rätsch. Kernel PCA and de-noising in feature spaces. In M.S. Kearns, S.A. Solla, and D.A. Cohn, editors, *Advances in Neural Inf. Proc. Systems (NIPS 98)*, pages 536–542. MIT Press, 1999.
- [77] B. Schölkopf, S. Mika, C.J.C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A.J. Smola. Input space vs. feature space in kernel-based methods. *IEEE Transactions on Neural Networks*, 10(5):1000–1017, September 1999.
- [78] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, A.J. Smola, and K.-R. Müller. Invariant feature extraction and classification in kernel spaces. In S.A. Solla, T.K. Leen, and K.-R. Müller, editors, *Proc. NIPS 12*, pages 526–532. MIT Press, 2000.
- [79] Gilles Blanchard, Olivier Bousquet, and Laurent Zwald. Statistical properties of kernel components analysis. *Machine Learning*, 66(2-3):259–294, 2007. URL <http://dx.doi.org/10.1007/s10994-006-6895-9>.

- [80] B. Schölkopf, S. Mika, A.J. Smola, G. Rätsch, and K.-R. Müller. Kernel PCA pattern reconstruction *via* approximate pre-images. In L. Niklasson, M. Bodén, and T. Ziemke, editors, *Proceedings of the 8th International Conference on Artificial Neural Networks*, Perspectives in Neural Computing, pages 147 – 152, Berlin, 1998. Springer Verlag.
- [81] Laurent Zwald and Gilles Blanchard. On the convergence of eigenspaces in kernel principal components analysis. In *Advances in Neural Inf. Proc. Systems (NIPS 05)*, volume 18, 2006.
- [82] Stefan Harmeling, Guido Dornhege, David Tax, Frank C. Meinecke, and Klaus-Robert Müller. From outliers to prototypes: ordering data. *Neurocomputing*, 69(13–15):1608–1618, 2006.
- [83] Ying Xue Min Wang, Xue-Gang Yang. Identifying herg potassium channel inhibitors by machine learning methods. *QSAR Comb. Sci.*, 27:1028–1035, 2008. URL <http://dx.doi.org/10.1002/qsar.200810015>.
- [84] Aixia Yan, Zhi Wang, and Zongyuan Cai. Prediction of human intestinal absorption by ga feature selection and support vector machine regression. *International Journal of Molecular Sciences*, 9(10):1961–1976, 2008. ISSN 1422-0067. doi: 10.3390/ijms9101961. URL <http://www.mdpi.com/1422-0067/9/10/1961>.
- [85] Andreas Bender, Hamse Y. Mussa, Robert C. Glen, and Stephan Reiling. Molecular similarity searching using atom environments, information-based feature selection, and a naive bayesian classifier. *Journal of Chemical Information and Computer Sciences*, 44(1):170–178, 2004. doi: 10.1021/ci034207y. URL <http://pubs.acs.org/doi/abs/10.1021/ci034207y>.
- [86] Jorg K. Wegner and Andreas Zell. Prediction of aqueous solubility and partition coefficient optimized by a genetic algorithm based descriptor selection method. *Journal of Chemical Information and Computer Sciences*, 43(3):1077–1084, 2003. doi: 10.1021/ci034006u. URL <http://pubs.acs.org/doi/abs/10.1021/ci034006u>.
- [87] Igor V. Tetko, Iurii Sushko, Anil Kumar Pandey, Hao Zhu, Alexander Tropsha, Ester Papa, Tomas Oberg, Roberto Todeschini, Denis Fourches, and Alexandre Varnek. Critical assessment of qsar models of environmental toxicity against tetrahymena pyriformis: Focusing on applicability domain and overfitting by variable selection. *Journal*

