

Enhanced Small Molecule Similarity for Quantitative
Structure-Activity Relationship Modeling and
Cheminformatics Applications

Tobias Girschick

TECHNISCHE UNIVERSITÄT MÜNCHEN

Institut für Informatik

Lehrstuhl für Bioinformatik

Enhanced Small Molecule Similarity for Quantitative
Structure-Activity Relationship Modeling and
Cheminformatics Applications

Tobias Girschick

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. E. W. Mayr

Prüfer der Dissertation:

1. Univ.-Prof. Dr. B. Rost
2. Univ.-Prof. Dr. St. Kramer,
Johannes Gutenberg Universität Mainz

Die Dissertation wurde am 18.12.2013 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 04.05.2014 angenommen.

Contents

| | | |
|-------|--|----|
| 1 | Introduction | 1 |
| 1.1 | Motivation | 1 |
| 1.2 | Quantitative Structure Activity Relationships | 2 |
| 1.2.1 | Applicability Domain | 3 |
| 1.3 | Virtual Screening | 4 |
| 1.3.1 | Evaluation Measures | 6 |
| 1.4 | Outline of the Thesis | 6 |
| 2 | Similarity in Cheminformatics | 9 |
| 2.1 | Molecular Similarity in 2D | 11 |
| 2.1.1 | Fingerprint Based Similarity | 11 |
| 2.1.2 | Maximum Common Substructure Based Similarity | 12 |
| 2.2 | Three Dimensional Similarity Measures | 13 |
| 2.3 | Cheminformatics Applications Relying on Molecular Similarity | 16 |
| 2.4 | Similarity based QSAR | 20 |
| 3 | Distance Learning and Inductive Transfer | 23 |
| 3.1 | Distance Learning | 23 |
| 3.2 | Inductive Transfer | 27 |
| 4 | Similarity Boosted QSAR | 35 |
| 4.1 | Materials and Methods | 36 |
| 4.1.1 | Similarity Descriptors | 36 |
| 4.1.2 | Molecular Similarity Measures | 37 |
| 4.1.3 | Selection of Reference Compounds | 38 |
| 4.1.4 | Core Descriptors | 39 |
| 4.1.5 | Datasets | 40 |
| 4.1.6 | Experimental Setup | 41 |
| 4.2 | Results | 42 |
| 4.2.1 | Similarity Descriptors | 43 |
| 4.2.2 | Selection of Reference Molecules | 43 |
| 4.2.3 | Combining Structural Descriptors and Similarity Descriptors | 45 |

| | | |
|-------|--|-----|
| 4.3 | Conclusions | 47 |
| 5 | Improving Structural Similarity Based Virtual Screening | 51 |
| 5.1 | Materials and Methods | 52 |
| 5.1.1 | Experimental Setup | 52 |
| 5.1.2 | Extended Similarity | 53 |
| 5.1.3 | Data | 56 |
| 5.1.4 | Docking Procedure | 59 |
| 5.2 | Results and Discussion | 61 |
| 5.2.1 | By-Hand Experiments | 61 |
| 5.2.2 | Mining-Based Experiments | 65 |
| 5.3 | Conclusions | 71 |
| 6 | Adapted Transfer of Distance Measures | 73 |
| 6.1 | Distance Learning, Inductive Transfer and Adapted Transfer | 74 |
| 6.2 | Data and Experimental Setup | 76 |
| 6.2.1 | Distances | 78 |
| 6.3 | Experiments | 79 |
| 6.3.1 | Learning Curves | 80 |
| 6.3.2 | Comparison of Approaches | 81 |
| 6.3.3 | Analysis of Optimized Weights | 84 |
| 6.4 | Data-driven Selection of Source Datasets | 87 |
| 6.4.1 | Source Dataset Selection | 87 |
| 6.4.2 | Discussion and Results | 88 |
| 6.4.3 | Comparison with Boosting for Regression Transfer | 89 |
| 6.5 | Conclusion | 92 |
| 7 | Relations Between the Presented Approaches | 95 |
| 7.1 | Similarity Boosted QSAR and Improved Structural Similarities | 96 |
| 7.2 | Adapted Transfer of Distance Measures and Improved Structural Similarities . | 99 |
| 7.3 | Similarity Boosted QSAR and Adapted Transfer of Distance Measures | 100 |
| 8 | Application: Distributed REST Web Services for Toxicity Prediction | 103 |
| 8.1 | OpenTox Philosophy and Background | 106 |
| 8.2 | REST Web Services | 107 |
| 8.3 | The OpenTox Application Programming Interface | 109 |
| 8.4 | Prototype Application: ToxPredict | 111 |
| 8.5 | Conclusion | 112 |
| 9 | Summary and Outlook | 115 |
| 9.1 | Summary | 115 |
| 9.2 | Outlook | 116 |

| | | |
|-----|---|-----|
| A | Additional Material for Chapter 4 | 119 |
| A.1 | \mathcal{R}_{LIT} Reference Compounds | 119 |
| A.2 | Additional Result Tables | 124 |
| A.3 | Diversity measures | 127 |
| B | Additional Material for Chapter 5 | 129 |
| C | Additional Material for Chapter 8 | 137 |
| | List of Figures | 141 |
| | List of Tables | 143 |
| | Bibliography | 145 |

Alle Ding' sind Gift und nichts ist ohn' Gift; allein
die Dosis macht, dass ein Ding' kein Gift ist.

Paracelsus

Acknowledgements

At this point I am glad to have the opportunity to thank all those people who accompanied, encouraged, supported, helped and guided me during my years working on this thesis and without who this thesis would not have been possible. Naturally, I owe the utmost gratitude to my advisor Stefan Kramer. Stefan not only made my participation in two highly interesting, challenging and scientifically fruitful international projects possible — which also constituted my financial support —, but he also guided me in many open discussions through all inevitable imponderabilities of daily research. Without his efforts I would not have the qualifications, experience, and possibilities that I have today.

I am also grateful for the three years in which I shared my office with Fabian Buchwald. We did not only work and publish together, we also had a great time debating FC Bayern games or the Tour de France. I would also like to thank all the other members of the TUM chair, especially Jana Schmidt, Madeleine Seeland, Constanze Schmitt, Matthias Böck, Andreas Hapfelmeier, Jörg Wicker, Simon Berger, Timothy Karl and Lothar Richter, who I had the pleasure to work – and sometimes party with.

I want to thank all members of the OpenTox consortium that I got to know during three years time through meetings and conferences all over Europe as well as through weekly technical discussions in virtual conferences. Particular mentioning is necessary for Barry Hardy and the late Nicki Douglas for coordinating OpenTox as well as for Nina and Vedrin Jeliaskov from Idea Consulting in Sofia, Bulgaria, who showed me and Thilo Winkelmann their beautiful Bulgarian mountains in winter and made an unforgettable fourteen days possible. Special thanks are also due to Jonna Stålring from the AstraZeneca Computational Toxicology group, for the dedicated cooperation on the Similarity Boosted QSAR project. Ulrich Rückert, Eric Alphonse and Eibe Frank worked with me on three projects; thank you for that. Thanks to Martina Rauhmeier for giving me valuable tips on language usage and writing.

In addition, there are people that made this thesis possible through their support and belief in my abilities and – maybe most importantly – their patience. These include my parents, my brother, my friends, Philipp Renner, and, in particular, Claudia Scharl who had to sacrifice uncounted hours that I spent with my work instead of her.

Abstract

This thesis covers enhancements to the concept of small molecule similarity as it is used in Quantitative Structure-Activity Relationships and other cheminformatics applications. The concept of similarity is very central to nearly all areas of cheminformatics research, and consequently, improvements achieved here can be transferred and show their impact in diverse application areas. Industrial applications of cheminformatics support, amongst others, the drug discovery and development workflow in the pharmaceutical industry, the resolving of regulatory problem settings in the whole chemical industry or the research process in agricultural and food sciences. This thesis first presents a novel approach to the usage of small molecule similarities in the descriptor space of Quantitative Structure-Activity Relationships that results in new molecular descriptors that are complementary to structural descriptors. Those descriptors are based on similarities to defined reference structures, either from a set of known active compounds representing different structural classes or from representative structures from the chemical space. Second, an approach that enables improvements to structural similarity based virtual screening by incorporating background knowledge into the similarity measure is presented. In that part, an approach based on literature review as well as one based on data mining is investigated. The new concept of adapted transfer is the third contribution to the field of small molecule similarities. This derivative of inductive transfer, like inductive transfer itself, is especially useful in cases where only limited amounts of training data are available, but a related dataset is at hand. The related dataset can then be utilized as additional knowledge in the learning process. The selection of the related dataset is made either by hand and expert knowledge, or in a semi-automated approach that relates biological assays through activity overlap. The thesis also discusses possible combinations and synergies that can be achieved with the three contributions. If the introduced enhanced concepts of small molecule similarity find their way into practical usage, they can help developing new drugs by speeding up the drug development and registration process or save the lives of laboratory animals by improving in silico toxicity models and making animal tests more and more obsolete.

Zusammenfassung

Diese Arbeit behandelt Verbesserungen im Bereich der Ähnlichkeit von kleinen Molekülen wie sie für quantitative Struktur-Wirkungsbeziehungen und andere Anwendungen der Chemieinformatik genutzt wird. Das Ähnlichkeitskonzept ist sehr zentral für nahezu alle Bereiche der Chemieinformatik, was dazu führt, dass Verbesserungen in vielfältigen Anwendungsbereichen genutzt werden können. Industrielle Anwendungen der Chemieinformatik gibt es beispielsweise im Arzneimittelentwicklungsprozess der Pharmabranche, beim Lösen von regulatorischen Problemstellungen der gesamten chemischen Industrie sowie in der agrar- und lebensmittelwissenschaftlichen Forschung. In dieser Arbeit wird zuerst ein neuer Ansatz zur Verwendung der Ähnlichkeiten von kleinen Molekülen im Deskriptorraum von quantitativen Struktur-Wirkungsbeziehungen vorgestellt, der in einer neuen Art von Moleküldeskriptoren resultiert, welche komplementär zu strukturellen Deskriptoren sind. Diese Deskriptoren basieren auf Ähnlichkeiten zu bestimmten Referenzmolekülen, die entweder aus einer Menge aktiver Strukturen, die verschiedene Strukturklassen repräsentieren, stammen oder repräsentative Strukturen des chemischen Raumes sind. Zweitens wird ein Ansatz vorgestellt, welcher virtuelles Screening basierend auf struktureller Ähnlichkeit dadurch verbessert, dass Hintergrundwissen in das Ähnlichkeitsmass aufgenommen wird. Hier wird ein Ansatz mit Literaturrecherche sowie ein auf Data Mining beruhender Ansatz vorgestellt. Das neue Konzept des adaptiven Transfers ist der dritte Beitrag zum Gebiet der Ähnlichkeit von kleinen Molekülen. Diese Weiterentwicklung von induktivem Transfer und Distanzlernen ist, genau wie der induktive Transfer selbst, besonders in Situationen nützlich, in denen man nur eine begrenzte Menge an Trainingsdaten zur Verfügung hat, jedoch ein verwandter Datensatz verfügbar ist. Der verwandte Datensatz kann dann als zusätzliches Wissen im Lernprozess genutzt werden. Die Auswahl der verwandten Datensätze wird entweder über Expertenwissen oder durch einen halbautomatischen Ansatz bewerkstelligt. Die Arbeit diskutiert auch mögliche Synergien die mit Kombinationen der drei vorgestellten Ansätzen erzielt werden können. Falls die Konzepte, die in dieser Arbeit vorgestellt werden, ihren Weg in die Praxis finden, können sie bei der Entwicklung neuer Medikamente helfen, da sie den Entwicklungsprozess sowie den Zulassungsprozess von Arzneimitteln beschleunigen können oder sie können sogar die Leben von Versuchstieren retten, da verbesserte computergestützte Toxizitätsmodelle Tierversuche zunehmend überflüssig machen.

CHAPTER 1

Introduction

The work presented in this dissertation has the goal of enhancing similarity-based applications in predictive toxicology, (Quantitative) Structure Activity Relationships and cheminformatics in general. This is achieved through enhancements to the central concept of small molecule similarity such as incorporation of background knowledge, (optimally) combining different complementary similarity measures or using similarities with respect to defined reference structures in descriptor space. The resulting similarity measures are applied in classification, regression and similarity ranking (virtual screening) scenarios.

In the first chapter I motivate the thesis, briefly introduce the basic concepts of (Q)SAR (including the applicability domain) as well as virtual screening and give an outline of the overall thesis. All datasets that were used for computation in this thesis are publicly available sets of small molecules. I use the term *small molecules* as it is widely used in pharmacology: denoting non-polymeric, organic molecules of low molecular weight (approximately below 800 Daltons). Note that most marketed drugs fall into the category of small molecules, although protein drugs like insulin exist. Roughly speaking, the way small molecular drugs work can be described as follows: In order to cause a biological effect in the target organism, the small molecule binds to the drug target. This can, for example, be an enzyme, and usually the drug target is part of or at least connected to the biological pathway associated with the disease to be addressed. Consequently, the biological activity of the target is altered in a way favorable for the patient. Examples for small molecular drugs are the drug class of statins. Statins are used to lower the cholesterol levels in the body by inhibiting the HMG-CoA reductase, a key enzyme in the endogenous cholesterol production pathway. This inhibition then leads to a decrease in the cholesterol level.

1.1 Motivation

In 2010, Paul *et al.* [100] reported that the productivity in pharmaceutical research and development (R&D) has to be drastically improved. Not only did cost estimates for the development of new molecular entities (NMEs, also known as new chemical entities (NCEs)) rise, also the numbers of innovative drugs approved by the US Food and Drug

Administration (FDA) decreased over the last five years [85]. The capitalized cost for a new NME is estimated with \$1.8 billion, with out-of-pocket costs of \$870 million. For the pharmaceutical industry, as it exists today, this is a great challenge that has to be faced and solved in order to sustain the current business model. I am sure that apart from optimizing the most expensive parts of drug development, the clinical phases II and III (according to Paul *et al.*), computational methods applied in all stages of the drug discovery process will enhance the productivity of R&D [83, 91]. Nowadays, computational approaches in drug discovery and design more than ever promise time and cost savings in practically all pre-clinical phases. This is made possible by the great advances in predictive sciences and computer technology and by the rising interest and research efforts in those areas.

With this context in mind, the research presented in this thesis revolves around enhancements to chemical similarity and methods dependent on chemical similarity. Although there are opinions stating, that with improved understanding of the chemistry and biology of drug action and a greater ability to model the underlying mechanisms, the need for similarity approaches will diminish [7], I think that for the time being and probably the next ten or twenty years at least, the concept of molecular similarity is central to (Q)SAR, ADMET (Absorption, Distribution, Metabolism, Excretion and Toxicity) predictions, predictive toxicology, *in silico* drug discovery and cheminformatics. This is emphasized by a quote of Nobel laureate Sir James W. Black (Nobel Prize in Physiology or Medicine 1988): “The most fruitful basis for the discovery of a new drug is to start with an old drug” [103]. This can be interpreted as an advice to use the similarity to existing drugs to find starting points for discovering new drugs. This does not necessarily account for neglected or “new” unsolved diseases, but certainly for the well-studied ones. In addition, one can mimic not only existing drugs, but also natural ligands and inhibitors of drug targets.

Similarity measures for small molecules are used in such versatile application areas as clustering [6, 120, 121], learning of prediction models [2, 15, 61], drug repositioning or repurposing [1, 30, 160], similarity searching [152] or virtual screening [106, 112]. This thesis focuses on two of them: virtual screening in the form of similarity ranking and learning of prediction models, where I distinguish between classification and regression approaches.

1.2 Quantitative Structure Activity Relationships

Quantitative structure-activity relationships (QSARs) and quantitative structure-property relationships (QSPRs) are models which quantitatively correlate chemical structure with biological activities, chemical reactivity or certain molecular properties. In technical and statistical terms, QSARs are regression models on graphs (molecular structures being modeled as graphs), while QSPRs are classification models. Throughout the remainder of this thesis I will not distinguish between QSARs and QSPRs, but rather use only the term QSAR, and where necessary, classification or regression instead.

Often, the Hammett Equation for the reaction constant ρ is mentioned as the first QSAR equation [50]:

$$\rho\sigma = \log K_{R-X} - \log K_{R-H}, \quad (1.1)$$

where ρ and σ are constants for arbitrary chemical reactions with equilibrium constants K . $R-X$ and $R-H$ symbolize substituted and unsubstituted aromatic compounds. Classical QSARs as published for example in a textbook by Hansch and Leo [51] are usually quite simple equations with only few molecular features considered and developed only on few compounds for a very specific problem. Several QSARs from the domain of metabolism like equations for Cytochrome P450 binding, microsomal oxidation or inhibition or glucuronidation are listed by Hansch and Leo. Overall, the book provides more than 6000 of those equations. But due to the very small training datasets the possible applications to new drugs and substances are limited. The rapid development in high-throughput wet-lab experiments (using lab robots), computer hardware and machine learning and data mining research over the last ten to twenty years has changed this and produced not only massive amounts of data, but also technical possibilities to process them. Modern QSAR models are developed with powerful algorithmic methods from the domain of machine learning, like random forests[84], neural networks[102] or support vector machines[56], and are – not always – based on vast numbers of parameters and descriptors. They are also developed from much larger sets of training compounds and can be validated on larger sets of test or validation compounds. This increased number of considered compounds enables us to make a statistically more sound evaluation of the performance and reliability of the trained prediction models and it makes the generated models applicable to a broader set of unknown drug candidates and chemicals.

Naturally, QSARs and other problems considering small molecules as input instances are the subject of very active research in machine learning and data mining research [60, 14, 86, 123], as well as in the more obvious field of cheminformatics or computational chemistry [8, 110, 157]. A very recent and practical project that is concerned with QSAR datasets, descriptors and prediction and validation algorithms is the OpenTox [45, 52] EU FP7 project that is explained in greater detail in Chapter 8. The main goal of the project is to provide an open and extensible framework for QSAR and toxicity data. It provides an Application Programming Interface (API) based on REST webservices[107] to handle the data, descriptor calculation and learning algorithms, the resulting prediction models, reporting and validation procedures as well as visualization tools.

1.2.1 Applicability Domain

In this section I briefly discuss the concept of *applicability domain* (AD)[69, 92] estimation that is closely related to QSAR research and will come up frequently in the QSAR related literature. Informally, the AD of a (QSAR) prediction model can be described as a measure or statement telling the user of a model if a compound can be reliably predicted with that model. In other words, it tells the user if the prediction model and the compound to be

predicted fit together. Basically, the AD can be estimated based on the model input, the model output or both. To the best of my knowledge, no established precise definition of the concept of AD exists.

The OECD mentions the necessity of giving an estimate or a quantification of the applicability domain in their “Guidance Document on the Validation of (Quantitative) Structure-Activity Relationships [(Q)SAR] Models” [95]. Chapter 4 of the document gives guidance on the principle of a defined domain of applicability and provides information on the OECD Validation Principle 3 that a (Q)SAR should be associated with “a defined domain of applicability”. Consequently, the AD of a prediction model is not only of interest to better estimate a prediction’s uncertainty, but also of regulatory interest.