- of Chemical Information and Modeling*, 48(9):1733–1746, 2008. doi: 10.1021/ci800151m. URL <http://pubs.acs.org/doi/abs/10.1021/ci800151m>.
- [88] Yuan Calan Qi, Thomas P. Minka, and Rosalind W. Picard. Predictive automatic relevance determination by expectation propagation. In *In Proceedings from the International Conference on Machine Learning*, pages 4–8, 2004.
- [89] Anthony O’Hagan. Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society, Series B: Methodological*, 40(1):1–42, 1978.
- [90] Olga Obrezanova, Gábor Csányi, Joelle M.R. Gola, and Matthew D. Segall. Gaussian processes: A method for automatic QSAR modelling of adme properties. *J. Chem. Inf. Model.*, 47(5):1847–1857, 2007. URL <http://dx.doi.org/10.1021/ci7000633>.
- [91] Carl Edward Rasmussen. The gaussian process webssite <http://www.gaussianprocess.org/>, 2009.
- [92] Bernhard Schölkopf and Alex J. Smola. *Learning with Kernels*. MIT Press, 2002.
- [93] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, 1995.
- [94] Radford M. Neal. *Bayesian Learning for Neural Networks*. Springer Verlag, 1996.
- [95] C. Zhu, R. H. Byrd, and J. Nocedal. L-bfgs-b: Algorithm 778: L-bfgs-b, fortran routines for large scale bound constrained optimization. *ACM Transactions on Mathematical Software*, 23(4):550–560, 1997.
- [96] Radford M. Neal. Regression and classification using Gaussian process priors. In José M. Bernardo, J.O. Berger, A.P. Dawid, and A.F.M. Smith, editors, *Bayesian Statistics 6*, volume 6, pages 475–501. Oxford University Press, 1998.
- [97] Alexandre Varnek, Cedric Gaudin, Gilles Marcou, Igor Baskin, Anil Kumar Pandey, and Igor V. Tetko. Inductive transfer of knowledge: Application of multi-task learning and feature net approaches to model tissue-air partition coefficients. *Journal of Chemical Information and Modeling*, 49(1):133–144, 2009. doi: 10.1021/ci8002914. URL <http://pubs.acs.org/doi/abs/10.1021/ci8002914>.

- [98] Olga Obrezanova, Joelle M. R. Gola, Edmund J. Champness, and Matthew D. Segall. Automatic qsar modeling of adme properties: blood-brain barrier penetration and aqueous solubility. *J. Comput.-Aided Mol. Des.*, 22:431–440, 2008. URL <http://dx.doi.org/10.1007/s10822-008-9193-8>.
- [99] Klaus-Robert Müller, Gunnar Rätsch, Sören Sonnenburg, Sebastian Mika, Michael Grimm, and Nikolaus Heinrich. Classifying ‘drug-likeness’ with kernel-based learning methods. *J. Chem. Inf. Model*, 45:249–253, 2005.
- [100] M. K Warmuth, J. Liao, G. Rätsch, Mathieson. M., S. Putta, and C. Lemmem. Active learning with Support Vector Machines in the drug discovery process. *Journal of Chemical Information and Computer Sciences*, 43(2):667–673, 2003.
- [101] Leo Breiman. Random forests. *Machine Learning*, 45:5–32, 2001. URL <http://dx.doi.org/10.1023/A:1010933404324>.
- [102] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Verlag, 2001.
- [103] Tapio Pahikkala, Jorma Boberg, and Tapio Salakoski. T.: Fast n-fold cross-validation for regularized least-squares. In *In: Proceedings of the Ninth Scandinavian Conference on Artificial Intelligence (SCAI, 2006*.
- [104] Gerard E. Dallal. Correlation coefficients <http://www.jerrydallal.com/LHSP/corr.htm>, 1999.
- [105] Maurice Kendall. *Rank Correlation Methods*. Charles Griffin & Company Limited, 1948.
- [106] Matthias Rupp, Ewgenij Proschak, and Gisbert Schneider. Kernel approach to molecular similarity based on iterative graph similarity. *Journal of Chemical Information and Modelling*, 47(6):2280–2286, 2007.
- [107] Gunnar Rätsch, Sören Sonnenburg, Jagan Srinivasan, Hanh Witte, Ralf Sommer, Klaus-Robert Müller, and Bernhard Schölkopf. Improving the c. elegans genome annotation using machine learning. *PLoS Computational Biology*, 3:e20, 2007.
- [108] Sören Sonnenburg, Gunnar Rätsch, Christin Schäfer, and Bernhard Schölkopf. Large Scale Multiple Kernel Learning. *Journal of Machine Learning Research*, 7:1531–1565, July 2006.



- [109] Alexander Tropsha. Variable selection qsar modeling, model validation, and virtual screening. In David C. Spellmeyer, editor, *Annual Reports in Computational Chemistry*, volume 2, chapter 7, pages 113–126. Elsevier, 2006.
- [110] Weida Tong, Qian Xie, Huixiao Hong, Leming Shi, Hong Fang, and Roger Perkins. Assessment of prediction confidence and domain extrapolation of two structure-activity relationship models for predicting estrogen receptor binding activity. *Environmental Health Perspectives*, 112(12):1249–1254, 2004.
- [111] Tatiana I. Netzeva, Andrew P. Worth, Tom Aldenberg, Romualdo Benigni, Mark T.D. Cronin, Paola Gramatica, Joanna S. Jaworska, Scott Kahn, Gilles Klopman, Carol A. Marchant, Glenn Myatt, Nina Nikolova-Jeliazkova, Grace Y. Patlewicz, Roger Perkins, David W. Roberts, Terry W. Schultz, David T. Stanton, Johannes J.M. van de Sandt, Weida Tong, Gilman Veith, and Chihae Yang. Current status of methods for defining the applicability domain of (quantitative) structure-activity relationships. *Alternatives to Laboratory Animals*, 33(2):1–19, 2005.
- [112] Bernard W. Silverman. *Density Estimation for Statistics and Data Analysis*. Number 26 in Monographs on Statistics and Applied Probability. Chapman & Hall, 1986.
- [113] Robert P. Sheridan, Bradley P. Feuston, Vladimir N. Maiorov, and Simon K. Kearsley. Similarity to molecules in the training set is a good discriminator for prediction accuracy in qsar. *J. Chem. Inf. Model.*, 44:1912–1928, 2004.
- [114] Igor V. Tetko, Pierre Bruneau, Hans-Werner Mewes, Douglas C. Rohrer, and Gennadiy I. Poda. Can we estimate the accuracy of ADME-tox predictions? *Drug Discovery Today*, 11(15/16):700–707, August 2006.
- [115] Pierre Bruneau and Nathan R. McElroy. Generalized fragment-substructure based property prediction method. *J. Chem. Inf. Model.*, 46:1379–1387, 2006.
- [116] Andreas H. Göller, Matthias Hennemann, Jörg Keldenich, and Timothy Clark. In silico prediction of buffer solubility based on quantum-mechanical and hqsar- and topology-based descriptors. *J. Chem. Inf. Model.*, 46(2):648–658, 2006.

- [117] David T. Manallack, Benjamin G. Tehan, Emanuela Gancia, Brian D. Hudson, Martyn G. Ford, David J. Livingstone, David C. Whitley, and Will R. Pitt. A consensus neural network-based technique for discriminating soluble and poorly soluble compounds. *J. Chem. Inf. Model.*, 43:674–679, 2003.
- [118] Ralph Kühne, Ralf-Uwe Ebert, and Gerrit Schürmann. Model selection based on structural similarity-method description and application to water solubility prediction. *J. Chem. Inf. Model.*, 46:636–641, 2006.
- [119] G. A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63:81–97, 1956.
- [120] Bernhard Schölkopf, Ralf Herbrich, and Alex Smola. A generalized representer theorem. In *Lecture Notes in Computer Science*, pages 416–426. Springer Berlin / Heidelberg, 2001. URL [http://dx.doi.org/10.1007/3-540-44581-1\\_27](http://dx.doi.org/10.1007/3-540-44581-1_27).
- [121] 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*, 2008. URL <http://dx.doi.org/10.1021/ci700142c>.
- [122] World drug index WDI. World drug index WDI, 1996. version 2/96.
- [123] R. Todeschini, V. Consonni, A. Mauri, and M. Pavan. *DRAGON v1.2*. Milano, Italy, 2006.
- [124] David Baehrens. Explaining machine-learned model predictions to support human expert decisions. Diplomarbeit, Machine Learning Dept., University of Technology Berlin, December 2009.
- [125] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *JMLR*, 3:1157–1182, 2003.
- [126] C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
- [127] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Number 31 in Applications of Mathematics. Springer, New York, 1996.