In their review of AD estimation by projection of the training set in descriptor space, Jaworska *et al.* [69] stress that a prediction is only reliable if the model’s assumptions are met. Part of those assumptions is the model’s applicability domain. Looking only at the input side of a model, the AD can be defined as the space that is spanned by the input parameters (physical, chemical, biological and other descriptors as well as other information and knowledge used to train the model). The reviewed methods are of statistical nature and can be summed up in four major approaches that all are used to define the region of interpolation: Range methods, distance-based methods, geometric methods and probability density distribution methods. It is noteworthy that the different interpolation methodologies produce different domains of applicability. In conclusion, Jaworska *et al.* advise practitioners to use structural similarity based AD assessment in combination with one of the reviewed AD estimation methods to obtain a more robust estimation of the chemical space that is valid for model application. This review was also used as a basis for the ECVAM workshop in 2005, where the status of AD methods for QSAR was discussed [92]. Another article that reviews methods for applicability domain estimation is presented by Tetko *et al.* [137]. Two groups of methods are assessed there: molecular similarity based methods and methods based on analyzing calculated properties. It is stressed that the AD also has the function of preventing improper filtering of compounds or compound series in high-throughput screening. If a compound is predicted to be harmful, but outside or even far outside the model’s AD, it can still be considered for further analysis. There have also been recent developments in machine learning research, like for example the work of Buchwald *et al.* on Fast Conditional Density Estimation [14] that automatically provide predictions with confidence intervals. Those confidence intervals can also be seen as an estimate of a prediction lying in the model’s AD or not.

1.3 Virtual Screening

Virtual screening is the computational analogue to the biological or wet-lab screenings, i.e. the classical approach to measuring, ranking and prioritizing databases of molecules [78]. The goal of virtual screening is to score, filter or rank the compounds in a (virtual) compound library to support decisions for further experimental procedures and steps in

the drug discovery process. As the size of the screened database can be quite excessive, it is of high importance that the applied computational methodologies are computationally efficient. A graphical overview of a generic virtual screening work flow is given in Figure 1.1.

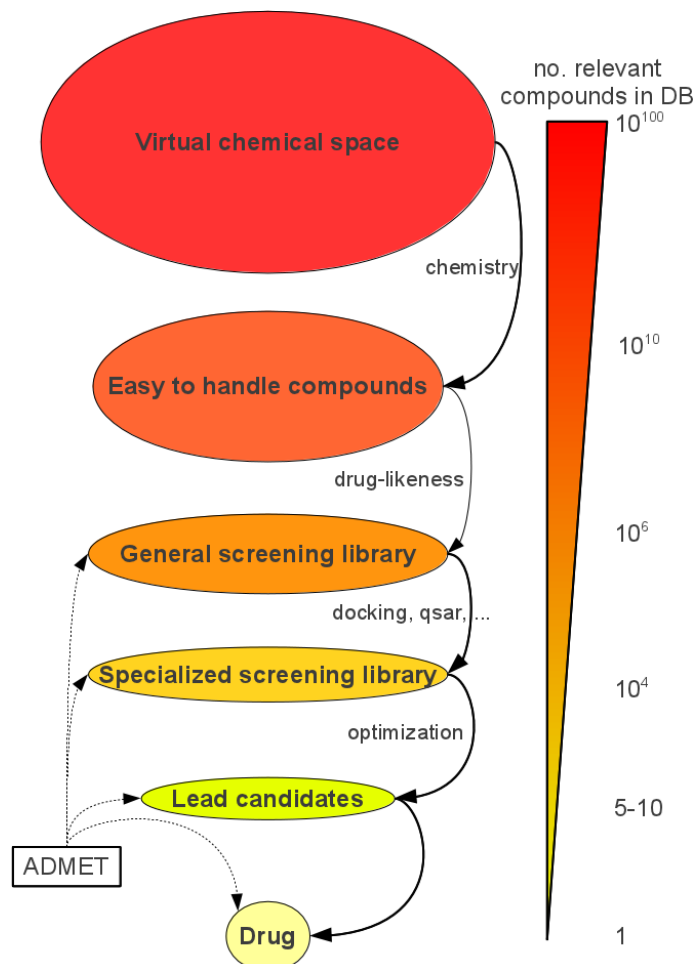


Figure 1.1: Schematic overview of an iterative virtual screening procedure (generic example)

The compounds in the database can be ranked according to different predicted or measured criteria, such as drug-likeness measures [21], e.g. Lipinski's rule of five [82], ADMET properties like solubility, blood brain barrier penetration and intestinal absorption or the occurrence of certain substructures. Wilton *et al.* [155] roughly group the different approaches to virtual screening according to the amount of data that is available into four classes:

- ▷ If only one active molecule is known, a similarity search approach can be used. Here, the database of compounds is ranked in decreasing order according to their similarity to the active molecule.
- ▷ If various active molecules are known, a pharmacophore model can be developed from the common properties of the known bioactive structures. The database of

compounds is then screened for molecules that match this pharmacophore.

- ▷ If it is not possible to design a pharmacophore model, e.g. because the known actives are too diverse and if a sufficient amount of actives is known, machine learning approaches can be used to build prediction models for the biological activity at hand.
- ▷ In the last case, the three-dimensional structure of the target protein has been elucidated and molecular docking can be applied to assess the database with respect to binding-pocket complementarity of the structures.

In practice, often a combination or sequential application of the above virtual screening variants is used. A recent review of trends in ligand-based virtual screening and relevant data mining algorithms has been published by Geppert *et al.* [44].

1.3.1 Evaluation Measures

To evaluate the performance of a virtual screening usually the enrichment factor (EF) [35] is considered. The enrichment factor reflects the amount of known related structures in the first $x\%$ of the ranked database. In practice, often only the highest ranked compounds are of interest and considered further in the drug discovery pipeline. The enrichment factor is defined for certain fractions of the database:

$$EF(\%) = \frac{(N_{active(\%)} / N_{(\%)})}{(N_{active} / N_{all})}, \quad (1.2)$$

where $EF(\%)$ is given for the specified percentage of the ranked database, $N_{active(\%)}$ is the number of active compounds in the selected subset of the ranked database, $N_{(\%)}$ is the number of compounds in the subset, N_{active} is the number of active molecules in the dataset and N_{all} is the number of compounds in the database. For an easier interpretation of the EF values, it is helpful to compare them to the maximum possible enrichment at the specified fraction of the database:

For easier comparison it is possible not to use the $EF(\%)$ directly, but the difference of maximum possible enrichment and achieved enrichment:

$$\Delta_{EF} = EF_{max} - EF(\%). \quad (1.3)$$

Please keep in mind that for Δ_{EF} smaller values are better and the optimal Δ_{EF} is zero.

1.4 Outline of the Thesis

After a motivation and brief introduction to the central concepts QSAR and virtual screening in the first chapter, I review existing similarity measures in computational chemistry in Chapter 2. First, classical two-dimensional similarity measures for small molecules like

fingerprint-based similarity and maximum common subgraph similarity are explained before a short introduction to three-dimensional techniques is given. Third, several cheminformatics applications that rely on or work with the concept of molecular similarity are presented before I talk about similarity-based QSAR in greater detail. Chapter 3 reviews two concepts that are the basic building blocks of our work on transferring and adapting distance measures. The first concept is distance learning. Here, approaches try to learn or adapt a certain distance measure or metric to a certain problem and thus improve it. The second concept, inductive transfer, is used to transfer knowledge, e.g. in the form of an inductive bias, from a related problem to the problem at hand. This is especially useful in cases where I do not have sufficient data on our problem to be able to generalize, but sufficient data for the related task.

The first chapter presenting my own contributions is Chapter 4, which introduces so-called similarity boosted Quantitative Structure-Activity Relationships. Here, I use the similarities of the training and test compounds to selected reference points as molecular descriptors. The reference compounds are selected with three methods: by literature review, by clustering the active structures in the assay or by clustering representatives from the chemical space. I also combine the similarity descriptors with state-of-the-art structural descriptors and show that the combination has improved predictive performance. This shows that the similarity descriptors encode information which is complementary to a certain extent to that encoded by the structural descriptors. Chapter 5 presents an approach that shows that virtual screening based on structural similarity can be significantly improved if background knowledge is incorporated. In a first step, I extract the relevant knowledge from the literature and visual inspection of the molecules. In a second step, I use a simple data mining approach to extract it. The background knowledge is then incorporated into the structural similarity measure in the form of a fingerprint similarity. Chapter 6 introduces adapted transfer of distance measures. This approach is a combination and enhancement of transfer learning and inductive transfer. Instead of only transferring the bias from a related problem to the given task, I additionally adapt it to the task at hand. In a first set of experiments the source datasets (from which the bias is transferred) are selected by hand, in a second set of experiments, a data-driven semi-automatic approach to the selection of the source dataset is used. A software application of QSAR and predictive toxicology algorithms is presented in Chapter 8: the distributed REST webservice-based toxicity prediction framework OpenTox. Finally, I summarize our work and give an outlook to possible extensions and future work in Chapter 9. Additional material for all chapters is presented in the appendix.

CHAPTER 2

Similarity in Cheminformatics

Similarity is in the eye of the beholder.

Many applications and problem settings in cheminformatics rely on the concept of similarity of small molecules. Examples are search functionalities like substructure search, prediction methods like k -nearest neighbors, variants of virtual screening or clustering. This makes molecular similarity one of the most central concepts in cheminformatics [93]. A basic assumption of the field by Johnson and Maggiora [71] is also built on similarity: *similar compounds have similar properties*. Many applications make use of this assumption to predict chemical, physical, biological, toxicological or other properties and functions of uncharacterized molecules. In the case of biological function, two chemical compounds have to be similar in their physico-chemical properties and their structure to be both able to fit into the often very specific active site region of an enzyme and induce or block the biological functionality. An example of three molecules that have highly similar structures and act in the same manner on the enzyme HMG-CoA reductase can be given with the structures fluvastatin, atorvastatin and pitavastatin (see Figure 2.1). HMG-CoA reductase is the rate-controlling enzyme of the mevalonate pathway. Its inhibition indirectly leads to a decrease of the plasma concentration of cholesterol and as such it is a target for cholesterol-lowering drugs like the three shown statins. In the remainder of this chapter,

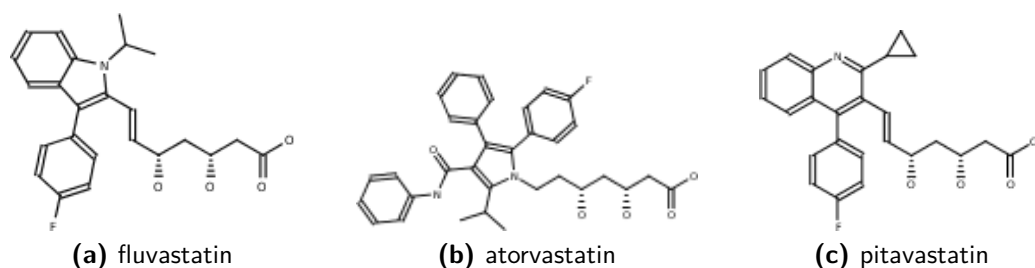


Figure 2.1: Compounds from the family of statins inhibiting the enzyme HMG-CoA reductase. They are marketed drugs used to lower cholesterol levels.

I will give a brief overview of similarity measures used in cheminformatics. I distinguish between fingerprint based methods and methods that try to align or map two molecules to obtain a meaningful similarity value. Then I will introduce some work on molecular similarity based on three dimensional data, before I discuss some cheminformatics applications that use similarity to improve predictions, performance or other evaluation measures. There are also approaches to small molecule similarity with kernel methods [16, 133], but to the best of our knowledge only in an indirect way. All relevant studies use the kernels directly for prediction and do not use, interpret or analyze the similarities represented by the kernel matrix. Consequently, I do not discuss these approaches in detail this thesis.

Relation to Distances and Kernels Due to conceptual and probably also historical reasons – looking for molecules that have similar properties – researchers in cheminformatics most of the time use the concept of similarity. A concept closely related to similarity is distance. Distance is a measure of dissimilarity. A distance function $d(x,y)$ is a metric, if it satisfies the following four conditions:

- ▷ *non-negativity*: $d(x,y) \geq 0$
- ▷ *identity of indiscernibles*: $d(x,y) = 0$, iff $x = y$
- ▷ *symmetry*: $d(x,y) = d(y,x)$
- ▷ *triangular inequality*: $d(x,z) \leq d(x,y) + d(y,z)$ [see Figure 2.2]

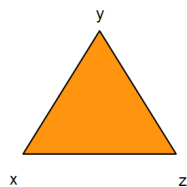


Figure 2.2: Triangle illustrating the triangular inequality of a metric.

There is no such strict definition for similarity functions. Often similarity measures are considered the inverse of distances, but mathematically no such definition or rule exists. Generally one can say that small distances correspond to large similarities and large distances correspond to small similarities. However, in cases where the similarity measure $s(x,y)$ is normalized to $[0,1]$ the equation $s(x,y) = 1 - d(x,y)$ is valid [18].

A kernel function $K(x,y)$ maps two objects to a real valued number, with the function being symmetric and positive-semidefinite. Balcan and Blum [3] mention that many kernels (Gaussian kernel or Fisher kernels [66]) describe notions of (dis-)similarity between objects. Consequently, the kernel matrix can be interpreted as (dis-)similarity matrix.

2.1 Molecular Similarity in 2D

2.1.1 Fingerprint Based Similarity

Using so-called fingerprints is the classical way in cheminformatics to assess the degree of similarity of two molecules. The fingerprint similarity calculation is a two-step process: First, a numerical or binary vector representation of the molecules is created and second, a similarity coefficient is used to calculate the similarity of the two vectors. In a cheminformatics context, the term *fingerprint* is usually used instead of the term *vector*. A graphical example of how a binary fingerprint based on substructural fragments is built, is given in Figure 2.3a.

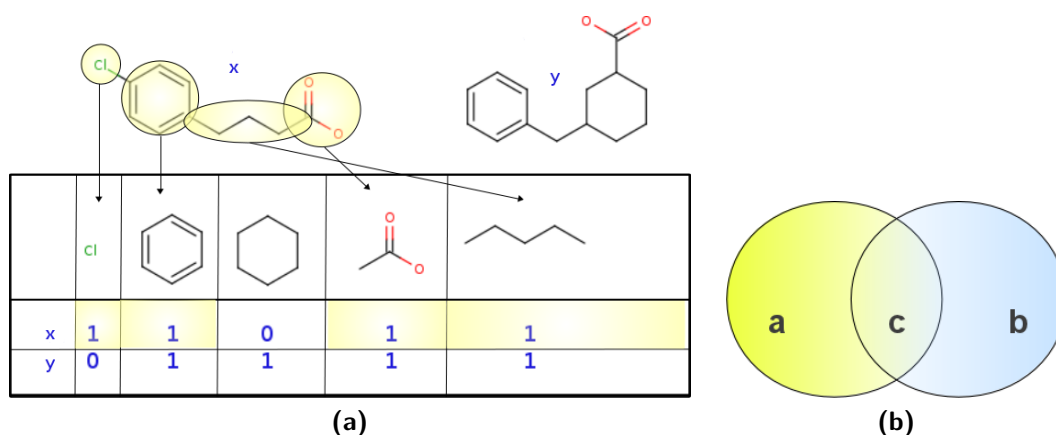


Figure 2.3: (2.3a) Molecular fingerprint example. If a fragment occurs in molecule x or y , the value 1 is assigned, 0 otherwise. (2.3b) Venn diagram visualizing fingerprint overlap used in similarity coefficients.

Usually, fingerprints are several hundred or a few thousand bits long. Established examples of fingerprints are the 920 bit MACCS keys [31] or the daylight fingerprints¹. An advantage of using fingerprints is the quite compact representation that enables highly efficient and fast comparisons rendering fingerprints a good method for processing large compound databases. Clear disadvantages are that dissimilar structures can have an identical fingerprint (if not enough discriminating bits are defined) and that usually only the occurrence of fragments is tested and not the number or the position of the fragment in the molecular graph. The Tanimoto coefficient [135] is the most common similarity coefficient used in the cheminformatics community to calculate the similarity between two molecular fingerprints. It is defined as follows:

$$sim_{Tanimoto}(x,y) = \frac{c}{a + b - c}, \quad (2.1)$$

where a and b is the number of bits set to 1 in molecules x and y , and c the number of bits set to 1 in both, x and y . A Venn diagram visualization of the relationship of a , b

¹ <http://www.daylight.com/dayhtml/doc/theory/theory.finger.html>

and c is given in Figure 2.3b. A second, very popular similarity coefficient for fingerprint similarity calculations that is related to the Tanimoto coefficient, is the Dice coefficient:

$$sim_{Dice}(x,y) = \frac{2c}{a+b}, \quad (2.2)$$

where a , b and c are defined as in the Tanimoto coefficient. Note, that the Dice coefficient is monotonic with the Tanimoto coefficient. An example for an asymmetric similarity is the Tversky index [140]. It is a generalization of the Tanimoto and Dice coefficient and defined as:

$$sim_{Tversky}(x,y) = \frac{c}{\alpha(a-c) + \beta(b-c) + c}, \quad (2.3)$$

where a , b and c are defined as above and α and β are user-defined constant values with $\alpha, \beta \geq 0$. Please note that if $\alpha = \beta = 1$, the Tversky index is equal to the Tanimoto index and equal to the Dice coefficient, if $\alpha = \beta = \frac{1}{2}$.

If the fingerprint is not binary but constructed from numerical molecular descriptors (e.g. logP, molecular weight, ...), metrics like the Euclidean distance

$$sim_{Euclid}(x,y) = 1 - \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2.4)$$

or a continuous variant of the Tanimoto coefficient can be used to calculate the similarity of two molecules x and y . For further information about chemical descriptors see Gasteiger [42] or Todeschini and Consonni [138]. For a more detailed introduction to basic molecular similarity techniques and coefficients I refer the interested reader to the cheminformatics text book by Leach and Gillet [78].

2.1.2 Maximum Common Substructure Based Similarity

As an example of a similarity measure that aligns or maps two molecular structures to calculate a similarity value, I chose similarity based on the maximum common substructure/subgraph (MCS – e.g., Raymond *et al.* [104]) of two molecules. Please keep in mind that the MCS problem is known to be NP-complete [40] and as such expensive to calculate. Only the small and sparse nature of small molecule graphs makes MCS approaches possible in large and medium scale cheminformatics applications. After calculating the size of the MCS of two structures, their similarity can be calculated, for example, with the measure proposed by Wallis *et al.* [144]:

$$sim_{MCS}(x,y) = \frac{|mcs(x,y)|}{|x| + |y| - |mcs(x,y)|}, \quad (2.5)$$

where $|\cdot|$ gives the number of vertices in a graph, and $mcs(x,y)$ calculates the MCS of molecules x and y . A visualization of a maximum common substructure is given in Figure 2.4.

Before I give an overview of similarities in the three dimensional space, I report on an

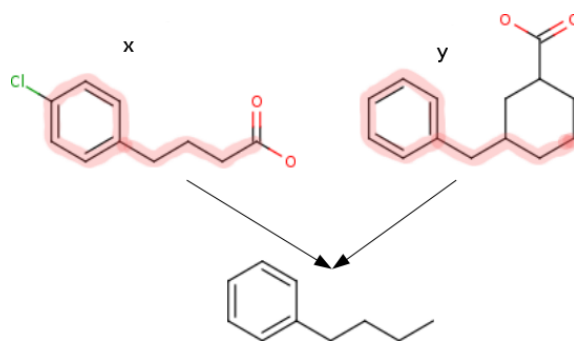


Figure 2.4: Maximum common substructure of molecules x and y .

important observation by Raymond and Willett [105]: In their study, the authors use the experimental setting of virtual screening of the MDDR and ID Alert databases. They find, that the two big families of 2D chemical similarity measures, fingerprint-based and maximum common substructure based measures, provide orthogonal information about chemical similarity. This is a very important finding which shows that representing a molecule only by its physical and chemical properties leads to a loss of information and is in some problem settings or applications insufficient. Also, this rationale can be used to develop more effective and accurate measures for small molecule similarity.