- [128] 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.
- [129] E. J. Horvitz, J. S. Breese, and M. Henrion. Decision theory in expert systems and artificial interigence. *Journal of Approximation Reasoning*, 2:247–302, 1988. Special Issue on Uncertainty in Artificial Intelligence.
- [130] H. Suermondt. *Explanation in Bayesian Belief Networks*. PhD thesis, Department of Computer Science and Medicine, Stanford University, Stanford, CA, 1992.
- [131] Marko Robnik-Sikonja and Igor Kononenko. Explaining classifications for individual instances. *IEEE TKDE*, 20(5):589–600, 2008.
- [132] Erik Strumbelj and Igor 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.
- [133] T.J. Hou and X.J. Xu. Adme evaluation in drug discovery. 3. modeling blood-brain barrier partitioning using simple molecular descriptors. *J. Chem. Inf. Comput. Sci.*, 43(6):2137–2152, 2003.
- [134] Eric D. Clarke and John S. Delaney. Physical and molecular properties of agrochemicals: An analysis of screen inputs, hits, leads, and products. *Chimia*, 57(11):731–734, 2003.
- [135] Jarmo Huuskonen. Estimation of aqueous solubility for a diverse set of organic compounds based on molecular topology. *J. Chem. Inf. Comput. Sci.*, 40(3):773–777, 2000.
- [136] Igor V. Tetko, Vsevolod Y. Tanchuk, Tamara N. Kasheva, and Alessandro E.P. Villa. Estimation of aqueous solubility of chemical compounds using e-state indices. *J. Chem. Inf. Comput. Sci.*, 41(6):1488–1493, 2001.
- [137] Aixia Yan and Johann Gasteiger. Prediction of aqueous solubility of organic compounds by topological descriptors. *QSAR Comb. Sci.*, 22(8):821–829, 2003.
- [138] Jörg K. Wegner and Andreas Zell. Prediction of aqueous solubility and partition coefficient optimized by a genetic algorithm based descriptor selection. *J. Chem. Inf. Comput. Sci.*, 43(3):1077–1084, 2003.

- [139] Aixia Yan and Johann Gasteiger. Prediction of aqueous solubility of organic compounds based on a 3d structure representation. *J. Chem. Inf. Comput. Sci.*, 43:429–434, 2003.
- [140] Ruifeng Liu, Hongmao Sun, and Sung-Sau So. Development of quantitative structure - property relationship models for early adme evaluation in drug discovery. 1. aqueous solubility. *J. Chem. Inf. Comput. Sci.*, 41(6):1633–1639, 2001.
- [141] John S. Delaney. Esol: Estimating aqueous solubility directly from molecular structure. *J. Chem. Inf. Comput. Sci.*, 44(3):1000–1005, 2004.
- [142] Pierre Bruneau. Search for predictive generic model of aqueous solubility using bayesian neural nets. *J. Chem. Inf. Comput. Sci.*, 41(6):1605–1616, 2001.
- [143] Denise Yaffe, Yoram Cohen, Gabriela Espinosa, Alex Arenas, and Francesc Giralt. A fuzzy artmap based on quantitative structure-property relationships (qsprs) for predicting aqueous solubility of organic compounds. *J. Chem. Inf. Comput. Sci.*, 41(5):1177–1207, 2001.
- [144] Ola Engkvist and Paul Wrede. High-throughput, in silico prediction of aqueous solubility based on one- and two-dimensional descriptors. *J. Chem. Inf. Comput. Sci.*, 42(5):1247–1249, 2002.
- [145] Aixia Yan, Johann Gasteiger, Michael Krug, and Soheila Anzali. Linear and nonlinear functions on modeling of aqueous solubility of organic compounds by two structure representation methods. *J. Comput.-Aided Mol. Des.*, 18:75–87, 2004.
- [146] T. J. Hou, K. Xia, W. Zhang, and X. J. Xu. Adme evaluation in drug discovery. 4. prediction of aqueous solubility based on atom contribution approach. *J. Chem. Inf. Comput. Sci.*, 44:266–275, 2004.
- [147] Holger Fröhlich, Jörg K. Wegner, and Andreas Zell. Towards optimal descriptor subset selection with support vector machines in classification and regression. *QSAR Comb. Sci.*, 23:311–318, 2004.
- [148] Matthew Clark. Generalized fragment-substructure based property prediction method. *J. Chem. Inf. Model.*, 45:30–38, 2005.
- [149] Fred-Reiner Rapp. *Das Programmsystem SOLVES zur Suche und Optimierung von Leitstrukturen*. PhD thesis, Fakultät für Informations-

- und Kognitionswissenschaften, Eberhard-Karls-Universität Tübingen, Germany, 2005.
- [150] Konstantin V. Balakin, Nikolay P. Savchuk, and Igor V. Tetko. In silico approaches to prediction of aqueous and DMSO solubility of drug-like compounds: Trends, problems and solutions. *Curr. Med. Chem.*, 13: 223–241, 2006.
- [151] Stephen R. Johnson and Weifan Zheng. Recent progress in the computational prediction of aqueous solubility and absorption. *The AAPS Journal*, 8(1):E27–E40, 2006. URL <http://www.aapsj.org/articles/aapsj0801/aapsj080104/aapsj080104.pdf>.
- [152] John S. Delaney. Predicting aqueous solubility from structure. *Drug Discovery Today*, 10(4):289–295, 2005.
- [153] Carol A. Kempa, Jean-Didier Marechal, and Michael J. Sutcliffe. Progress in cytochrome P450 active site modeling. *Arch. Biochem. Biophys.*, 433:361–368, 2005.
- [154] Ilia G. Denisov, Thomas M. Makris, Stephen G. Sligar, and Ilme Schlichting. Structure and chemistry of cytochrome P450. *Chem. Rev. (Washington, DC, U.S.)*, 105:2253–2277, 2005.
- [155] Arthur G. Roberts, A. Patricia Campbell, and William M. Atkins. The thermodynamic landscape of testosterone binding to cytochrome P450 3a4: Ligand binding and spin state equilibria. *Biochemistry*, 44:1353–1366, 2005.
- [156] David F. V. Lewis and Maurice Dickins. Substrate sars in human p450s. *Drug Discovery Today*, 7:918–925, 2002.
- [157] Sean Ekins. Predicting undesirable drug interactions with promiscuous proteins in silico. *Drug Discovery Today*, 9:276–285, 2004.
- [158] Jan M. Kriegl, Thomas Arnhold, Bernd Beck, and Thomas Fox. Prediction of human cytochrome P450 inhibition using support vector machines. *QSAR Comb. Sci.*, 24:491–502, 2005.
- [159] Christian Merkwirth, Harald Mauser, Tanja Schulz-Gasch, Olivier Roche, Martin Stahl, and Thomas Lengauer. Ensemble methods for classification in cheminformatics. *J. Chem. Inf. Comput. Sci.*, 44:1971–1978, 2004.

- [160] Achim Kless and Tatjana Eitrich. Cytochrome P450 classification of drugs with support vector machines implementing the nearest point algorithm. In *Knowledge Exploration in Life Science Informatics: International Symposium KELSI 2004*, volume 3303 of *Lecture Notes in Artificial Intelligence*, pages 191–205. Springer Verlag, 2004.
- [161] Paul Rowland, Frank E. Blaney, Martin G. Smyth, Jo J. Jones, Vaughan R. Leydon, Amanda K. Oxbrow, Ceri J. Lewis, Mike G. Tennant, Sandeep Modi, Drake S. Eggleston, Richard J. Chenery, and Angela M. Bridges. Crystal Structure of Human Cytochrome P450 2D6. *J. Biol. Chem.*, 281(11):7614–7622, 2006. doi: 10.1074/jbc.M511232200. URL <http://dx.doi.org/10.1074/jbc.M511232200>.
- [162] Pamela A. Williams, Jose Cosme, Dijana Matak Vinkovic, Alison Ward, Hayley C. Angove, Philip J. Day, Clemens Vornrhein, Ian J. Tickle, and Harren Jhoti. Crystal Structures of Human Cytochrome P450 3A4 Bound to Metirapone and Progesterone. *Science*, 305(5684):683–686, 2004. URL <http://dx.doi.org/10.1126/science.1099736>.
- [163] Pamela A. Williams, Jose Cosme, Alison Ward, Hayley C. Angove, Dijana Matak Vinkovi, and Harren Jhoti. Crystal structure of human cytochrome p450 2c9 with bound warfarin. *Nature*, 424:464–468, 2003. URL <http://dx.doi.org/10.1038/nature01862>.
- [164] Vellarkad N. Viswanadhan, Arup K. Ghose, Ganapathi R. Reyankar, and Roland K. Robins. Atomic physicochemical parameters for three dimensional structure directed quantitative structure-activity relationships. 4. additional parameters for hydrophobic and dispersive interactions and their application for an automated superposition of certain naturally occurring nucleoside antibiotics. *J. Chem. Inf. Comput. Sci.*, 29:163–172, 1989.
- [165] Tommi Jaakkola and David Haussler. Exploiting generative models in discriminative classifiers. In *In Advances in Neural Information Processing Systems 11*, pages 487–493. MIT Press, 1998.
- [166] Frank R. Burden. A chemically intuitive molecular index based on the eigenvalues of a modified adjacency matrix. *Quant. Struct. Act. Relat.*, 16:309–314, 1997.
- [167] Robert S. Pearlman and K.M. Smith. Novel software tools for chemical diversity. *Perspectives in Drug Discovery and Design*, 9-11:339–353, 1998.