2.2 Three Dimensional Similarity Measures

There has always been high interest in similarity methods based on three dimensional data as molecular recognition depends on the molecular conformation in three dimensional space. It is possible to align graph structures in 3D to a sufficiently high extent, although they are dissimilar in 2D. The latter suggests that methods considering only two dimensional input data can miss important similar structures that are only recognizable with three dimensional information. In general, methods for 3D molecular similarity are either independent of the orientation of the molecules or the methods need to align the molecules in 3D space before calculating their similarity.

Orientation Independent Techniques Pepperrell and Willett [101] evaluate several orientation independent techniques for the calculation of molecular similarity in three dimensions. In their study, the molecules are represented as inter-atomic distance matrices.

The first presented technique is denoted *Distance Distribution* method. It approximates the overall topology of a structure through the inter-atomic distance distribution. To obtain the distribution, all inter-atomic distances of molecule A are sorted into bins of size R . This user-defined parameter R can be set to, for example, 0.5\AA . The similarity is then calculated by comparing the distance distributions FA and FB of molecules A and B . If only a single distribution is used to describe a molecule, different inter-atomic distances are assumed to be of equal length, regardless of their atom types. A more realistic and useful

scenario is to use different distributions for carbon-carbon (CC), carbon-heteroatom (CX) and heteroatom-heteroatom (XX) distances. The authors define the degree of similarity between molecules A and B as:

$$\alpha G(FA_{CC}, FB_{CC}) + \beta G(FA_{CX}, FB_{CX}) + \gamma G(FA_{XX}, FB_{XX}), \quad (2.6)$$

where α , β and γ are user-defined weights and G is some goodness-of-fit criterion.

The second presented technique is called *Individual Distances* method. It is based on the number of identical components (n_c) in the molecules to be compared. In this case identical means that the distances are equal with a tolerance of $\pm 0.5\text{\AA}$ and that the element types are also identical. The overall similarity of molecules A and B is then calculated with the Tanimoto coefficient:

$$sim = \frac{n_c}{n_A + n_B - n_c}, \quad (2.7)$$

where n_A and n_B are the number of inter-atomic distances for molecules A and B .

The third technique presented by Pepperrell and Willett is the *Atom Mapping* method. This method tries to map atoms from molecule A to atoms of molecule B that are most similar to them. First, a $n_A \times n_B$ atom match matrix S is created, by comparing single atoms from molecule A to those of molecule B using equation (2.7). Subsequently, a matching algorithm is applied to find the best matching atoms in molecule B for the atoms in molecule A . The overall similarity is then calculated as the sum of all similarities over all atoms in molecule A .

The fourth method presented by the authors is the *Maximum Common Substructure* method. Here, the MCS is defined as the largest pattern of atoms in 3D space that is isomorphic, i.e., structurally identical (keeping the tolerance of $\pm 0.5\text{\AA}$). The similarity of A and B is then calculated as given in the two dimensional MCS case (2.5).

In their experimental evaluation of the four presented similarity methods, the authors use 10 small datasets (109 to 209 instances) for which they have both 3D coordinates and biological activity data. They calculated similarity rankings (in descending order) for each dataset, using an active compound as query molecule. This procedure is repeated for all active compounds. Only a single conformation per molecule is considered – the lowest energy conformation. The authors find that the Distance Distribution method and the Individual Distances method perform comparably, but clearly inferior to the Atom Mapping method. The MCS method is sometimes better than the Atom Mapping method, but is the computationally most demanding one. Additionally, the authors note that they were not able to show that 3D methods are superior to 2D methods, although they render different sets of resulting similar structures. This might be overcome by using not only one molecular conformation, but a set of molecular conformations or by allowing for structure flexibility.

In a more recent study, Kim *et al.* [73] present several 3D structure similarity measures

available under PubChem3D². The first one is a shape-Tanimoto ST :

$$ST = \frac{V_{AB}}{V_{AA} + V_{BB} - V_{AB}}, \quad (2.8)$$

where V_{AA} and V_{BB} are the self-overlap volumes³ [74] of conformers A and B and V_{AB} is the overlap volume of A and B . The second measure presented by the authors quantifies the similarity based on the three-dimensional orientation of six functional groups (hydrogen-bond donors and acceptors, cation, anion, hydrophobes and rings):

$$CT = \frac{\sum_f V_{AB}^f}{\sum_f V_{AA}^f + \sum_f V_{BB}^f - \sum_f V_{AB}^f}, \quad (2.9)$$

where f indicates any of the six feature atom types, V_{AA}^f and V_{BB}^f are the self-overlap volumes for feature atom type f and V_{AB}^f is the overlap volume of conformers A and B for feature atom type f . The two measures ST and CT are also combined to $ComboT = ST + CT$. In their study, the authors try to answer the question, what a biologically meaningful similarity value is and analyze (all-against-all) a set of 734,486 biologically tested compounds (multiple conformers) from 1,389 biological assay datasets. They calculate mean and standard deviation for the mentioned similarity measures to create a statistical framework to build upon. In the second part of their study, the authors try to answer the question if the presented similarity measures are able to separate the “actives” and “inactives” of the biological assays. In this study they find that $ComboT$ is the most efficient 3D score type. However, considering only one conformation per molecule in the second study, they were only able to separate a small number of assays in “active” and “inactive” compounds. The authors say that this is due to different underlying mechanisms of action or binding configurations. As mentioned in the discussion of the work by Pepperrell and Willett, it would be interesting to see if such problems reported with three dimensional similarity measures can be circumvented or even solved by implying either structure flexibility directly, or by implying several conformations of the molecular structure.

Alignment Techniques Generally speaking, methods that try to superposition or align three dimensional molecules are computationally more demanding than orientation independent techniques, as they have to consider more degrees of freedom. This is why I only give a very brief overview on them in this thesis.

Lemmen and Lengauer [79] review methods for aligning molecules. The first discussed method is *rms-fitting* [72] plus its extension *directed tweak* [63], which is rms-fitting while also considering the flexibility of the input molecules. To align two molecules, the sum of

² <http://pubchem.ncbi.nlm.nih.gov/vw3d/vw3d.cgi?>

³ The conformer monopole volume (V) and three components of the shape quadrupole moments (Q_x , Q_y , and Q_z , which give a sense of the conformer length, width, and height dimensions, respectively)

squared differences of corresponding points is minimized. The second method is *volume overlap optimization*, which comprises three basic steps: (1) decompose the molecules into spheres, (2) generate a sample of starting configurations and (3) do a local optimization using the corresponding normalized similarity indices. *Geometric hashing* [77], a technique originally used in computer vision, is a two step technique: (1) highly redundant representations that are invariant under rotation and translation, are generated for the first molecule and stored in a hash table before (2) the hash table is queried with molecular features from the second molecule. Hits in the hash table correspond to transformations between the molecules. Lemmen and Lengauer also report on approaches based on *clique detection* [77] and *distance geometry* [22].

2.3 Cheminformatics Applications Relying on Molecular Similarity

As the concept of molecular similarity is such an abundant topic in cheminformatics it is nearly impossible to give a sufficiently large selection of examples within the scope of this thesis. Nevertheless, I discuss a small selection of two approaches in different areas of cheminformatics that rely heavily on the concept of molecular similarity, just to give a very brief impression of what has been done to date and how differently the concept can be used.

The first example is the usage of small molecule similarity for clustering in combination with a local QSAR modeling approach. The first work mentioned in this context is that of He and Jurs [54], who use the concept of molecular similarity to assess the reliability of QSAR predictions. Their working hypothesis is that if a query compound is more similar to the compounds used to train the QSAR model, the prediction should also be more accurate. This is a concept closely related to the estimation of the applicability domain. The authors employ hierarchical clustering to form dissimilar clusters of molecules. For each cluster, a QSAR model is learned and the query compounds are then predicted with all learned models. The authors then correlate the similarity of the query compounds to the clusters with the resulting prediction accuracies. For the applied dataset of 322 organic compounds with fathead minnow acute toxicity, the authors can show a direct relationship between the similarity of the query compound and the training set and the achieved prediction accuracy. Consequently, the similarity value can be used to assess the reliability of the prediction. The second work I want to mention in this context has been published recently by Buchwald *et al.* [15]. The authors present an approach that makes use of the structural similarity of molecules to improve the predictivity of QSAR models. In a first step, the authors apply an online structural clustering procedure [121, 120] that groups compounds based solely on their structural similarity. This similarity is calculated from the size of the structural overlap, comparably to the MCS similarity, but computationally more efficient. To compute the structural overlap, the authors use a modified version of the graph mining algorithm gSpan [161, 67] with the minimum frequency constraint set to 100%. Formally,

the clustering criterion is then defined as:

$$\exists s \in cs(\{x_1, \dots, x_m\}) \forall x_i \in C : |s| \geq \theta |x_i|, \quad (2.10)$$

where s is a subgraph, $C = \{x_1, \dots, x_m\}$ a cluster, $X = \{x_1, \dots, x_n\}$ a set of molecules and $\theta \in [0,1]$ a user-defined similarity coefficient. Based on the calculated sets of similar compounds, local QSAR models are learned in a second step. That means that for every cluster found in step one that fulfills a defined size threshold *minClusterSize*, a QSAR model is learned. Basically, the users are free in their choice of learning algorithms and molecular descriptors that they want to apply. In addition to the local models a global model using all training instances is learned that acts as a default or fall-back model.

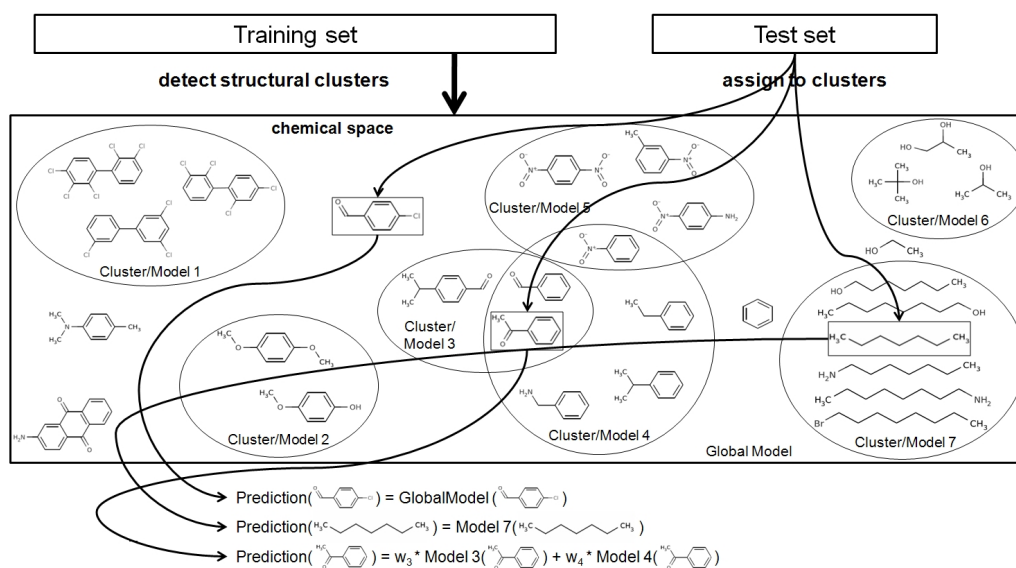


Figure 2.5: Schematic overview of the local model approach.

During the prediction phase, the query compound is assigned to zero, one or more clusters (using the same similarity criterion as above) and predicted with the corresponding model(s). If no cluster is assigned, the corresponding model is the default model trained on the complete training set. This is the same procedure that would be applied in non-local QSAR modeling. Figure 2.5 shows a schematic overview of the overall workflow of the approach. The results presented by the authors indicate that the predictive power is – in most cases – statistically significantly improved compared to an approach based on fingerprint clustering, on hierarchical clustering and compared to locally weighted learning as implemented in the WEKA workbench [49, 38]. Drawing a conclusion, one can say that in the work of Buchwald *et al.*, the structural similarity was successfully applied to improve QSAR predictions.

Another application example that relies on molecular similarity is the work by Wang *et al.* [145]. The authors present an approach to accurate similarity search in large compound databases using graph kernel methods. The motivation behind this approach

is that public compound databases like PubChem [11] have grown rapidly over the last years. PubChem⁴ now contains over 32 million compounds⁵, while Wang *et al.* reported 18 million compounds in 2010. Naturally, methods for efficient searching in those large databases of structure data are of high interest. The default choice for similarity searching are methods based on high-dimensional fragment based fingerprints, e.g. the Daylight fingerprints⁶ in combination with a Tanimoto coefficient. An alternative are maximum common subgraph based methods or other graph based methods. Those, however, are usually slow and only useful for smaller databases. The goal of Wang *et al.* is to bridge the gap between graph kernel functions and similarity search to provide an efficient graph based search method. Their search method is built on previous work for general graphs, called *G-hash* [146]. G-hash first extracts node features and local features for both, the graphs in the database and the query graph. For the features generated from the database an index structure, in this case a hash table, is built. Using the sets of features, a distance is calculated and the k nearest neighbors of the query compound are reported. The node (atom) features used in this application are: atomic number, the histogram of atom types of immediate neighbors of the atom, the local functional group information, and the histogram of the immediate bond information. In the applied hash table, the key is the related node feature vector and the value is the node. Consequently, two chemical compounds are regarded similar, if they share a lot of nodes in the same hash cell. The authors constructed the following kernel function to measure the graph similarity between two graphs G and G' :

$$K_m(G, G') = \sum_{(u,v) \in V[G] \times V[G']} K(\Gamma(u), \Gamma(v)), \quad (2.11)$$

where K can be any kernel function defined in the co-domain of Γ . This function is revised keeping in mind that the feature vectors are discretized and a hash table is used. The resulting kernel function is:

$$K_m(G, G') = \sum_{v \in G'} |simi(v)|, \quad (2.12)$$

where $simi(v)$ are the nodes from G that are hashed to the same cell as the node v . Paraphrasing this concept, the number of common nodes in G and G' are counted.

The authors state that the k nearest neighbors retrieved with the G-hash approach are more likely similar to the query compound than those retrieved by Daylight fingerprints or C-tree. In addition, the approach is more scalable in that it provides faster index construction and faster querying. Another recent approach which focuses on kernel-based similarity search in chemical compound databases which is not discussed in detail in this

4 <http://pubchem.ncbi.nlm.nih.gov>

5 19.02.2012

6 <http://www.daylight.com>

section, is a wavelet tree based approach presented by Tabei and Tsuda [134]. Apart from similarity search, also substructure search approaches to querying large compound databases exist [46, 162, 122]. The problem setting here, however, is a little bit different: The search methods retrieve all graphs containing the query as a subgraph.

2.4 Similarity based QSAR

The basic idea underlying similarity-based QSAR approaches was enunciated explicitly by Johnson and Maggiora, who postulate that molecules that are structurally similar, will likely have similar properties [71].

Cuadrado *et al.* [23] present an approach to QSAR based on similarity used as descriptors. They predict a set of 31 steroids using PLS regression as learning algorithm and an index called Approximate Similarity (AS) as descriptor space. The AS measure is based on structural similarity that is refined with a dissimilarity measure based on non-isomorphic patterns. In greater detail, the similarity part of the Approximate Similarity $S_{A,B}$ is based on the graph isomorphism of the two input molecules A and B and calculated with a maximum common subgraph (MCS) algorithm and the cosine index. The dissimilarity part is based on the structural difference $\Gamma_{A,B}$ of the input graphs G_A and G_B :

$$\Gamma_{A,B} = g [td(N_A), td(N_B)], \quad (2.13)$$

where N_A and N_B represent the subgraphs that do not form the isomorphism (uncommon fragments), g is a distance function like, e.g., the Euclidean distance and $td(\cdot)$ is a topological descriptor which describes the non-isomorphic fragments. An examples for such an index is the Wiener index [151]. The overall AS is then calculated as follows:

$$AS_{A,B} = S_{A,B} - w_\Gamma \bar{\Gamma}_{A,B}, \quad (2.14)$$

with $\bar{\Gamma}_{A,B}$ being the scaled version of $\Gamma_{A,B}$ and w_Γ being a weighting factor. An all-against-all AS matrix is used as descriptor set. As this index is based on MCS calculations, an application to larger dataset may be impractical because of immense computational costs due to the NP-hardness of the problem [40]. Cuadrado *et al.* report the quality of their learned PLS regression models with Q^2 values between 0.71 and 0.84, the best model being evaluated on the complete test set (no outliers removed) achieving 0.77. Later, Ruiz *et al.* [116] also applied this methodology to a dataset of 30 anti obesity drugs.

Richter *et al.* [108] show that they can significantly improve the regression mean absolute error for growth inhibition prediction on NCI DTP human tumor cell line screening data by using background knowledge. The background knowledge in this case consists of structures and a mode of action grouping of standard anti-cancer agents (ACAs). This information is encoded and added to the description of the molecules in terms of similarities of the training structures with respect to those ACA reference structures or with respect to the groups of modes of action. The NCI DTP human tumor cell line screening data that was used comprises of 42,247 chemical structures⁷ and their corresponding GI_{50} values (measure of growth inhibition). For the set of 107 ACAs⁸ overlapping with the NCI

⁷ <http://dtpsearch.ncifcrf.gov/FTP/CANS03SD.BIN>

⁸ http://dtp.nci.nih.gov/docs/cancer/searches/standard_agent.html

cancer dataset also the mechanism of action is known. The mechanisms can be divided into 6 groups. The molecular structures are encoded by substructure occurrence fingerprints calculated with the Free Tree Miner software [113]. The similarities are calculated by using those fingerprints and one of three different similarity coefficients (Tanimoto, Cosine, Kulczynski). To get a similarity of a compound with respect to a group of compounds with known mechanism of action, three variants were examined: (1) similarity to the closest group member, (2) mean similarity and (3) similarity to the closest and furthest group member. The best performance that was achieved in the experimental evaluation of the approach (60/40% hold-out split evaluation), is based on a balanced set of fragments generated from the training compounds and the ACAs. The machine learning algorithm Cubist was applied for learning. The authors report a statistically significant improvement of the mean absolute error over the reference experiments. The results are also an improvement to results reported in previous work [109]. The usage of the additional group information on the mechanisms of action further improved performance.

From a machine learning perspective, a prominent example of using similarities in the descriptor space is the concept of empirical kernel maps [139, 119]. With the empirical kernel map approach, a similarity function can be transformed into a valid kernel. To achieve this, a similarity vector is used to represent an instance in the training set. The kernel itself is then defined as the dot product of two similarity vectors. In the original publication of Tsuda from 1999, the concept is introduced as an extension of the Support Vector Machine and without the explicit name *empirical kernel map*. The difference to the “classical” SVM lies only in the feature extraction part. The classifier (the optimal hyperplane in the feature space) remains the same. More formally, this means if \mathcal{X} is a set and $s : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ a measure of similarity, an object $\mathbf{x} \in \mathcal{X}$ can be represented by a similarity vector:

$$\varphi(\mathbf{x}) = (s(x, t_1), \dots, s(x, t_r))^T, \quad (2.15)$$

where $t_1, \dots, t_r \in \mathcal{X}$ are called templates. The kernel formulated via the dot product then looks like this:

$$\forall \mathbf{x}, \hat{\mathbf{x}} \in \mathcal{X}, \quad k(\mathbf{x}, \hat{\mathbf{x}}) = \varphi(\mathbf{x})^T \varphi(\hat{\mathbf{x}}) = \sum_{i=1}^r s(\mathbf{x}, t_i) s(\hat{\mathbf{x}}, t_i). \quad (2.16)$$

This valid kernel can be plugged into an SVM and used to build prediction models based on the implied similarity measure. Note that usually all instances in the training set are used as templates and the resulting kernel is based on an all-against-all similarity matrix.