- [168] U.S.A. Tripos Inc.: St.Louis, MO. Unity reference manual, 2006.
- [169] G. Cruciani, P. Crivori, P.-A. Carrupt, and B. Testa. Molecular fields in quantitative structure-permeation relationships: The volsurf approach. *THEOCHEM*, 503:17–30, 2000.
- [170] Manuel Pastor, Gabriele Cruciani, Iain McLay, Stephen Pickett, and Sergio Clementi. Grid-independent descriptors (grind): A novel class of alignment-independent three-dimensional molecular descriptors. *J. Med. Chem.*, 43:3233–3243, 2000.
- [171] J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1992.
- [172] T.M. Cover and P.E. Hart. Nearest neighbor pattern classifications. *IEEE transaction on information theory*, 13(1):21—27, 1967.
- [173] V.N. Vapnik. *The nature of statistical learning theory*. Springer Verlag, New York, 1995.
- [174] B. Schölkopf and A.J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.
- [175] K.P. Bennett and O.L. Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software*, 1:23–34, 1992.
- [176] T. Graepel, R. Herbrich, B. Schölkopf, A.J. Smola, P.L. Bartlett, K.-R. Müller, K. Obermayer, and R.C. Williamson. Classification on proximity data with LP-machines. In D. Willshaw and A. Murray, editors, *Proceedings of ICANN’99*, volume 1, pages 304–309. IEE Press, 1999.
- [177] M. Shen, Y. Xiao, A. Golbraikh, V.K. Gombar, and A. Tropsha. An in silico screen for human S9 metabolic turnover using k-nearest neighbor QSPR method. *J. Med. Chem.*, 46:3013–3020, 2003.
- [178] Berith F. Jensen, Morten D. Sorensen, Kissmeyer Anne-Marie, Frederik Björkling, Kim Sonne, Sören B. Engelsen, and Lars Norgaard. Prediction of in vitro metabolic stability of calcitriol analogs by QSAR. *J. Comput.-Aided Mol. Des.*, 17(12):849–859, 2003.
- [179] Bruce N. Ames, E. G. Gurney, James A. Miller, and H. Bartsch. Carcinogens as Frameshift Mutagens: Metabolites and Derivatives of 2-Acetylaminofluorene and Other Aromatic Amine Carcinogens. *Proceed-*

- ings of the National Academy of Sciences of the United States of America*, 69(11):3128–3132, 1972. URL <http://www.pnas.org/content/69/11/3128.abstract>.
- [180] Christoph Helma, Tobias Cramer, Stefan Kramer, and Luc De Raedt. Data mining and machine learning techniques for the identification of mutagenicity inducing substructures and structure activity relationships of noncongeneric compounds. *Journal of Chemical Information and Computer Sciences*, 44(4):1402–11, 2004. doi: 10.1021/ci034254q. URL <http://www.ncbi.nlm.nih.gov/pubmed/15272848>. PMID: 15272848.
- [181] J Kazius, R Mcguire, and R Bursi. Derivation and validation of toxiphores for mutagenicity prediction. *J. Med. Chem.*, 48(1):320, 312, 2005. URL <http://dx.doi.org/10.1021/jm040835a>.
- [182] Jun Feng, Laura Lurati, Haojun Ouyang, Tracy Robinson, Yuanyuan Wang, Shenglan Yuan, and S Stanley Young. Predictive toxicology: benchmarking molecular descriptors and statistical methods. *Journal of Chemical Information and Computer Sciences*, 43(5):1463–70, 2003. URL <http://www.ncbi.nlm.nih.gov/pubmed/14502479>. PMID: 14502479.
- [183] Accelrys Software Inc. *SciTegic Pipeline Pilot v7.0 Help Guide 2009*, 2008. available at <http://accelrys.com/products/scitegic/>.
- [184] DM Sanderson and CG Earnshaw. Computer prediction of possible toxic action from chemical structure, the derek system. *Hum Exp Toxicol*, 10(4):261–73, jul 1991.
- [185] Gilles Klopman. Multicase 1. a hierarchical computer automated structure evaluation program. *Quantitative Structure-Activity Relationships*, 11(2):176–184, 1992.
- [186] R.R. Fenichel, M. Malik, C. Antzelevitch, M. Sanguinetti, D.M. Roden, S.G. Priori, J.N. Ruskin, R.J. Lipicky, and L.R. Cantilena. Drug-induced torsades de pointes and implications for drug development. *J. Cardiovasc. Electrophysiol.*, 15:475–495, 2004.
- [187] M. Recanatini, E. Poluzzi, M. Masetti, A. Cavalli, and F. De Ponti. Qt prolongation through herg k(+) channel blockade: Current knowledge and strategies for the early prediction during drug development. *Med. Res. Rev.*, 25:133–166, 2005.