CHAPTER 3

Distance Learning and Inductive Transfer

QSAR modeling and related problems in predictive toxicology are often tackled by instance-based and distance-based methods, which predict biological activity based on the (dis-)similarity of structures. As the success of those methods critically depends on the availability of a suitable distance measure, it would be desirable to automatically determine a measure that works well for a given dataset and endpoint. In the following, we first describe methods from the literature that directly learn distance measures from the data before we discuss inductive transfer approaches. Here, a learning bias is transferred from a related learning task to the learning problem at hand. This is especially interesting in cases where only few data is available to train a learner, but sufficient or additional data on related problems. In this chapter we will, for convenience reasons, mostly not talk about the concept of similarity, but about distance or dissimilarity measures. However, as Sippl [124] shows, and as is intuitive, similarity and distance are just two sides of the same coin. If we have the distance there are well known ways to transform one into the other [18]. In an optimization setting, for example, one can either minimize the distance or maximize the similarity between a set of points, to get information about the most related neighbors of a certain instance.

3.1 Distance Learning

In distance learning, the distance measures (e.g., parameterized distances like the Mahalanobis distance) are directly learned from labeled training instances [149]. That makes it possible to, e.g. improve predictive accuracy or clustering results that rely on the choice of an adequate similarity or distance measure. In this section we are introducing several methods from the literature that are concerned with learning such a measure directly from data.

In their work from 2005 Goldberger *et al.* [48] present an approach to Mahalanobis distance measure learning called *Neighbourhood Components Analysis* (NCA). They experimentally assess their method plugged into the k -nearest neighbor (k NN) classification algorithm. Goldberger *et al.* motivate the NCA approach by pointing out a drawback of

the k NN method: How should the distance metric used to define the k “nearest” neighbors be defined? The authors try to solve this issue by learning a quadratic distance metric. This metric optimizes the expected leave-one-out (LOO) error of the classifier on the training set. In more detail, n real valued input data vectors x_1, \dots, x_n in \mathcal{R}^D with corresponding class labels c_1, \dots, c_n are used. By restricting themselves to learning Mahalanobis metrics, the authors ensure that the metric can always be represented by symmetric positive semi-definite matrices. With that, a linear transformation of the input space (denoted by matrix A) can be learned, such that the k NN classifier performs well in the transformed space. To overcome the discontinuous nature of the actual LOO error function, a differentiable cost function is introduced that is based on a stochastic neighborhood assignment. The probability p_{ij} that a point i selects a point j as its neighbor is given by:

$$p_{ij} = \frac{\exp(-\|Ax_i - Ax_j\|^2)}{\sum_{k \neq i} \exp(-\|Ax_i - Ax_k\|^2)}. \quad (3.1)$$

Consequently, the probability p_i that point i is correctly classified, can be computed with:

$$p_i = \sum_{j \in C_i} p_{ij}, \quad (3.2)$$

where $C_i = \{j | c_i = c_j\}$ is the set of points in the same class as i . Considering this, the objective function to maximize (expected number of correctly classified points) is:

$$f(A) = \sum_i \sum_{j \in C_i} p_{ij} = \sum_i p_i. \quad (3.3)$$

The differential of this function $f(A)$ with respect to A can then be used in a gradient rule for learning. The experimental evaluation is done on six datasets, of which five are from the UCI repository⁹. Compared to the standard Euclidean distance ($A = I$), the “whitening” transformation (A is a sample data covariance matrix) and the relevant components analysis (RCA) [4] transformation (A is the average of the within-class covariance matrices), the presented NCA approach performs always equally or better regarding classification performance. In addition to the metric learning approach, NCA can also be used for dimensionality reduction of the input data space, given A is restricted in size to $d \times D$. As this topic is irrelevant to metric learning per se, we will not go into detail here. Summing up, Goldberger *et al.* have introduced a relatively simple, yet effective non-parametric learning method that handles the task of distance learning, where the learned distance metric is always a Mahalanobis metric.

Woznica *et al.* [158] combine distances for complex representations, assessing three instantiations of the generally formulated problem of metric learning in classification. They show how to learn distance measures for the k NN classification method, by combining pre-defined distance measures with the corresponding complex representations. The rationale

⁹ archive.ics.uci.edu/ml/

behind this is that the representation of the learning instances and the distance metric used in the classification algorithm should be determined considering background knowledge. The distance measures that are combined in the work of Woznica *et al.* are seven set distances: The *Sum of Minimum Distances*, *Hausdorff*, *RIBL*, *Surjections*, *Linkings*, *Fair Surjections* and *Matchings* distance. The problem of combining the distance metrics is framed as an optimization problem. The authors focus on learning a “global” distance measure, not a “local” one that aims to define a neighborhood around each query instance. Formally, the general problem of supervised metric learning is defined as:

$$\min_Z \mathcal{F}_Z(\mathcal{S}, \mathcal{D}, D_Z^2), \quad (3.4)$$

where \mathcal{F}_Z is a differentiable function, $\mathcal{S} = \{(x_i, x_j) | y_i = y_j\}$ the set of instances sharing the same class values $y_i = y_j$, $\mathcal{D} = \{(x_i, x_j) | y_i \neq y_j\}$ and D_Z^2 is the quadratic combination of m distances defined as, e.g.:

$$D_A^2(x_i, x_j) = \vec{D}(s_i, s_j)^T A \vec{D}(x_i, x_j), \quad (3.5)$$

for $Z = A$ and A being a positive semi-definite matrix. The three instantiations of this generally formulated optimization problem evaluated by the authors are the methods proposed by Xing *et al.* [159], the Maximally Collapsing Metric Learning (MCML) method [47] and the NCA method by Goldberger *et al.* discussed earlier in this section. The method by Xing *et al.*, originally developed for semi-supervised clustering, has the disadvantage that it implicitly assumes that instances from the same class form a single compact connected set. Consequently, the cost function will be severely penalized if the negative class contains *any* examples that do not encode the positive class property. The main advantage of the MCML method over Xing’s method is that it puts more emphasis on pairs of points which are in different classes and as such the MCML method is better suited for classification problems. The advantage of NCA is that it is non-parametric, but on the other hand NCA has the disadvantage that it does not guarantee a global optimum. The discussed methodology is experimentally evaluated on five datasets and the number of nearest neighbors was optimized in an inner 10-fold cross-validation loop over $k = \{1, 3, 9\}$. The presented results show that MCML and NCA outperform Xing’s method as expected. They also perform better than the two applied baseline k NN methods that outperform Xing’s method sometimes significantly. The visualizations of the contribution of the different distance measures show that especially NCA assigns high weights to distance measures that individually show good performance and low weights to measures with poor performance. Xing’s method fails to do so. Concluding, we can say that the work by Woznica *et al.* presents a framework that allows for combination of different distance measures and their corresponding instance representations. It should be noted that this is the first approach to distance combination on non-vectorial data.

Another approach to metric learning – called Gaussian Coding Similarity (GCS) – was

presented by Hillel and Weinshall [59]. They learn distance functions by coding similarity in the context of image retrieval and graph based clustering. The basic idea is that two objects are considered more similar, the more one encoding can be compressed given the information of the other. Their notion of similarity is derived through the joint distribution of points x and y , $p(x,y|H_1)$, where H_1 is the hypothesis stating that x and y share the same label. This probability is estimated and used to define the coding similarity $codsim(x,y)$ as

$$codsim(x,y) = \log p(x|y, H_1) - \log p(x). \quad (3.6)$$

In other words, $codsim(x,y)$ is the information y conveys about x . At this point, Hillel and Weinshall make the assumptions that first, $p(x,y|H_1)$ is Gaussian with $G(\cdot|\mu,\Sigma)$ and second that $p(x)$ should be the marginal distribution of $p(x,y|H_1)$ with respect to both arguments. Using those assumptions, the following equation is derived for the GCS:

$$x^t \Sigma_x^{-1} x - (x - My)^t \Sigma_{x|y}^{-1} (x - My), \quad (3.7)$$

where $\Sigma_x = E[xx^t]$, $\Sigma_{xy} = E[x(y)^t]$, $M = \Sigma_{xy} \Sigma_x^{-1}$, $\Sigma_{x|y} = \Sigma_x - \Sigma_{xy} \Sigma_x^{-1} \Sigma_{xy}$ and leaving out the multiplicative and additive constants. The authors evaluate their GCS approach first on several synthetic datasets, second in a semi-supervised clustering environment using UCI datasets, and finally on the task of image retrieval. They compare GCS to three methods from the literature that learn a Mahalanobis metric: the method by Xing *et al.* [159] that learns the metric by non-linear optimization, the RCA method [5] and the method of De-Bie *et al.* [26]. The experiments with the synthetic data showed that Gaussian coding similarity has a clear advantage when the data contains several Gaussian data sets compared to RCA and the Euclidean metric. As one would expect, GCS totally fails on a synthetic dataset of concentric rings which violate the class convexity. Obviously, the Gaussian assumption is central to the presented approach and very strongly influencing on which kind of data it is applied best. In the graph based clustering experiments (agglomerative average linkage clustering), the authors found a large variance in the results of the algorithms, but conclude that their applied coding similarity gives the best average performance. On the image retrieval problem, the coding similarity approach outperformed the other methods. This allows for the (reverse) conclusion that the data in the image retrieval experiments has to be of Gaussian nature.

Weinberger's and Tesauro's [148] "Metric Learning for Kernel Regression" (MLKR) is an approach to learning a distance metric for regression problems. In kernel regression, the target value \hat{y}_t of a test instance is estimated by a weighted average over the training instances, where the weight is usually rapidly decaying with the distance of the training instance to the test instance:

$$\hat{y}_t = \frac{\sum_{j \neq i} y_j k_{ij}}{\sum_{j \neq i} k_{ij}}, \quad (3.8)$$

where $k_{ij} = k(\vec{x}_i, \vec{x}_t) \geq 0$ is the kernel function. In the discussed MLKR algorithm the distance-based kernel function with parameters θ is optimized using a gradient descent

procedure with step-size ϵ and loss function \mathcal{L} :

$$\Delta\theta = -\epsilon \frac{\partial \mathcal{L}}{\partial \theta}. \quad (3.9)$$

The applied loss function is the cumulative leave-one-out quadratic regression error on the training instances. The authors chose the Gaussian kernel and a Mahalanobis metric. Generally, the Gaussian kernel is defined as

$$k_{ij} = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{d(\vec{x}_i, \vec{x}_j)}{\sigma^2}}. \quad (3.10)$$

If local minima are of concern in a specific application, Weinberger and Tesauro advise to re-run the MLKR gradient descent several times with different random initializations and consequently choosing the outcome with the minimum training error. They stress that apart from metric learning, the MLKR algorithm can also be used for dimensionality reduction – as well as the NCA method by Goldberger *et al.* discussed earlier – and that MLKR can be understood as a supervised version of Principal Component Analysis (PCA). This application has been proven helpful on a synthetic dataset. In experiments on the performance of the metric learning scenario of MLKR, the algorithm is applied to a set of Delve¹⁰ datasets. As the experimental evaluation shows, MLKR out-performs linear regression, k -nearest neighbors with cross validation for parameter optimization and hierarchical mixture of experts trained with early stopping. Additionally, it is almost as precise as Gaussian Process Regression, which is considered a state-of-the-art technique by the authors.

3.2 Inductive Transfer

The main characteristic of inductive transfer is that the bias of one learning task is transferred to another related one. This is especially useful in cases where only few data is available for the given learning task. In cases when data on a similar or related learning problem is accessible, we can use that data to improve the predictive performance of the problem at hand. Please keep in mind that there is no uniform terminology for the datasets involved in inductive transfer. In the remainder of this section we will stick to the terms as used in the discussed original papers.

Rückert and Kramer [115] describe an inductive transfer approach in the domain of kernel-based learning. Learning a suitable kernel automatically from data for a given problem reduces the amount of data for the actual training of the learner. That is one reason why it makes sense to apply inductive transfer in this scenario. The rationale in the described approach is that a kernel that performs well on a related task will also perform well on the given task. The authors call the related datasets *transfer datasets*

¹⁰ Data for Evaluating Learning in Valid Experiments - <http://www.cs.toronto.edu/~delve>

and the dataset the information is transferred to *base dataset*. The first contribution of their presented work is a method that selects kernels from a set of kernels \mathcal{K} that perform well on the transfer datasets. Using those kernels and n transfer datasets, they frame the problem of finding a suitable kernel for the base dataset as a meta learning problem. This meta learning problem is solved in three steps: First, one highly predictive kernel \bar{k} per transfer dataset is generated. Second, a meta classifier \bar{f} that uses a meta kernel \bar{k} predicts a new base kernel k , given a base dataset. The meta kernel is learned from the transfer datasets and the transfer kernels \bar{k} . The meta classifier is then applied to the base dataset to obtain k , which is plugged into a standard SVM to construct a classifier for the base problem. Having a more detailed look at the first step, two problems arising when optimizing over a transfer dataset are solved. First, when optimizing \bar{k} and \bar{f} at the same time, the classifier \bar{f} induced by the SVM on new data might not fit well with the optimal kernel \bar{k}^* . Second, as some kernel classes tend to give kernel matrices with full rank, the optimization is prone to over-fitting. Those problems are circumvented by splitting the transfer dataset (\bar{X}, \bar{Y}) in two. The first part is used to optimize \bar{f} , while the transfer kernel $\bar{k} \in \mathcal{K}$ is evaluated on the whole dataset. The standard SVM regularization risk – modified to fit the above scenario – is used to rate the classifier \bar{f} :

$$\operatorname{argmin}_{\bar{\alpha} \in \mathbb{R}_+^{n'}, \bar{b} \in \mathbb{R}} C \sum_{i=1}^{n'} l_h \left(\bar{y}_i, \left[\bar{K}' \bar{D}' \bar{\alpha} \right]_i + \bar{b} \right) + \bar{\alpha}^T \bar{D}' \bar{K}' \bar{D}' \bar{\alpha}, \quad (3.11)$$

where $\bar{\alpha}$ is the coefficient vector, \bar{b} the threshold of the linear classifier to be found, n' the number of instances of the dataset subset (\bar{X}', \bar{Y}') , \bar{K}' the corresponding $n' \times n'$ kernel matrix, \bar{D}' a diagonal $n' \times n'$ matrix that contains the class labels in the diagonal and l_h the hinge loss. The optimal kernel is then chosen as follows:

$$\operatorname{argmin}_{\bar{\mu} \in \mathbb{R}_+^n, \|\bar{\mu}\| \leq 1} C \sum_{i=1}^n l_h \left(\bar{y}_i, \left[\bar{M} \bar{\mu} \right]_i + \bar{b} \right) + \bar{r}^T \bar{\mu}, \quad (3.12)$$

where $\bar{M} \in \mathbb{R}^{n \times l}$ with $\bar{M}_{ij} = \bar{y}_i \sum_{k=1}^{n'} \bar{y}_n \bar{\alpha}_k \bar{k}_j(\bar{x}_i, \bar{x}_k)$, and $\bar{r} \in \mathbb{R}^l$ with $\bar{r}_k = \sum_{i=1}^{n'} \sum_{j=1}^{n'} \bar{y}_i \bar{y}_j \bar{\alpha}_i \bar{\alpha}_j \bar{k}_k(\bar{x}_i, \bar{x}_j)$. After generating the transfer kernels, the main question of what a suitable kernel for the base problem is, can be answered. As default meta kernel the authors propose a histogram kernel. Where a domain specific kernel is available, it makes sense to use that instead of the default kernel. The meta learning procedure works with regularized loss minimization. As norm the authors choose the 2-norm to estimate the loss of predicted versus true values. The aim is now to find a coefficient vector $\hat{\alpha} \in \mathbb{R}^t$ and a threshold $\hat{b} \in \mathbb{R}^l$ that minimizes the kernel loss. This is formulated as follows:

$$\operatorname{argmin}_{\hat{\alpha} \geq 0, \hat{b}} C \sum_{i=1}^t l_2 \left(\bar{\mu}, \hat{f}^i(\bar{X}, \bar{Y}) \right) + \hat{\alpha}^T \hat{D} \hat{\alpha}, \quad (3.13)$$

where \hat{D} is the meta kernel matrix normalized with the main kernel weight vectors and

$f^{\setminus i}$ is f without incorporating the contribution of the instance it is evaluated on. To test their approach, Rückert and Kramer conducted experiments on six datasets of biological activity as well as ten datasets for text categorization. The molecular graphs in the biological activity datasets were represented with binary fingerprints of roughly 1000 non-redundant subgraphs. One of the datasets is used as base dataset, the remaining as transfer datasets. The presented results show that the inductive transfer approach is never statistically significantly worse than kernel learning or the main kernels. Inductive transfer outperforms kernel learning on all six datasets. The results on text categorization are in line with that. This shows that applying inductive transfer in the domain of kernel learning is a useful tool, if only limited data is available, but related datasets can be used.

In their approach on modeling transfer relationships between learning tasks for improved inductive transfer, Eaton *et al.* [32] represent the relationships of the available *source tasks*¹¹ as an undirected graph. Each source task is a node in the graph, whereas the weighted edges represent the concept of transferability. With this concept, the authors want to avoid a phenomenon of inductive transfer that is called negative transfer. Negative transfer means that knowledge is (accidentally or by method of the used algorithm) transferred from one or more irrelevant tasks and thus decreases the predictive performance instead of increasing it. The knowledge that is transferred in the presented approach is a vector of model parameters. As learner, the authors use biased logistic regression. Overall, the presented method is a three step process: First, n base models $\{m_i\}_{i=1}^n$ are learned¹² for the n source tasks $\{t_i\}_{i=1}^n$. Second, the model transfer graph is constructed from these models, including representing the transfer relationships. In the third step of the process, the transfer to the target task t_{n+1} is implemented. This is done by extending the transfer graph with a new node and learning a transfer function to determine the parameter vector that is transferred to the target task. The concept of transferability, which is central to this approach, is modeled as follows: Tasks with a high transferability are close in space and tasks with a low transferability are far apart. More formally, the transferability from task t_i to task t_j is defined as:

$$\text{transfer}_{i \rightarrow j}(q) = \text{performance}_{i \rightarrow j}(q) - \text{performance}_j(q), \quad (3.14)$$

where $\text{performance}_j(q)$ is the performance on task t_j without transfer given q training instances and $\text{performance}_{i \rightarrow j}(q)$ is the performance on task t_j with transfer from t_i given q training instances from task t_j . To be able to use an undirected graph, a symmetric undirected version of transferability is defined:

$$\text{transfer}_{i,j}(q) = \min(\text{transfer}_{i \rightarrow j}(q), \text{transfer}_{j \rightarrow i}(q)). \quad (3.15)$$

¹¹ In accordance to the discussed paper, the tasks that are used to learn the bias for the problem are here called source datasets/tasks and the given problem is called target problem/task.

¹² with WEKA's biased logistic regression implementation

Following this, the symmetric adjacency matrix A for q training instances is defined as:

$$A_{i,j}(q) = \begin{cases} 0 & \text{if } i = j, \\ \max(0, \text{transfer}_{i,j}(q)) & \text{otherwise.} \end{cases} \quad (3.16)$$

This matrix is extended for the transfer results to a matrix \hat{A} :

$$\hat{A} = \begin{bmatrix} A(\hat{q}) & \hat{w}^T \\ \hat{w} & 0 \end{bmatrix}, \quad (3.17)$$

where \hat{w} are the weights with $\hat{w}_i = \text{transfer}_{i \rightarrow n+1}(\hat{q})$ and \hat{q} are the data samples from task t_{n+1} . Using a set of basis functions determined by the graph Laplacian from spectral graph theory, the transfer function $\hat{f} : \hat{V} \rightarrow \mathbb{R}^\theta$ is modeled: $\hat{f} = \hat{Q}W$, with eigenvectors \hat{Q} and some $(n+1) \times \theta$ matrix W . Each column of W is fit separately using regularized least-squares. After constraining the smoothness of \hat{f} by penalizing the least-squares problem, W gets

$$W = (Q^T Q + \hat{A})^{-1} Q^T f, \quad (3.18)$$

where \hat{A} is the regularization operator. The transfer function \hat{f} can then be used to assign a parameter vector to each task on the transfer surface. Eaton *et al.* evaluate this *graph transfer* approach in the domain of letter recognition as well as in the domain of newsgroup recognition. To get an inductive transfer setting with several source datasets, the tasks are artificially split in source and target datasets. The approach is compared with an approach where they hand-select the source tasks and a base-line approach that averages over all available source tasks. The graph transfer statistically improves on the average approach and achieves performances near the hand-selected approach. The authors note that this approach is usually expensive. Those findings are reported on both discussed problem domains.

A recent approach by Zha *et al.* [163] learns distance metrics from training data and auxiliary knowledge. Here, an auxiliary dataset is basically the same as a “source” or “transfer” dataset in the approaches previously discussed. Again, the problem is formulated as a regularized loss minimization, but in addition to exploiting only labeled auxiliary examples, one of the two presented regularization approaches can also exploit unlabeled examples. Formally, Zha *et al.* describe the regularized loss function as:

$$f(M, \mathcal{S}, \mathcal{D}, \mathcal{C}, \mathcal{M}) = L(M, \mathcal{S}, \mathcal{D}) + R(M, \mathcal{C}, \mathcal{M}), \quad (3.19)$$

where \mathcal{S} and \mathcal{D} are the labeled examples grouped as similar and dissimilar ones, \mathcal{C} is the set of all labeled and unlabeled examples, \mathcal{M} is the available auxiliary knowledge (the auxiliary metrics) and M is the Mahalanobis metric. $L(\cdot)$ is a loss function and $R(\cdot)$ is the regularization term. The two presented algorithms share the loss function and have different regularization terms. Algorithm one is named L-DML (i.e., Log-determinant regularized Distance Metric Learning) and is defined such that the minimization of R

results in minimizing the divergence between the target metric M and the auxiliary metric $M_k \in \mathcal{M}$:

$$R(M, \mathcal{C}, \mathcal{M}) = \sum_{k=1}^K \mu_k (\text{tr}(M_k^{-1} M) - \log \det M), \quad (3.20)$$

with $\text{tr}(\cdot)$ being the trace operation on the matrix and $K = |\mathcal{M}|$. The second algorithm, M-DML (i.e., Manifold regularized Distance Metric Learning), encodes the auxiliary metrics and the unlabeled examples in the data collection \mathcal{C} :

$$R(M, \mathcal{C}, \mathcal{M}) = \sum_{k=1}^K \alpha_k \text{tr}(X L_k X^T M), \quad (3.21)$$

where L_k is the graph Laplacian. The experimental evaluation of the two algorithms is done on four face recognition datasets, with three datasets being used as auxiliary data and one as target dataset. The process is treated as a multi-class problem and solved with the k NN learner. The presented results suggest, although the transfer of knowledge is tested only for one target dataset that the developed methods outperform the discussed related and base line approaches (e.g. RCA, Xing’s method, NCA).

A topic in machine learning research that is closely related to inductive transfer is multi-task learning. Evgeniou *et al.* [36], for example, study the problem of learning many related tasks simultaneously using kernel methods and regularization. They show that the family of multi-task kernel functions presented in their work makes it possible to link estimating many task functions with regularization to single task learning. The problem of multi-task learning is formulated as follows: We have n tasks and corresponding to the l -th task we have m examples that are sampled from a distribution P_l on $\mathcal{X}_l \rightarrow \mathcal{Y}_l$. The goal of multi-task learning is to learn all n functions $f_l : \mathcal{X}_l \rightarrow \mathcal{Y}_l$ from the available data $\{(x_{il}, y_{il}) : i \in \mathbb{N}_m, l \in \mathbb{N}_n\}$. This goal is approached by the authors using the assumption that the functions f_l are linear and the parameter vector $u = (u_l : l \in \mathbb{N}_n) \in \mathbb{R}^n d$ is estimated with the minimizer of the following regularization function:

$$R(u) := \frac{1}{nm} \sum_{l \in \mathbb{N}_n} \sum_{j \in \mathbb{N}_m} L(y_{jl}, u_l' x_{jl}) + \gamma J(u), \quad (3.22)$$

where $\gamma > 0$ and J is a homogeneous quadratic function. Evgeniou *et al.* introduce a kernel function they call *linear multi-task kernel* using the feature matrix B :

$$K((x, l), (t, q)) = x' B_l' B_q t, \quad x, t \in \mathbb{R}^d, l, q \in \mathbb{N}_n. \quad (3.23)$$

After introducing the linear multi-task kernel the examples of the framework that should be valuable for applications are discussed. Among these examples are task clustering regularization and graph regularization. The approach is evaluated on two real-world datasets: A set from the domains of customer choice data and a set of “school data”

from the Inner London Education Authority¹³. In the evaluation the (standard) single-task kernel machine (SVM) is compared to the developed multi-task kernel machine. Two versions of the multi-task kernel machine are evaluated: A simple version (a), where the identity matrix is used to derive the multi-task kernel and a version (b) where the identity matrix in the kernel equation is replaced by a matrix estimated by running a Principal Component Analysis on the previously learned task parameters. The authors note that using one SVM for all tasks the performance is very poor with test errors between 38 and 42 percent (data treated as coming from one task). To study the effect of the number of tasks and the amount of data per task, the number of tasks is varied (50, 100, 200) and the number of data per task is also varied (20, 30, 60, 90). From their results, the authors conclude that the multi-task approaches outperforms the single-task one, if there is only few data per task (up to 60). Also, for few data, the simple variant (a) outperforms variant (b). Evgeniou *et al.* suspect that the PCA variant overfits here. Otherwise, if there is many data, only variant (b) significantly outperforms the single-task SVM. With an increasing number of tasks also the performance of the multi-task approaches increases.

Transfer learning is not restricted to the domain of supervised learning as presented so far. Frank *et al.* [39] present a novel transfer strategy for unsupervised learning, called the minimum transfer cost (MTC) principle that is used to transfer knowledge in the form of the model-order. The authors stress that the MTC principle renders the concept of cross-validation applicable to unsupervised learning problems. Consequently it can be used as valuable approach in such settings. The MTC principle is designed to find the model-order that performs best on a second test dataset from the same distribution, in other words, the optimized model complexity is transferred. This is the first conceptual difference to the approach presented in Chapter 6 of this thesis, in which the performance task is distance learning and not model-order selection. In contrast to the supervised case where a transferred model is directly applicable, a mapping function is necessary in the unsupervised case. The authors present such mapping functions for the application of the MTC to singular-value decomposition, maximum likelihood inference, k -means clustering, Gaussian mixture models, and correlation clustering. The authors also use the minimum transfer cost principle to find the optimal model order for a set of real-world problems: image denoising, role mining and detection of misconfigurations in access-control data. In the scope of this work, we refrain from a direct experimental comparison with the work by Frank *et al.* as it is only remotely related to our main contribution.

A recent publication by Pardoe *et al.* presents *TrAdaBoost.R2* [98], a regression variant of the well-established *TrAdaBoost* classification method [25]. *TrAdaBoost* is conceptually different from most transfer learning methods in that it uses the source dataset directly in combination with the target training set, instead of keeping a clean separation. In their work, Pardoe *et al.* first present an AdaBoost regression variant (AdaBoost.R2) before they give details on the two-stage transfer variant *TrAdaBoost.R2*. The two-stage version

¹³ available at: <http://multilevel.io.ac.uk/intro/datasets.html>

addresses the problem of overfitting by introducing a second step in which the weights are adjusted. In the first step, only the source instance weights are adjusted downwards. In a second step, only the target instance weights are adjusted. For the final model only the hypotheses from resulting from the second step are used. We now take a more detailed look at the algorithm: The input for the algorithm are the two labeled datasets T_{source} and T_{target} of sizes n and m , as well as the parameters for the number of steps S , the boosting iterations N , and the cross-validation folds F . In the WEKA implementation presented by the authors the base learner can also be chosen. The input datasets T_{source} and T_{target} are combined into T such that the first n instances are from T_{source} . The weight vector is initialized with equal weights and updated according to:

$$w_i^{t+1} = \begin{cases} w_i^t \beta_t^{e_i^t} / Z_t & 1 \leq i \leq n \\ w_i^t / Z_t & n + 1 \leq i \leq n + m, \end{cases} \quad (3.24)$$

where Z_t is a normalizing constant, and β_t is chosen such that the resulting weight of the target instances is $\frac{m}{(n+m)} + \frac{t}{S-1} \left(1 - \frac{m}{(n+m)}\right)$. A pseudo-code of the algorithm is given in Algorithm 3.1.

Algorithm 3.1 TrAdaBoost.R2

Input $T_{source}, T_{target}, S, N, F$
for $t = 1, \dots, S$ **do**
 $model_t \leftarrow$ AdaBoost.R2(T, w^t, N) using F -fold cross-validation
 call the learner with T, w^t to get hypothesis $h_t : X \rightarrow \mathbb{R}$
 calculate adjusted error e_i^t for each instance
 update the weight vector according to (3.24).
end for
Output $model_t$ where $t = \operatorname{argmin}_i error_i$

CHAPTER 4

Similarity Boosted QSAR – A Systematic Study of Enhancing Structural Descriptors by Molecular Similarity

Many applications and problem settings in cheminformatics rely on the concept of similarity of small molecules. Examples are search functionalities like substructure search, learning methods like k -nearest neighbor as well as variants of virtual screening or clustering. This makes molecular similarity one of the most central concepts in cheminformatics [93].

One particular application of similarities is their utilization as molecular descriptors. Using similarities of molecular graphs to encode the input space for building (Quantitative) Structure Activity Relationships ((Q)SARs) has been in the air – in one way or another – for some time. Cuadrado *et al.* [23] present an approach to QSAR based on similarity used as descriptors. They predict a set of 31 steroids using PLS regression as learner and an index called Approximate Similarity (AS) as descriptor space. An all-against-all AS matrix is used as descriptor set. As this index is based on maximum common subgraph (MCS) calculations, an application to larger datasets is impractical due to immense computational cost raised by the NP-hardness of the problem [40]. Another study that makes use of similarities in the descriptor space has been done by Richter *et al.* [108]. The authors show that they can significantly improve the regression mean absolute error for growth inhibition prediction on NCI DTP human tumor cell line screening data by using background knowledge. The background knowledge in this case are structures and a mode of action grouping of standard anti-cancer agents (ACAs). This information is encoded and added to the description of the molecules in terms of similarities of the training structures with respect to those ACA reference structures or to the groups of modes of action.

From a machine learning perspective, an elegant example of using similarities in the descriptor space is the concept of the empirical kernel map [139, 119]. With the empirical kernel map, any similarity function can be transformed into a valid kernel. To achieve this, a similarity vector is used to represent an instance with respect to the training set. The kernel itself is then defined as the dot product of two similarity vectors. Despite these efforts, many open questions remain, e.g. which similarity measure to use, which reference

molecules for the similarity calculations to use, or what the added value of the similarity information is.

In this study we introduce Similarity Boosted QSAR modeling using chemical similarity scores as descriptors. Our basic idea is to include knowledge about the similarity with respect to a set of reference structures as descriptors. The motivation behind this is that many toxicity responses result from multi-mechanistic processes and consequently, there can be structural diversity among the active compounds. The derived similarity scores with respect to representative active compounds aid a statistical learning algorithm in recognizing the various activity classes. Furthermore, extensive sets of reference compounds can be used to span the chemical space in the form of a structural representation, thereby positioning a molecule in it. Extended reference sets make the approach conceptually similar in the structural domain to the ChemGPS [96] method, where the principal components of the physico-chemical properties of a set of reference compounds are used as descriptors. Building upon the work by Richter *et al.*, we perform a systematic study assessing the usefulness of various similarity descriptors over a range of public data sets. We use different similarity measures for small molecules alone or in combination, we experiment with three variants to collect or compute reference molecules for the similarity calculations (one based on literature search and two based on clustering) and we combine the similarity descriptors with different sets of structural descriptors. The latter experiment aims to show that our similarity descriptors encode information complementary to that encoded by state-of-the-art structural descriptors, enhancing predictive performance when used in a classification setting.

The remainder of the chapter is structured as follows: In the next section we give a step-wise explanation of technical details of the similarity descriptors as well as the experimental setup. This is followed by an overview of the experimental results and a discussion before we conclude.

4.1 Materials and Methods

In this section we describe how the similarity descriptors are built, as well as the datasets and setup we use in our experimental evaluation. All developed methods are available within the open source cheminformatics package AZOrange [127].

4.1.1 Similarity Descriptors

Throughout the rest of this chapter, molecules will be denoted with the letter x and, if necessary, an index to distinguish between different molecules. To compile a *similarity descriptor vector* (SDV) for a molecule x , we use two building blocks: First of all, we need a set of similarity functions $\{s_j(x_a, x_b)\}_{j=1}^l \in \mathcal{S}$, with $l = |\mathcal{S}|$ that return real-valued measures of similarity given two molecules x_a and x_b . Second, we need a set of reference compounds \mathcal{R} with respect to which we want to calculate the similarities of our training

and test compounds. Molecules used as reference will be denoted x^r . More formally, we let $\{x_k^r\}_{k=1}^m \in \mathcal{R}$, with $m = |\mathcal{R}|$, denote a set of reference molecules. Given a set of symmetric molecule similarity measures \mathcal{S} , we define the *similarity descriptor set* $SD(\mathcal{S}, \mathcal{R})$ as

$$SD(\mathcal{S}, \mathcal{R}) = \left\{ sd_{s_1, x_1^r}(\cdot), \dots, sd_{s_1, x_m^r}(\cdot), \dots, sd_{s_l, x_1^r}(\cdot), \dots, sd_{s_l, x_m^r}(\cdot) \right\}, \quad (4.1)$$

where $sd_{s_j, x_k^r}(\cdot)$, $s_j \in \mathcal{S}$, $x_k^r \in \mathcal{R}$ denotes a *similarity descriptor*, i.e. a function returning the similarity s_j between a reference molecule x_k^r and its argument. Correspondingly, we obtain the *similarity descriptor vector* SDV of length $l \times m$ for molecule x_i :

$$SDV(\mathcal{S}, \mathcal{R}, x_i) = \left(sd_{s_1, x_1^r}(x_i), \dots, sd_{s_1, x_m^r}(x_i), \dots, sd_{s_l, x_1^r}(x_i), \dots, sd_{s_l, x_m^r}(x_i) \right), \quad (4.2)$$

$$\forall s_j \in \mathcal{S}, \forall x_k^r \in \mathcal{R},$$

where $sd_{s_j, x_k^r}(x_i)$ is the descriptor value for molecule x_i for descriptor $sd_{s_j, x_k^r}(\cdot)$. A schematic overview of the SDV composition is shown in 4.1.

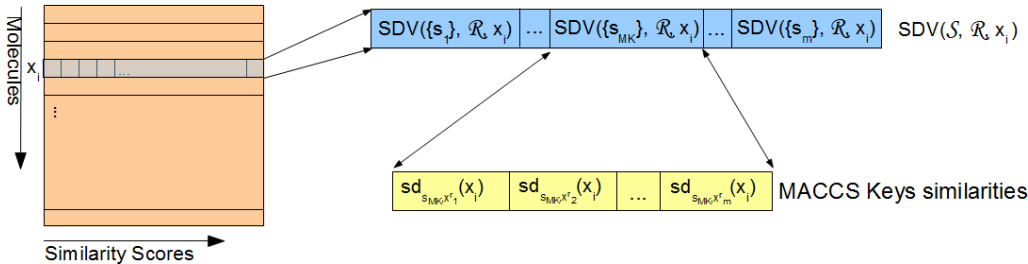


Figure 4.1: Schematic depiction of similarity descriptor vector composition for molecule x_i . MACCS keys fingerprint similarity is abbreviated with s_{MK} . \mathcal{R} is the set of reference compounds.

4.1.2 Molecular Similarity Measures

The molecular similarity measures used in the experimental evaluation of our approach are built on five molecular fingerprints available in the AZOrange framework: RDKit¹⁴ MACCS keys, RDKit topological fingerprints, RDKit extended and functional connectivity fingerprints, ECFP and FCFP, respectively and RDKit Atom Pairs fingerprints. In accordance with the RDKit reference manual recommendations, the similarity between two fingerprints A and B is either calculated with the Tanimoto similarity coefficient (2.1) (topological fingerprints, MACCS keys) or with the related Dice coefficient (2.2) (ECFP, FCFP, Atom Pairs). This results in the similarity measures s_{MK} , s_{topo} , s_{ECFP} , s_{FCFP} and s_{AP} , respectively. In our evaluation experiments we also use a combination of the five fingerprint similarities for building the SDV. Combination in this case means that all similarity measures in \mathcal{S} are used to calculate the similarity of a compound with respect to the reference compounds in \mathcal{R} and thus the length of the similarity descriptor vector will be five times the length when using just one of the similarity measures. The combination

¹⁴ <http://rdkit.org>

can be understood as union of the single similarity descriptors and the respective set of similarity measures will be denoted \mathcal{S}_{ALL} .

4.1.3 Selection of Reference Compounds

We experiment with three variants of how to obtain the set of reference molecules \mathcal{R} for the SDV calculations. The most intuitive way to obtain knowledge about representative active structures for an assay or another biological problem setting is literature search, which is our first variant (R_{LIT}). The descriptors constructed from similarity calculations with respect to the set of reference molecules \mathcal{R}^{lit} will be denoted $SD(\mathcal{S}, \mathcal{R}^{lit})$. The rationale is that we want to use *a priori* knowledge on different scaffolds or types of actives for the assay at hand. We use one representative for each of those types as reference compound and calculate the similarities of our data set with respect to those reference compounds. A detailed list of the \mathcal{R}^{lit} reference compounds is given in the supporting information. The second and third variant are based on structural clustering [121, 120]. The reference compounds \mathcal{R}^{act} in variant two (variant denoted R_{ACT} ; resulting in similarity descriptors $SD(\mathcal{S}, \mathcal{R}^{act})$) are cluster representatives from clustering all active compounds of an assay. Extending the number of active reference compounds as compared to R_{LIT} , might account for additional activity classes and hence mechanisms of action not covered by compounds resulting from a manual literature inspection. The clustering algorithm produces overlapping (non-disjoint) and non-exhaustive clusters. The parameters used for the structural clustering procedure are given in 4.1. The *thr* and *minSize* parameters are chosen in such a way that the number of clusters is roughly comparable for all seven data sets. We select one compound as cluster representative randomly. Consequently, variant two can be seen as an automated version of variant one, where the different types of actives are found by clustering (although, in case of R_{LIT} the reference compounds do not have to be contained in the assay set). As this second variant uses information about the class of a compound (only actives are clustered), extra care has to be taken during validation. Hence, to ensure a strict validation process the clustering is repeated for each fold, clustering only the actives contained in the training set. In variant three (R_{DB}), we cluster a subset of 300,000 compounds¹⁵ of the ChemDB [17] to obtain the reference compounds \mathcal{R}^{db} (with resulting descriptors $SD(\mathcal{S}, \mathcal{R}^{db})$). Here, the database subset represents the available chemical space and we want to position the molecules in our data set relative to representative compounds from the chemical space. This makes the third variant a more generic approach to the problem than variants one and two. As the number of clusters is relatively high due to the size of the clustered database, we set the minimum size of a cluster to provide a reference compound to 1500 resulting in 201 reference compounds in \mathcal{R}^{db} .