- [188] B. Fermini and A.A. Fossa. The impact of drug-induced qt interval prolongation on drug discovery and development. *Nat. Rev. Drug Discov.*, 2:439–447, 2003.
- [189] M.C. Sanguinetti and M. Tristani-Firouzi. herg potassium channels and cardiac arrhythmia. *Nature*, 440:463–469, 2006.
- [190] P.J. Stansfeld, M.J. Sutcliffe, and J.S. Mitcheson. Molecular mechanisms for drug interactions with herg that cause long qt syndrome. *Expert Opin. Drug Metab. Toxicol.*, 2:81–94, 2006.
- [191] A.M. Aronov. Predictive in silico modeling for herg channel blockers. *Drug Discov. Today*, 10:149–155, 2005.
- [192] A.M. Aronov. Tuning out of herg. *Curr. Opin. Drug Discov. Devel.*, 11:128–140, 2008.
- [193] C. Jamieson, E.M. Moir, Z. Rankovic, and G. Wishart. Medicinal chemistry of herg optimizations: Highlights and hang-ups. *J. Med. Chem.*, 49:5029–5046, 2006.
- [194] M.C. Hutter. In silico prediction of drug properties. *Curr. Med Chem.*, 16:189–202, 2009.
- [195] 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. *J. Physiol. Sci.*, 58:459–470, 2008.
- [196] Britta Nisius and Andreas H. Göller. Similarity-based classifier using topomers to provide a knowledge base for herg channel inhibition. *J. Chem. Inf. Model.*, 49(2):247–256, 2009.
- [197] T. Fox and J.M. Kriegl. Machine learning techniques for in silico modeling of drug metabolism. *Curr. Top. Med. Chem.*, 6:1579–1591, 2006.
- [198] O. Obrezanova, G. Csanyi, J.M. Gola, and M.D. Segall. Gaussian processes: a method for automatic qsar modeling of adme properties. *J. Chem. Inf. Model.*, 47:1847–1857, 2007.
- [199] K.M. Thai and G.F. Ecker. Predictive models for herg channel blockers: ligand-based and structure-based approaches. *Curr Med. Chem.*, 14:3003–3026, 2007.

- [200] Britta Nisius, Andreas H. Göller, and Jürgen 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. *Chem. Biol. Drug Des.*, 73:17–25, 2009.
- [201] Q. Li, F.S. Joergensen, T. Oprea, S. Brunak, and O. Taboureau. hERG classification model based on a combination of support vector machine method and grind descriptors. *Mol. Pharm.*, pages 117–127, 2008.
- [202] G. Schneider, W. Neidhart, T. Giller, and G. Schmid. "scaffold-hopping" by topological pharmacophore search: A contribution to virtual screening. *Angew. Chem Int. Ed Engl.*, 38:2894–2896, 1999.
- [203] G. Cruciani, M. Pastor, and W. Guba. VolSurf: a new tool for the pharmacokinetic optimization of lead compounds. *Eur. J. Pharm. Sci.*, 11 Suppl. 2:29–39, 2000.
- [204] Cosimo Gianluca Fortuna, Vincenza Barresi, Giuliano Berellini, and Giuseppe Musumarra. Design and synthesis of trans 2-(furan-2-yl)vinyl heteroaromatic iodide with antitumor activity. *Bioorg. Med. Chem.*, 16: 4150–4159, 2008.
- [205] R. Todeschini, V. Consonni, A. Mauri, and M. Pavan. Dragon for windows and linux, 2006. URL [http://www.talete.mi.it/help/dragon\\_help/](http://www.talete.mi.it/help/dragon_help/). accessed May 14th 2006.
- [206] Don H. Johnson and Sinan Sinanovic. Symmetrizing the kullback-leibler distance. Technical report, IEEE Transactions on Information Theory, 2000.
- [207] Jeroen Kazius, Ross McGuire, and Roberta Bursi. Derivation and validation of toxicophores for mutagenicity prediction. *J. Med. Chem.*, 48:312–320, 2005.
- [208] Hansruedi Glatt, Reinhard Jung, and Franz 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. doi: DOI:10.1016/0027-5107(83)90056-8. URL [http://dx.doi.org/10.1016/0027-5107\(83\)90056-8](http://dx.doi.org/10.1016/0027-5107(83)90056-8).
- [209] Christoph Rücker, Marco Scarsi, and Markus Meringer. 2d qsar of ppar $\gamma$  agonist binding and transactivation. *Bioorganic & Medicinal Chemistry*, 14(15):5178–5195, 2006.