¹⁵ The subset has been generated by random sampling from the nearly five million commercially available small molecules in the ChemDB.

| Dataset | AID | <i>thr</i> | <i>minSize</i> | <i>nClusters</i> _{<i>R</i>_{ACT}} |
|---------|------|------------|----------------|--|
| hERG | 1511 | 0.6 | 5 | 126 |
| AhR | 2796 | 0.7 | 10 | 75 |
| ER | 639 | 0.5 | 5 | 49 |
| SRC-1 | 631 | 0.4 | 5 | 53 |
| THR | 1479 | 0.4 | 5 | 50 |
| KCNQ2 | 2156 | 0.7 | 5 | 45 |
| M1 | 677 | 0.4 | 5 | 56 |

Table 4.1: Parameters of the structural clustering algorithm. *thr* is the similarity threshold for a compound to be added to a cluster. *minSize* is the minimum size (number of compounds) of a cluster to provide a reference structure, while *nClusters*_{*R*_{ACT}} is the number of clusters resulting from using the *R*_{ACT} method (mean value from the 100-times repeated hold-out experiments). *nClusters*_{*R*_{ACT}} is equivalent to the size of \mathcal{R}^{act} .

| Dataset | <i>n</i> | BBRC | ECFP _{<i>r</i>₁} | \mathcal{R}^{lit} | \mathcal{R}^{act} | \mathcal{R}^{db} |
|---------|----------|------|--------------------------------------|---------------------|---------------------|--------------------|
| hERG | 3104 | 142 | 1526 | 30 | 630 | 1005 |
| AhR | 15980 | 257 | 1989 | 60 | 375 | 1005 |
| ER | 2302 | 147 | 1160 | 35 | 245 | 1005 |
| SRC-1 | 1622 | 117 | 1158 | 35 | 265 | 1005 |
| THR | 1632 | 88 | 1442 | 25 | 250 | 1005 |
| KCNQ2 | 6814 | 172 | 2119 | 25 | 225 | 1005 |
| M1 | 1446 | 28 | 1123 | 70 | 280 | 1005 |

Table 4.2: Summary of the used PubChem assay datasets and the number of descriptors in each descriptor set. The number of examples *n* comprises 50% actives and 50% inactives. Column \mathcal{R}^{lit} shows values for $|SD(\mathcal{S}_{ALL}, \mathcal{R}^{lit})|$, columns \mathcal{R}^{act} and \mathcal{R}^{db} the corresponding values for different \mathcal{R} values.

4.1.4 Core Descriptors

One of the goals of this work is to show potential improvement with respect to state-of-the-art structural representations. Consequently, we not only evaluate how our similarity descriptors perform, but we also assess if adding our similarity descriptors to a set of core descriptors improves the performance of a prediction model. We use two sets of structural core descriptors: Backbone Refinement Class (BBRC) descriptors [87] and Extended-Connectivity Fingerprints (ECFP) [111]. Please note that the ECFP descriptors used as core descriptors are different than the ones used in the similarity descriptors, although they are compiled with the same algorithm. The difference is in the parametrization and the way the fingerprint bits are used. When used as core descriptors, their descriptor values are used directly as input to the learning algorithms, when used as fingerprint for the similarity descriptors they are used to calculate the similarity with respect to \mathcal{R} and build the SDV. The algorithm compiling BBRC descriptors mines for frequently occurring class-correlated substructural features of molecules. Class-correlated means that the extracted substructural features not only have to occur frequently, but also have to show a significant correlation with the active class. Here, the significance is estimated with a chi-squared *p*-value lower bound. For the smaller datasets (*n* < 5,000 instances) we use an absolute

minimum support parameter of $minsup = 150$, for the larger ones ($n > 5,000$ instances) we use $minsup = 500$. As chi-squared significance parameter for the class correlation we use the default value of $ChisqSig = 0.95$. Remaining parameters are left at default values if not mentioned otherwise. When considering the results for predictions based on BBRC descriptors later in the chapter, please keep in mind that the $minsup$ and $ChisqSig$ parameters are not optimized in any way and performance improvements can be gained by doing so. The algorithm used to compile the BBRC descriptors was integrated into the AZOrange software and is available via the `getStructuralDesc.py` module. The Extended-Connectivity Fingerprint descriptors are circular fingerprint descriptors that use as input information not only the atom and bond type, but the six atom numbering independent Daylight atomic invariants [150] to encode atoms: the number of immediate heavy atom neighbors, the valence minus the number of hydrogens, the atomic number, the atomic mass, the atomic charge, the number of attached hydrogens, plus a seventh invariant added by Rogers *et al.* [111]: whether the atom is contained in at least one ring. The ECFP descriptor values were calculated with the RDKit functionality of AZOrange. We use default settings and set the radius parameter to $r = 1$ (ECFP_{r1}). 4.2 displays the number of compounds in each data set and the dimensionality of the calculated descriptor vectors. The size of the BBRC, ECFP_{r1} and $SD(\mathcal{S}, \mathcal{R}^{act})$ are mean values of the 100 hold-out training sets, as these feature sets are data-dependent.

4.1.5 Datasets

To evaluate the performance of our similarity descriptors we gathered seven datasets from the public database PubChem BioAssays [147], related to toxicologically relevant endpoints. The PubChem variable "PUBCHEM_ACTIVITY_OUTCOME" was used as the categorical response variable and due to the computational requirements, the study was restricted to binary classifiers. To avoid problems related to unbalanced data sets, which are considered outside the scope of this study, inactive compounds were randomly deselected from the PubChem data sets to assure an equal distribution of active and inactive structures. A tabular overview of all datasets including PubChem BioAssay ID (AID), endpoint and number of instances n is given in 4.2 (left hand side). The first dataset (hERG; PubChem AID: 1511) originates from a primary cell-based high-throughput screening assay for identification of compounds that protect hERG from block by proarrhythmic agents. Its size is 3,104 instances (1,552 active compounds and 1,552 inactives). The second dataset (AhR; AID: 2796) is compiled from a luminescence-based primary cell-based high throughput screening assay to identify activators of the aryl hydrocarbon receptor. This is the largest dataset of our study, with 15,980 compounds. The third dataset (ER; AID: 639) contains 2,302 compounds from a high throughput screening for estrogen receptor- α co-activator binding potentiators. The fourth dataset (SRC-1; AID:631) is comprised of 1,622 instances from a primary biochemical high throughput screening assay for agonists of the steroid receptor co-activator 1 recruitment by the peroxisome proliferator-activated

receptor gamma (PPARgamma). The fifth dataset (THR; AID: 1479) is compiled from a total fluorescence counter screen for inhibitors of the interaction of thyroid hormone receptor and steroid receptor co-regulator 2 (SCR-2). Its size is 1,632 compounds. The sixth dataset (KCNQ2; AID: 2156) consists of 6,814 chemicals from a primary cell-based high-throughput screening assay for identification of compounds that inhibit KCNQ2 potassium channels. Finally, the seventh dataset (M1; AID: 677) results from an antagonist confirmation screen aiming to discover novel allosteric modulators of the M1 muscarinic receptor and it contains 1,446 compounds.

4.1.6 Experimental Setup

The experimental evaluation of our approach was done with two validation strategies (cross-validation and hold-out validation), using the two learning algorithms Random Forests (RF) [13] and Support Vector Machines (SVM) [142] (CvSVM with RBF kernel), as provided in AZOrange. RF and SVM were selected as examples of popular and conceptionally different machine learning methods used by the QSAR community. The more basic experiments comparing single similarity measures with their combination and the experiments comparing the individual reference molecules selection variants were evaluated with a ten-fold cross-validation with the two learning algorithms. The remaining experiments, where we also assess the statistical significance of our findings with respect to data sampling, were conducted with a 100 times repeated hold-out evaluation with a 2:1 training set test set split ratio. The reason for this is that estimating the statistical significance of the difference of two classifiers in this way is easier to establish as compared to cross-validation. For those experiments only Random Forests were used due to running time issues. The statistical evaluation was done with a corrected resampled paired t-test [89] at a 95% significance level. The main difference to a standard t-test is that it takes into account the high Type I error the t-test produces in conjunction with random subsampling [29], which is due to the statistical dependence of the samples. In the result section we report prediction accuracy for cross-validation results and mean accuracy values with their respective standard deviations¹⁶ for hold-out experiments. The prediction accuracy is calculated as follows:

$$accuracy = \frac{correctpredictions}{overallpredictions}, \quad (4.3)$$

and thus represents the percentage of correct predictions.

In both validation scenarios (hold-out and cross-validation), an internal five-fold cross-validation for model hyper-parameter optimization was applied. For Random Forests the number of randomly selected descriptors at each node splitting in the constituting decision trees ($nActVars$) was optimized over all integers from $\frac{1}{4}\sqrt{n_{desc}}$ to $\frac{1}{2}n_{desc}$ with

¹⁶ Please note that standard deviations only quantify the scatter among the values and do not allow for any conclusions on the statistical significance of the difference of the means [24].

increments of $\frac{1}{4}\sqrt{n_{desc}}$, with n_{desc} being the number of descriptors in the training set. For the SVMs the C and the γ parameters were optimized in the ranges $C \in (2^{-5}, 2^{-3}, \dots, 2^{15})$ and $\gamma \in (2^3, 2^1, \dots, 2^{-15})$. These optimization intervals are defaults from the AZOrange software framework. As the running times for SVMs with internal cross-validation for parameter optimization are quite excessive for the larger datasets (see 4.2), we set a time threshold of 21 days¹⁷ after which experiments are terminated – only results obtained within that time frame are reported in the following. Cells marked with ** in the result tables or figures reflect this time constraint. Tables underlying the result figures and further additional result tables are shown in the Supporting Information.

The experiments are conducted in three consecutive steps: In a first step we analyze the performance that is achievable with the five similarity metrics used individually. The second experiment series evaluates the three strategies to select the reference molecules for the similarity descriptor calculations, before the combination of structural and similarity descriptors is evaluated in the third experiment.

4.2 Results

In this section we show and discuss our experimental results in a stepwise manner: First, we analyze the performance of the individual similarity measures s_{MK} , s_{topo} , s_{ECFP} , s_{FCFP} and s_{AP} and their combination in the set \mathcal{S}_{ALL} . Second, we try to find out which of the three methods to select the reference molecules for similarity calculations – R_{LIT} , R_{ACT} or R_{DB} – works best. In a third step, we assess the performance of combining the structural core descriptors with our similarity descriptors. This is done twice, in a simple approach of only pooling together the two types of descriptor sets and in an ensemble method approach.

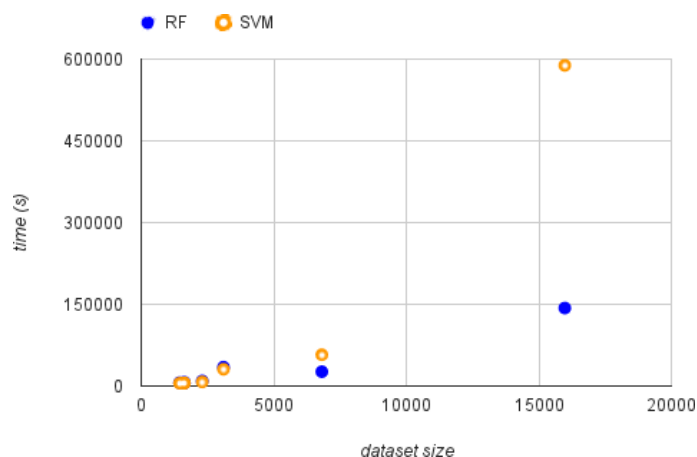


Figure 4.2: Scatter plot of running times. Shown are mean values of ten-fold cross-validation running times with Random Forests (RF) and Support Vector Machines (SVM) for the descriptor set $SD(\mathcal{S}_{ALL}, \mathcal{R}^{act})$.

¹⁷ 21 days on a single AMD Athlon 64 X2 Dual Core 5200+ machine with 2.6 GHz, 512KB Cache and 4GB DDR2 SDRAM 800

4.2.1 Similarity Descriptors

In the first experiment we compare the performance with solely one fingerprint type in the similarity descriptor vector to the performance achieved when including all 5 types ($\mathcal{S} = \mathcal{S}_{ALL}$) as displayed in 4.1. We select the literature review variant for providing the set of reference molecules (\mathcal{R}^{lit}), while there is no reason to expect different relative results with any of the clustering methods. Results compiled with the Random Forest learner are shown in 4.3a and with SVMs in 4.3b. Looking at the accuracies we can say that pooling the five basic similarity measures always gives an improvement in predictive performance (the $SD(\mathcal{S}_{ALL}, \mathcal{R}^{lit})$ descriptors are always the rightmost bar in a block). For Random Forests we show that this finding is statistically significant in all cases (see Table S11). The conclusion that can be drawn is that the five similarity measures applied in this study together outperform the results achieved individually. In the following, we consequently only consider the similarity descriptors with $\mathcal{S} = \mathcal{S}_{ALL}$.

4.2.2 Selection of Reference Molecules

The next experiment compares the descriptor sets based on the different strategies for selecting the reference compounds to calculate the similarities resulting in three descriptor sets: based on manual selection from background knowledge ($SD(\mathcal{S}_{ALL}, \mathcal{R}^{lit})$), based on clustering the assay actives ($SD(\mathcal{S}_{ALL}, \mathcal{R}^{act})$) and based on clustering a database representing the chemical space ($SD(\mathcal{S}_{ALL}, \mathcal{R}^{db})$). A tabular overview of the cross-validation results is given in 4.3.

The manual selection variant R_{LIT} is outperformed by the two clustering variants with both learning algorithms. In addition, the clustering methods do not require any manual work searching the literature for scaffold representatives or a priori mechanistic understanding. Comparing the two clustering variants we see that the $SD(\mathcal{S}_{ALL}, \mathcal{R}^{act})$

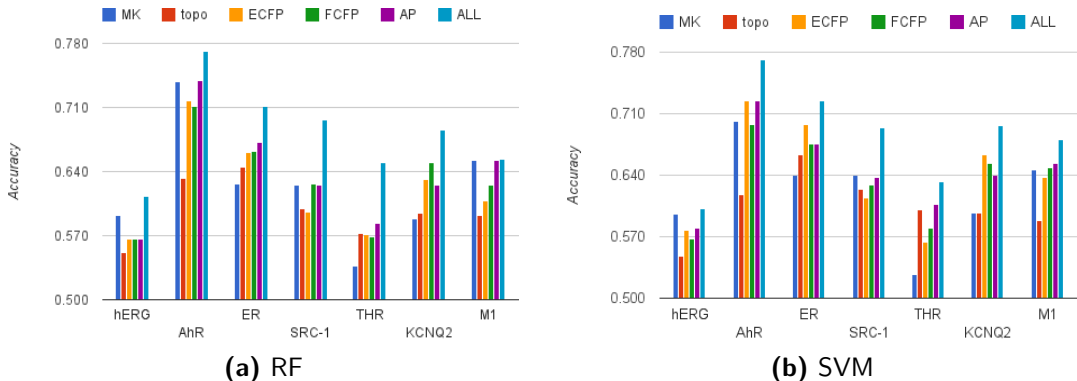


Figure 4.3: Bar charts of the classification accuracy in a 10-fold cross validation with Random Forest (RF) and Support Vector Machine (SVM) models based on similarity descriptors with different sets of similarity measures \mathcal{S} . In the key of the chart only the indices of the similarity measure (or set thereof) are given. Consequently, MK corresponds to $SD(\{s_{MK}\}, \mathcal{R}^{lit})$ (analogously for topo, ECFP, FCFP, AP), and ALL to $SD(\mathcal{S}_{ALL}, \mathcal{R}^{lit})$. The set of reference compounds \mathcal{R} is always set to \mathcal{R}^{lit} in these experiments.

| Dataset | RF | | | SVM | | |
|---------|---------------------|---------------------|--------------------|---------------------|---------------------|--------------------|
| | \mathcal{R}^{lit} | \mathcal{R}^{act} | \mathcal{R}^{db} | \mathcal{R}^{lit} | \mathcal{R}^{act} | \mathcal{R}^{db} |
| hERG | 0.613 | 0.635 | 0.630 | 0.602 | 0.660 | 0.653 |
| AhR | 0.772 | 0.781 | 0.774 | 0.772 | 0.807 | ** |
| ER | 0.711 | 0.734 | 0.738 | 0.725 | 0.747 | 0.730 |
| SRC-1 | 0.696 | 0.730 | 0.743 | 0.694 | 0.761 | 0.743 |
| THR | 0.650 | 0.673 | 0.644 | 0.633 | 0.691 | 0.655 |
| KCNQ2 | 0.686 | 0.742 | 0.732 | 0.697 | 0.777 | 0.768 |
| M1 | 0.653 | 0.694 | 0.678 | 0.680 | 0.693 | 0.690 |
| wins | 0 | 5 | 2 | 0 | 6 | 0 |

Table 4.3: Random Forest (RF) and SVM ten-fold cross-validation prediction accuracies using similarity descriptors only. The three descriptor sets based on similarity with respect to reference molecules are shown. The best descriptor set per learning algorithm is marked in bold print. Column headers only give the set of reference compounds, \mathcal{S} is always \mathcal{S}_{ALL} .

descriptors outperform the $SD(\mathcal{S}_{ALL}, \mathcal{R}^{db})$ descriptors in five of seven cases using Random Forests and in all cases using SVMs making clustering of training set activates the preferred method for selection of reference molecules.

The performance of models based solely on similarity descriptors ($SD(\mathcal{S}_{ALL}, \mathcal{R}^{act})$) is further compared to models using the two sets of established structural descriptors (core descriptors). 4.4 and Tables S12 and S13 display the accuracies and a statistical assessment of the differences in a 100 times repeated hold-out validation. We see that $SD(\mathcal{S}_{ALL}, \mathcal{R}^{act})$ descriptors obtain mean accuracies significantly better than BBRC in five out of seven datasets but also that ECFP_{r1} is better than $SD(\mathcal{S}_{ALL}, \mathcal{R}^{act})$ in all seven cases. This shows that the similarity descriptors used on their own are competitive to structural descriptors, although they do not outperform the best.

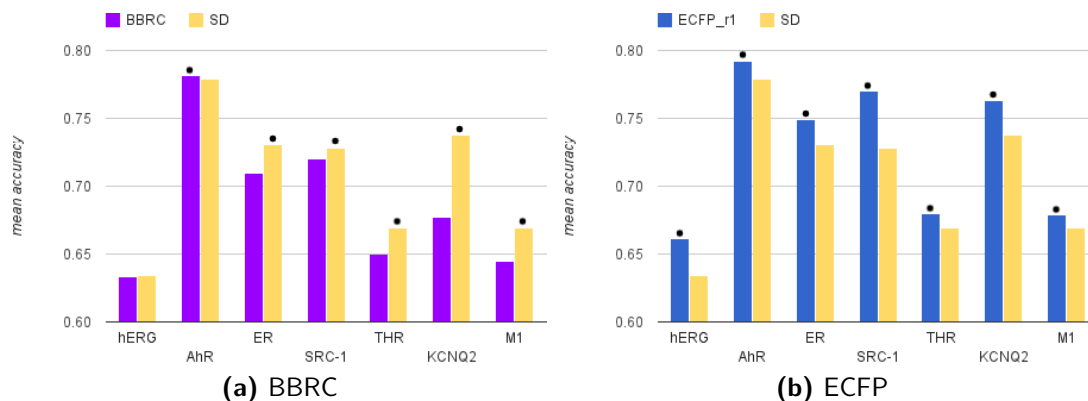


Figure 4.4: Bar charts showing the predictive accuracies for BBRC vs. $SD(\mathcal{S}_{ALL}, \mathcal{R}^{act})$ and ECFP_{r1} vs. $SD(\mathcal{S}_{ALL}, \mathcal{R}^{act})$. The similarity descriptors are abbreviated SD in the key of the chart. Bars representing results that are significantly better than their corresponding neighbor are marked with a black dot on top.

4.2.3 Combining Structural Descriptors and Similarity Descriptors

Our last experiment assesses the complementarity of the similarity descriptors and standard structural descriptors used for QSAR modeling. For this purpose, we add the structural BBRC and ECFP_{r1} descriptors to the $SD(\mathcal{S}_{ALL}, \mathcal{R}^{act})$ similarity descriptors and assess the performance of the combined descriptor sets (the union) to see if the similarity descriptors complement the structural descriptors. 4.5 and 4.4 show the results for the significance analysis done with Random Forests. All further analyses are conducted with Random Forests only for run time reasons. An important finding is that the combinations are always either significantly better than the structural core descriptors alone or on par with them. This suggests that there is information complementary to the structural descriptors encoded in the similarity descriptors.

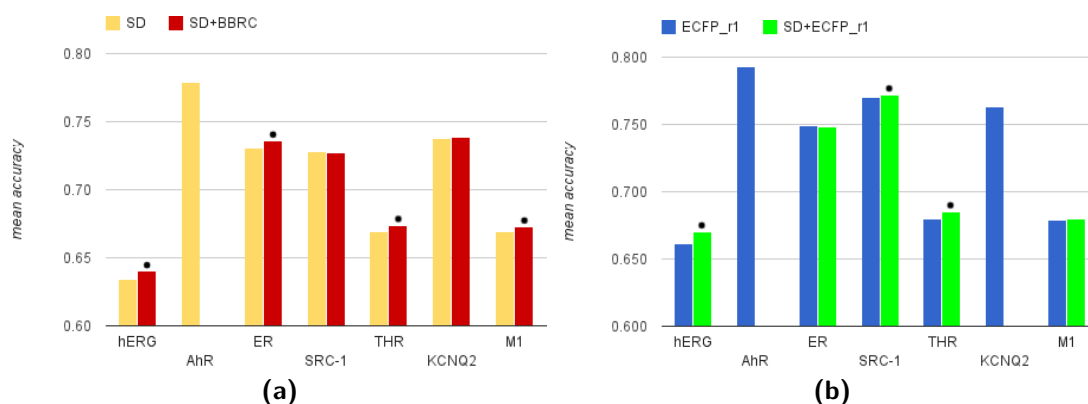


Figure 4.5: Bar charts showing the predictive accuracies for $SD(\mathcal{S}_{ALL}, \mathcal{R}^{act})$ (SD in the key) and the combination with BBRC and ECFP_{r1}. Bars representing results that are significantly better than the corresponding $SD(\mathcal{S}_{ALL}, \mathcal{R}^{act})$ or ECFP_{r1} bar are marked with a black dot. $SD+BBRC$ and $SD+ECFP_{r1}$ denote the classifiers built on the combined descriptor sets.

| Dataset | Random Forests | | | |
|---------|----------------|--------------|--------------------|----------------|
| | SD | $SD+BBRC$ | ECFP _{r1} | $SD+ECFP_{r1}$ |
| hERG | 0.634±0.015 | 0.640±0.011● | 0.661±0.012 | 0.670±0.014● |
| AhR | 0.779±0.008 | **± ** | 0.792±0.015 | **± ** |
| ER | 0.731±0.014 | 0.736±0.014● | 0.749±0.014 | 0.748±0.016 |
| SRC-1 | 0.728±0.018 | 0.727±0.016 | 0.770±0.016 | 0.772±0.011● |
| THR | 0.669±0.016 | 0.674±0.011● | 0.680±0.018 | 0.685±0.014● |
| KCNQ2 | 0.738±0.010 | 0.739±0.009 | 0.763±0.009 | **± ** |
| M1 | 0.669±0.017 | 0.673±0.019● | 0.679±0.021 | 0.680±0.017 |

●/○ statistically significant improvement/deterioration wrt. column $SD(\mathcal{S}_{ALL}, \mathcal{R}^{act})$ and ECFP_{r1}, respectively.

Table 4.4: Statistical significance analysis of improvement when combining $SD(\mathcal{S}_{ALL}, \mathcal{R}^{act})$ (SD in the table header) with structural descriptors. The null hypothesis is that there is no improvement compared to the $SD(\mathcal{S}_{ALL}, \mathcal{R}^{act})$ or ECFP_{r1} column. Shown are mean accuracy values ± standard deviations for one hundred hold-out runs¹⁸. $SD+BBRC$ and $SD+ECFP_{r1}$ denote the classifiers built on the combined descriptor sets.

To substantiate the assertion that the similarity descriptors and the structural descriptors are complementary, we analyze the diversity of the classifiers based on BBRC and $SD(\mathcal{S}_{ALL}, \mathcal{R}^{act})$ descriptors as well as that based on ECFP_{r1} and $SD(\mathcal{S}_{ALL}, \mathcal{R}^{act})$ descriptors. As measures of classifier diversity we calculate the Yule’s Q statistic, the correlation coefficient ρ and the double-fault measure DF as proposed by Kuncheva and Whitaker [76]. The exact mathematical definitions of the three measures are given in the supporting information. In addition, we perform a chi-squared test of independence for the two classifiers. The expectation of Q is zero for statistically independent classifiers, with $Q \in (-1,1)$. Classifiers that predict the same instances correctly will have values of $Q > 0$ and classifiers committing prediction errors on different instances will result in $Q < 0$. The double-fault measure is the proportion of cases in which both classifiers commit a prediction error and smaller values indicate a higher diversity of the classifiers ($DF \in (0,1)$). For this analysis, we arbitrarily select the AhR and the SRC-1 datasets as one of the larger and one of the smaller datasets. The results are given in 4.5. If we consider a significance level of 10^{-3} for the chi-squared test of independence with the null hypothesis being that the events of error of both classifiers (one based on $SD(\mathcal{S}_{ALL}, \mathcal{R}^{act})$ and one based on structural descriptors) are independent the results give an unclear picture. For the AhR dataset we have to reject the null hypothesis, for the SRC-1 we accept it. The Q statistic values for both datasets are slightly negative, as are the correlation coefficients. This indicates that the structural descriptor based models (BBRC or ECFP_{r1}) and $SD(\mathcal{S}_{ALL}, \mathcal{R}^{act})$ based models commit prediction errors on different instances, which is also reflected by the double-fault measure. The DF value of 0.11 for the AhR data set (BBRC and $SD(\mathcal{S}_{ALL}, \mathcal{R}^{act})$) means that the theoretical upper accuracy limit to be achieved with an ensemble method working with BBRC and $SD(\mathcal{S}_{ALL}, \mathcal{R}^{act})$ base classifiers is 88.9%. While our goal is clearly to eliminate all errors in case of conflicting predictions, this can hardly be achieved in practice.

Table 4.5: Analysis of classifier diversity for the AhR and SRC-1 datasets. Given are hold-out mean values for diversity measures of BBRC and $SD(\mathcal{S}_{ALL}, \mathcal{R}^{act})$ based classifiers as well as for ECFP_{r1} and $SD(\mathcal{S}_{ALL}, \mathcal{R}^{act})$ based classifiers. $SD(\mathcal{S}_{ALL}, \mathcal{R}^{act})$ is abbreviated SD in the table.

| | AhR | | | | SRC-1 | | | |
|---------------------------|--------|--------|-------|-------------|--------|--------|-------|------------|
| | Q | ρ | DF | $\chi^2 p$ | Q | ρ | DF | $\chi^2 p$ |
| BBRC, SD | -0.184 | -0.084 | 0.111 | $< 10^{-3}$ | -0.106 | -0.050 | 0.141 | 0.027 |
| ECFP _{r1} , SD | -0.195 | -0.089 | 0.107 | $< 10^{-3}$ | -0.143 | -0.066 | 0.125 | 0.023 |

The observation of classifier diversity suggests a more evolved approach (than just using all descriptors at once) to combining the structural descriptors and the similarity descriptors based on so-called ensembles [12, 156]. Ensembles are combinations of classifiers aiming for the reduction of the errors committed by the individual classifiers. Empirical

¹⁸ Please note again that standard deviations only quantify the scatter among the values and do not allow for any conclusions on the statistical significance of the difference of the means [24].

and theoretical results [12] have shown that there exists a positive correlation between the accuracy of ensembles and the diversity amongst the constituting base classifiers. Diversity in this case means that the base classifiers commit prediction errors on different instances and consequently can complement each other when combined. We applied a simple variant of the ensemble method stacking [156] in such a way that individual random forest models are learned for the similarity descriptors and for the structural descriptors. As random forests (as well as SVMs) can provide class probability estimates, we can use a combining function to get a single class probability from the two input class probabilities. The first combining function we applied is simply the mean of the two probabilities ($Stacking_{mean}$), the second multiplies the input probabilities ($Stacking_{mult}$). In the second variant the decision threshold for the result class is shifted from the standard value of 0.5 to 0.25. The results for both combining function variants are given in 4.6. Both stacking variants are able to improve the overall mean prediction accuracy significantly by roughly 4% compared to the best results so far.

| Dataset | Random Forests | | | |
|---------|--------------------|----------------|-------------------|-------------------|
| | ECFP _{r1} | $SD+ECFP_{r1}$ | $Stacking_{mean}$ | $Stacking_{mult}$ |
| AhR | 0.792±0.015 | **± ** | 0.835±0.007● | 0.809±0.028 ● |
| SRC-1 | 0.770±0.016 | 0.772±0.011 | 0.833±0.007● | 0.790±0.025 ● |

●/○ statistically significant improvement/deterioration wrt. column $SD+ECFP_{r1}$ (or ECFP_{r1} where results for the former are not available).

Table 4.6: Prediction accuracy results (\pm standard deviations) of the two stacking variants combining the $SD(\mathcal{S}_{ALL}, \mathcal{R}^{act})$ and ECFP_{r1} descriptor set based classifiers in comparison to the ECFP_{r1} and $SD(\mathcal{S}_{ALL}, \mathcal{R}^{act})+ECFP_{r1}$ based classifiers. $SD(\mathcal{S}_{ALL}, \mathcal{R}^{act})$ is abbreviated SD in the table.

To further understand the properties of the similarity descriptors, we analyze the mean sensitivity and specificity values corresponding to the BBRC, ECFP_{r1}, $SD(\mathcal{S}_{ALL}, \mathcal{R}^{lit})$, $SD(\mathcal{S}_{ALL}, \mathcal{R}^{act})$ and $SD(\mathcal{S}_{ALL}, \mathcal{R}^{db})$ classifications (see Tables S14 - S16). Except for $SD(\mathcal{S}_{ALL}, \mathcal{R}^{lit})$, where we observe a slight advantage for the sensitivity, the negative class seems to be predicted marginally better than the positive class by all descriptor sets for all datasets. However, the small differences between the sensitivity and specificity show that none of the descriptor sets can be identified as biased with respect to any of the classes.

4.3 Conclusions

In this chapter, we systematically studied Similarity Boosted QSAR, using chemical similarity to a finite set of selected reference compounds for QSAR modeling. We derived those references with three variants out of which one is based on literature search and two on automatic structural clustering. The two clustering variants outperformed the literature search-based method in our experiments. We suspect that the relative success of the $SD(\mathcal{S}, \mathcal{R}^{act})$ descriptors as compared to the results achieved with the $SD(\mathcal{S}, \mathcal{R}^{lit})$ descriptors derived from the limited set of activity classes of \mathcal{R}^{lit} compounds, represent-

ing only a subset of the mechanisms responsible for the activity, while the \mathcal{R}^{act} reference compounds might cover those activity classes better with a potentially greater structural diversity amongst the reference compounds. The $SD(\mathcal{S}, \mathcal{R}^{db})$ descriptors can be understood as a generic representation of chemical structure, in the spirit of ChemGPS in the physico-chemical space, perhaps primarily alternative rather than complementary to the BBRC and ECFP_{r1} representations. Keeping in mind that the parameters for the structural clustering have not been optimized at all, there should still be room for performance improvements. For example, the clustering algorithm can be further optimized by a refined selection of cluster representatives or hierarchical clustering variants.

For all three variants of the similarity descriptors we use a combination of five similarity measures. We show that they are complementary to a certain extent as using them in combination (\mathcal{S}_{ALL}) increases the predictive performance. The similarity descriptors could be further enhanced by adding pharmacophore-based, maximum common substructure (MCS) based or yet other similarity measures. Especially MCS based similarities could be of particular importance in toxicological modeling, as such responses are often triggered by the presence of a larger fragment, rather than the global properties or small fragments of the compound.

An interesting point of discussion is the information content of the different sets of descriptors. The ECFP_{r1} descriptors use information about the structure, based on circular atom neighborhoods. They also incorporate information on atom properties (atomic invariants). The second structural descriptor set, BBRC, is based on important substructural features. Important in this case means that the features are frequent and correlated with the endpoint variable.

Because it is theoretically possible to construct infinitely many structural features for structured data, such structural descriptor sets pose the difficult challenge [114, 143] of selecting a small number of relevant patterns or features from a larger set. The similarity descriptors on the other hand encode information about the chemical similarity with respect to a set of reference compounds and the similarity itself is based on diverse information: MACCS keys, topological information, ECFP and FCFP circular neighborhoods and atom pair information. Clustering actives of the training set to define the reference compounds (and also in the case of the R_{LIT} method if the reference compounds are part of the training set) and using similarity with respect to these compounds as descriptors, can be interpreted as instance selection. This option to reduce dataset redundancy and complexity is also provided by kernel machines like Support Vector Machines that intrinsically perform instance selection, as they use only the support vectors instead of all instances to discriminate between classes [142]. As the success of kernel methods is documented in particular for structured data like graphs [41], instance selection appears as a promising alternative to feature selection for such data, either during learning (kernel machines) or, as investigated in this chapter, during feature generation (similarity descriptors).

In our experiments using Random Forests and SVMs we showed that similarity descriptors in Similarity Boosted QSAR modeling perform quite well compared to es-

tablished structural descriptor sets. In addition, combining similarity descriptors with structural descriptors can often further enhance the performance, improving the accuracies achievable with solely structural descriptors. This indicates that the similarity descriptors encode information complementary to structural descriptors.

We support this finding by a statistical analysis of the diversity of classifiers based on either structural or similarity descriptors. The analysis shows that the different sets of descriptors commit prediction errors on different instances. We use this information in a simple stacking approach that improves the prediction results further and confirms that the structural and the similarity descriptors encode complementary information.

Finally, all methods are interfaced with the publically available AZOrange software framework. For additional material, lists of reference compounds \mathcal{R}^{lit} are given in Tables S1 - S7. Tables S8 - S10 contain the data underlying 4.2 and 4.3. Table S11 shows the significance analysis for the single similarity measures with Random Forests. Tables S12 and S13 contain the data underlying 4.4. Tables S14 - S16 show mean sensitivity and specificity values including their differences for BBRC, ECFP_{r1}, $SD(\mathcal{S}_{ALL}, \mathcal{R}^{lit})$, $SD(\mathcal{S}_{ALL}, \mathcal{R}^{act})$ and $SD(\mathcal{S}_{ALL}, \mathcal{R}^{db})$. A section with mathematical formulas for the diversity measures is provided. A section with mathematical formulas for the diversity measures is provided.

CHAPTER 5

Improving Structural Similarity Based Virtual Screening Using Background Knowledge

Medical needs are the starting point for every drug discovery and development project. Apart from the classical *in vitro* and *in vivo* studies used in this process, pharmaceutical research relies more and more on *in silico* methods like (high throughput) virtual screening or molecular docking simulations [136, 141]. Computational methods promise to shorten the typically time-consuming efforts that come with the development of new market-approved drug compounds. In the early drug discovery process, virtual screening is used to rank or select compounds from huge databases of potential drug candidates that are later assessed in wet-lab and animal studies. In case one or more ligand structures of the target protein are known and available, virtual screening based on ligand similarities can be used to calculate a ranking of candidate compounds in a database. This approach is applied if no X-ray or NMR structure of the protein target is available and receptor based approaches are not easily accessible.

In this chapter, we present a concept of how structural similarity based methods used in virtual screening can be improved by integrating chemical background knowledge in the form of binding relevant or informative structural elements. Improvement in this case means higher enrichment of chemical compounds related to the query compound in the similarity ranking of a compound database. Consequently, more potentially biologically active and less potentially inactive compounds are selected in virtual screening for further processing in the drug discovery pipeline (e.g. *in vitro*, *in vivo*). To achieve an improved enrichment we extract binding relevant substructures from known ligands and transform them into a fingerprint. This fingerprint is then used to extend a structural similarity measure. We present two approaches to extract the binding relevant information: first we use visual inspection of a known ligand as well as literature review to identify binding relevant substructures, second we test a relatively basic data mining approach. We apply the Free Tree Miner (FTM) software [113] that takes a set of two-dimensional chemical structures as input. FTM mines for and returns all substructures that occur frequently (more often than a user defined minimum support threshold) in the given set. These relevant substructures are then fragmented and the fragments' occurrences in a chemical

structure are used as bits in a binary occurrence fingerprint. A limitation of the data mining based approach is the need for more than one known ligand (active compound). An advantage of the approach is that it can still be applied if no literature information on the binding relevant substructures or structural patterns is available and that it saves human effort.

In our experiments we extend two structural similarity measures with background knowledge and apply them to rank compounds in a database according to their similarity to a known active structure. The first similarity measure is based on the size of the maximum common substructure (MCS – e.g., Raymond *et al.* [104]) of two molecules, the second is based on Extended Connectivity Fingerprints (ECFP) [111]. No other factors like drug-likeness, Cytochrome P450 interaction or physico-chemical properties are used. This enables an isolated view on the effects of the similarity methods used for the rankings. The extended similarity measures are compared to their non-extended versions to assess their performance by calculating enrichment factors for 1%, 5% and 10% of the database.

We show that adding background knowledge on important binding components of ligands to both, the MCS similarity and the ECFP similarity, changes the virtual screening ranking in such a way that the top structures have improved docking scores, related structures are ranked at better positions and clearly improved enrichment factor values are obtained. We also show that replacing the visual inspection and literature search by a data mining approach improves the similarity rankings for most assessed data sets. The data mining approach performs slightly weaker than the by-hand approach, but gives competitive results.

The remainder of the chapter is organized as follows: In the next section we give detailed information on the data and methods we use for the similarity calculations and our experimental setup. This is followed by a presentation and discussion of our results before we conclude. Additional result tables can be found in the appendix.

5.1 Materials and Methods

In this section we give detailed information on our experimental setting, on how we extend a similarity measure and on the data sources and evaluation measures used in our virtual screening experiments.