- [210] Asinex. [www.asinex.com](http://www.asinex.com), 3 2008.
- [211] Gisbert Schneider, Werner Neidhart, Thomas Giller, and Gerard Schmid. “scaffold-hopping” by topological pharmacophore search: A contribution to virtual screening. *Angewandte Chemie International Edition*, 38(19):2894–2896, 1999.
- [212] Canada Chemical Computing Group Inc., Montreal. Moe: The molecular operating environment, 2008. URL <http://www.chemcomp.com/journal/descr.htm>. accessed July 31st 2008.
- [213] Arup K. Ghose and Gordon M. Crippen. Atomic physicochemical parameters for three-dimensional structure-directed quantitative structure-activity relationships i. partition coefficients as a measure of hydrophobicity. *J. Comput. Chem.*, 7:565–577, 1986.
- [214] Arup K. Ghose and Gordon M. Crippen. Atomic physicochemical parameters for three-dimensional-structure-directed quantitative structure-activity relationships. 2. modeling dispersive and hydrophobic interactions. *J. Chem. Inf. Comput. Sci.*, 27:21–35, 1987.
- [215] John Guiver and Edward Snelson. Learning to rank with softrank and gaussian processes. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 259–266, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-164-4. doi: <http://dx.doi.org/10.1145/1390334.1390380>. URL <http://dx.doi.org/10.1145/1390334.1390380>.
- [216] Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8:1027–1061, 2007.
- [217] Shimodaira Hidetoshi. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90:227–244(18), 1 October 2000. doi: [10.1016/S0378-3758\(00\)00115-4](https://doi.org/10.1016/S0378-3758(00)00115-4).
- [218] Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul von Buehnan, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in Neural Information Processing Systems 20*, Cambridge, MA, 2007. MIT Press.

- [219] B. Zadrozny. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the 21th International Conference on Machine Learning*, 2004. URL [citeseer.ist.psu.edu/zadrozny04learning.html](http://citeseer.ist.psu.edu/zadrozny04learning.html).
- [220] J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. Lawrence, editors. *Dataset Shift in Machine Learning*. MIT Press, Cambridge, MA, 2009.
- [221] Gerald M. Maggiora. On outliers and activity cliffs why qsar often disappoints. *Journal of Chemical Information and Modeling*, 46(4): 1535–1535, 2006. URL <http://dx.doi.org/10.1021/ci060117s>.
- [222] Rajarshi Guha and John H. Van Drie. Structure-activity landscape index: Identifying and quantifying activity cliffs. *Journal of Chemical Information and Modeling*, 48(3):646–658, 2008. URL <http://dx.doi.org/10.1021/ci7004093>.
- [223] Slobodan Rendic. Summary of information on human CYP enzymes: Human P450 metabolism data. *Drug Metabolism Reviews*, 34:83–448, 2002.
- [224] P450 table. <http://medicine.iupui.edu/flockhart/table.htm> (accessed 14 May 2006), 2006.
- [225] R.S. Pearlman. *Concord*. St. Louis, Missouri, USA, 2005.
- [226] P. J. Goodford. A computational procedure for determining energetically favorable binding sites on biologically important macromolecules. *J. Med. Chem.*, 28:849–857, 1985.