5.1.1 Experimental Setup

When virtual screening by means of similarity ranking is performed in a drug discovery project, the similarities of all compounds in the screening database are calculated with respect to one or more known ligands of the protein target (used as reference compounds). The compounds in the database are subsequently sorted according to their similarity scores in descending order so that the compounds most similar to the reference appear first in the ranking. A good similarity measure will find structures that are related to the reference –

or that potentially interact with the target protein – in the first few percent of the list. To assess the performance of different similarity measures we mix a set of known ligands into a set of decoys to form a screening database. As reference compound for the similarity rankings we use a randomly selected representative of the known ligands. After applying the standard similarity ranking procedure individually with each similarity measure, we can evaluate the performance of the similarity measures by examining the results for the known ligands in the screening database. The better a similarity measure is, the more known ligands will be in the top section of the ranking.

The experiments on extending a structural similarity measure can be divided into two lines of experiments: line “A” considers the by-hand selection of the binding relevant information that is used to extend the similarity measure and line “B” considers the data mining based selection of this information.

Table 5.1 shows a comparison of the steps necessary to apply the two presented approaches to extend similarity measures and rank a screening database.

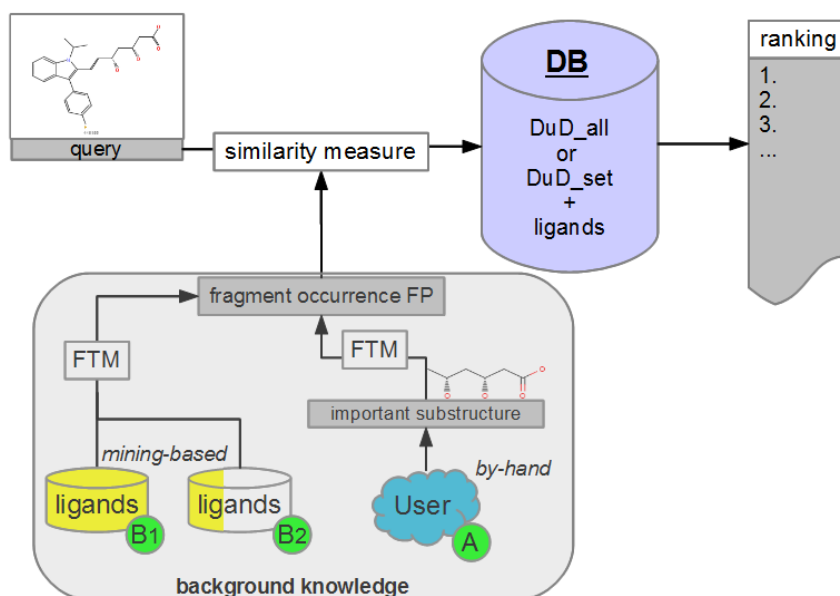


Figure 5.1: Overview of the experimental setup of the (A) by-hand extension experiments (B) mining-based extension experiments. The upper half of the workflow shows a similarity ranking without the incorporation of background knowledge.

5.1.2 Extended Similarity

The extended similarity measures proposed in this chapter are constructed from two building blocks: a structural similarity measure used as base similarity (sim_{base}) and a fingerprint-based similarity that is based on the binding relevant substructures (sim_{bind_fp}). After defining the extended similarity measure we will first explain the base

similarities and second explain the two variants used to derive sim_{bind_fp} . The *extended similarity* of two molecules a and b is defined as:

$$sim_{ext}(a,b) = 1 - \alpha sim_{base}(a,b) + \alpha sim_{bind_fp}(a,b), \quad (5.1)$$

where $sim_{bind_fp}(a,b)$ gives the Tanimoto similarity coefficient (2.1) of two binary sub-structural occurrence fingerprints of molecules a and b . The choice of $\frac{1}{3}$ as weight coefficient for the fingerprint-based part is arbitrary and no experimental evaluation or optimization regarding this parameter has been attempted. In our experiments the substructures constituting the fingerprint for sim_{bind_fp} are selected by visual inspection and literature review or by a data mining approach.

| step | A: By-hand approach | B: Mining-based approach |
|------|---|---|
| 1 | Review literature/examine structure to determine BI | Calculate frequently occurring substructures (BI) in known ligands with FTM |
| 2 | Fragment relevant substructure and build binary occurrence fingerprint from all fragments | Build fingerprint from frequently occurring substructures |
| 3 | Rank DB with sim_{ext} | Rank DB with sim_{ext} |

Table 5.1: Overview table of the steps necessary to apply the two presented approaches to extend similarities. DB: database; BI: binding-relevant information.

The first structural similarity measure (sim_{base}) that we extend is based on the notion of maximum common substructures (MCS). For computation of the size of the MCS of two molecular structures, the JChem Java classes were used (JChem 5.4.0.0, ChemAxon (<http://www.chemaxon.com>)). The similarity between two structures was then calculated with the similarity measure proposed by Wallis *et al.* [144]:

$$sim_{MCS}(a,b) = \frac{|mcs(a,b)|}{|a| + |b| - |mcs(a,b)|}, \quad (5.2)$$

where $|\cdot|$ gives the number of vertices in a graph, and $mcs(a,b)$ calculates the MCS of molecules a and b . Consequently, $|mcs(a,b)|$ is the number of atoms of the MCS of molecules a and b . The second structural similarity measure is based on Extended-Connectivity Fingerprints (ECFP) [111], a standard method in pharmaceutical research and industry. ECFP fingerprints are circular, structural feature fingerprints that use as input information not only the atom and bond type, but the six atom numbering independent Daylight atomic invariants [150] to encode atoms: the number of immediate heavy atom neighbors, the valence minus the number of hydrogens, the atomic number, the atomic mass, the atomic charge, the number of attached hydrogens, plus a seventh invariant added by Rogers *et al.* [111]: whether the atom is contained in at least one ring. These fingerprints are available via the RDKit functionality of the open source cheminformatics software AZOrange [127]. The radius parameter for the ECFP fingerprint calculation was used at the

default value of $r = 2$. The fingerprint similarity of two ECFP fingerprints is calculated with the Dice coefficient (for a mathematical definition see the supplementary material).

Our first approach (approach A) to extend sim_{base} relies on literature review or visual inspection of a set of known ligands to retrieve a binding relevant substructure (or fragment). Once such a substructure is known we apply the Free Tree Miner [113] software without minimum frequency constraint to produce all possible fragments of the substructure. From these fragments we build a binary occurrence fingerprint that is used to encode the reference molecules and all database molecules. The fingerprints are then used to calculate sim_{bind_fp} . In our experimental evaluation of approach A on the HMGR data set, we derive the binding relevant substructure not only by literature review (which would be the standard approach and sufficient in most cases), but we support the process by additional calculations. First, we use the MCS similarity measure to rank the screening database. Subsequently, the top 25 compounds of the similarity ranking are docked to the HMGR receptor. The examination of the results in combination with the literature review is used to derive the binding relevant structural parts that are used as background knowledge. For the second data set used to evaluate approach A (PPAR γ) we derive the binding relevant substructure from reviewing known ligands from the DrugBank [75] database. We expect the PPAR γ hand-selection experiments to show less improvement than those on HMGR as the binding relevant information is selected with less effort.

In our second approach to extend sim_{base} , the data mining based approach - denoted approach B, we try to substitute the by-hand selection of the additional knowledge that is integrated into the similarity measure by applying data mining techniques. To retrieve the substructure fingerprint used for the similarity measure extension we calculate the set of frequently occurring substructures from a set of known ligands with the FTM algorithm. From those frequent substructures we build the binary occurrence fingerprint used to encode our molecules and used to calculate sim_{bind_fp} . Two variants of input ligand sets are tested: (B1) We use all available ligands for the generation of the fingerprint fragments. The minimum support parameter (*minsup*) for the FTM software was chosen in such a way for each data set that it resulted in approximately the same number of substructural features as the fingerprint of approach A did (57 features). The parameters are given in Table 5.3. This ensures that we can exclude the length of the fingerprint (feature number) as driving force of improvement or degradation. (B2) We use only 10% (20% in case of the DuD HMGR, ADA and TK data sets) of the ligand compounds randomly chosen from the respective DuD ligand sets to work with a more realistic setting, where only few compounds interacting with the protein are known in advance. The minimum support parameter of FTM was set to 0.9 for all data sets. This second, reduced variant provides less information on the ligands to be found in the ranking and consequently poses a more realistic but harder problem. The resulting enrichment factor values of the extended similarity measures should show less improvement over the non-extended versions compared to the first variant that uses all ligands.

For a graphical overview of the two extension approaches as well as how they interact

with the base-line similarity ranking please see Figure 5.1.

5.1.3 Data

In the first line of experiments (by-hand selection) we use only two data sets for our analysis, in line two of the experiments (data mining based extension) we use ten data sets from the Directory of useful Decoys (DuD) [62] as well as 25 ChEMBL activity class data sets [55]. We use different database setups in our evaluation: For experiments with the DuD data sets we use either all 95,000 decoy structures of the DuD (DuD_{all}) or only those DuD decoys as database that were designed especially for the target ligand system considered (DuD_{set}). For the experiments with the ChEMBL activity classes we use a subset of the ZINC [64] database.

HMGR and statins In our approach A experiments we first consider the problem of inhibition of the enzyme HMG-CoA reductase (HMGR). Well-known inhibitors of HMGR are chemicals from the drug class of statins (HMG-CoA reductase inhibitors). Most of them are marketed drugs or drugs under development. Inhibition of HMGR lowers the cholesterol levels and prevents cardiovascular diseases [80], which are a major problem in developed countries as coronary artery disease affects 13 to 14 million adults in the United States alone [33]. The statins are structurally quite similar as can be seen in Figures 5.2a - 5.2f. All of them are competitive inhibitors of HMGR with respect to binding of the substrate HMG-CoA, but not with respect to binding of NADPH [34]. The protein receptor used in the docking procedure is the structure of HMGR co-crystallized with fluvastatin (Figure 5.4a, CID 446155), which is available in the PDB [10] with identifier [PDB:1HWI] [65]. We use two sets of known ligands that are mixed with the decoys and provide the reference compound in this first set of experiments: first the set of statins and second the HMGR ligands provided by the DuD HMGR data set. In case the statins are used as ligand set, we repeat the experiment with each statin as query compound, otherwise we randomly select ten DuD HMGR ligands and use each one of those as query compound.

PPAR γ In addition to HMGR we test the by-hand selection approach on the PPAR γ data set. The PPAR γ receptor binds peroxisome proliferators such as hypolipidemic drugs and fatty acids. Once activated by a ligand, the receptor binds to a promoter element in the gene for acyl-CoA oxidase and activates its transcription. It therefore controls the peroxisomal beta-oxidation pathway of fatty acids and is a key regulator of adipocyte differentiation and glucose homeostasis [118]. The DrugBank [75] database lists - amongst others - these eight drugs that are market approved and known PPAR γ interactors: Bezafibrate, Glipizide, Ibuprofen, Mesalazine, Sulfasalazine, Balsalazide, Rosiglitazone and Pioglitazone. An overview of the drugs, their DrugBank IDs and structures are given in Table 5.2 and Figure 5.3. We use the same query and database set-up as with the HMGR experiments.

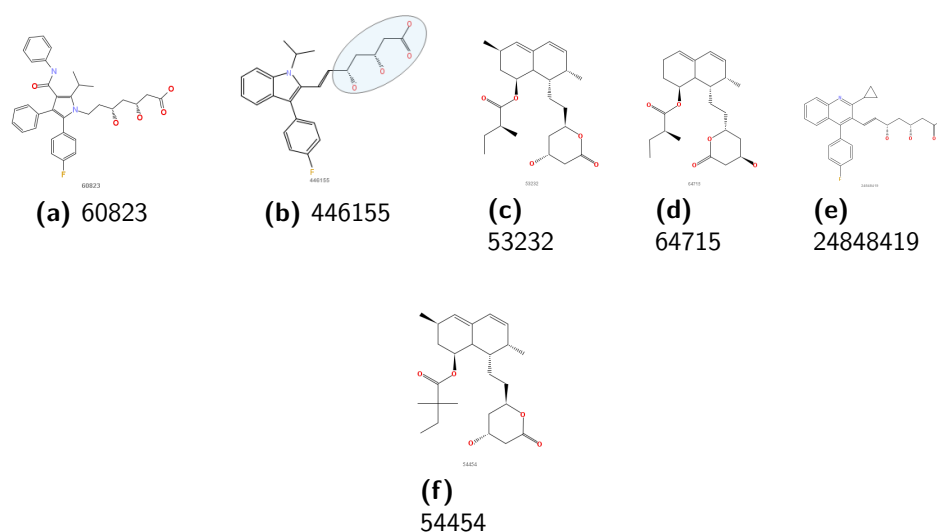


Figure 5.2: 2D depiction of the six statin structures with their corresponding PubChem compound identifiers

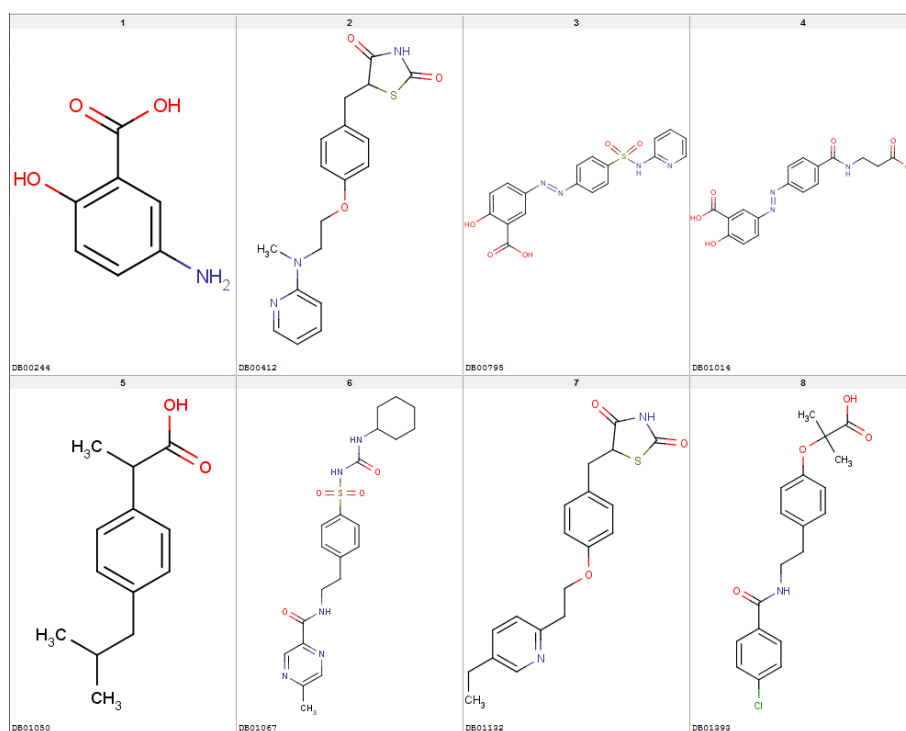


Figure 5.3: Eight DrugBank listed PPAR γ active drugs that have “approved” status. The DrugBank ID is shown with the molecule.

Directory of useful Decoys As database for the line two experiments, we use the Directory of useful Decoys that is designed to avoid bias in docking and screening studies. The DuD database consists of more than 95,000 decoy structures and 2,950 ligand structures (more than 30 decoy structures per ligand) for 40 protein targets including HMGR. We chose nine target structures from the DuD database in addition to HMGR. The original

| DrugBank ID | Drug Name |
|-------------|---------------|
| DB01393 | Bezafibrate |
| DB01067 | Glipizide |
| DB01050 | Ibuprofen |
| DB00244 | Mesalazine |
| DB00795 | Sulfasalazine |
| DB01014 | Balsalazide |
| DB00412 | Rosiglitazone |
| DB01132 | Pioglitazone |

Table 5.2: PPAR γ market approved drugs with DrugBank ID and drug name.

forty DuD target sets are grouped into six classes: nuclear hormone receptors, kinases, serine proteases, metalloenzymes, folate enzymes and other enzymes. We selected the additional nine protein targets to cover all six classes: estrogen receptor (ER; antagonists) and peroxisome proliferator-activated receptor γ (PPAR γ) from the class of nuclear hormone receptors, p38 mitogen-activated protein kinase (P38 MAP) and thymidine kinase (TK) for the class of kinases, factor Xa (FXa) for the class of serine proteases, adenosine deaminase (ADA) for the class of metalloenzymes, dihydrofolate reductase (DHFR) for the class of folate enzymes and the acetylcholine esterase (AChE) as well as cyclooxygenase 2 (COX-2) for the remaining “other enzyme” class. An overview of the DuD datasets used in this study is given in Table 5.3. For DHFR three and for FXa two ECFP similarities could not be calculated due to software problems (the applied RDKit software was not able to process those molecules). The respective compounds were removed from the experimental setting. For this second set of experiments we always chose the ligand with the best docking score as provided in the DuD database as reference compound and mix the remaining ligands with the decoys.

| Protein | PDB code | ligands | decoys | protein class | <i>minsup</i> | <i>fp_length</i> |
|---------------|----------|---------|--------|-----------------|---------------|------------------|
| HMGR | 1HW8 | 35 | 1242 | other enzyme | 0.9 | 66 |
| ER | 3ERT | 39 | 1399 | nuclear h.r. | 0.7 | 62 |
| PPAR γ | 1FM9 | 81 | 2910 | nuclear h.r. | 0.96 | 90 |
| P38 MAP | 1KV2 | 234 | 8399 | kinase | 0.83 | 57 |
| TK | 1KIM | 22 | 785 | kinase | 0.9 | 74 |
| FXa | 1F0R | 142 | 5102 | serine protease | 0.8 | 81 |
| ADA | 1STW | 23 | 822 | metalloenzyme | 0.8 | 70 |
| DHFR | 3DFR | 201 | 7150 | folate enzyme | 0.8 | 70 |
| AChE | 1EVE | 105 | 3732 | other enzyme | 0.77 | 93 |
| COX-2 | 1CX2 | 349 | 12491 | other enzyme | 0.6 | 65 |

Table 5.3: Overview of the used DuD data sets. *minsup* gives the minimum support parameter used in the FTM calculations and *fp_length* the length of the resulting fingerprint. hormone receptor is abbreviated h.r..

Evaluation measures

To evaluate the performance of the similarity measures, we consider the enrichment factor (EF) [35] that is achieved by a virtual screening. The enrichment factor reflects the amount of known related structures in the first $x\%$ of the ranked database. In practice, often only the highest ranked compounds are of interest and considered further in the drug discovery pipeline. The enrichment factor is defined for certain fractions of the database:

$$EF(\%) = \frac{(N_{active(\%)} / N_{(\%)})}{(N_{active} / N_{all})}, \quad (5.3)$$

where $EF(\%)$ is given for the specified percentage of the ranked database, $N_{active(\%)}$ is the number of active compounds in the selected subset of the ranked database, $N_{(\%)}$ is the number of compounds in the subset, N_{active} is the number of active molecules in the dataset and N_{all} is the number of compounds in the database. For an easier interpretation of the EF values, it is helpful to compare them to the maximum possible enrichment at the specified fraction of the database:

For easier comparison we do not use the $EF(\%)$ directly, but the difference of maximum possible enrichment and achieved enrichment:

$$\Delta_{EF} = EF_{max} - EF(\%). \quad (5.4)$$

Keep in mind that for Δ_{EF} smaller values are better and the optimal Δ_{EF} is zero. In our study, we use the top 1%, 5% and 10% fractions of the ranked database to calculate the EF values. In the results section of this work we restrict ourselves to showing the Δ_{EF} values, but the $EF(\%)$ and EF_{max} values are given in the supplementing material. In addition to the enrichment factor we calculated the mean ranks (μ_{Rank}) of the ligands in the similarity rankings. Smaller μ_{Rank} values indicate better ranking quality of the similarity measure.

5.1.4 Docking Procedure

Molecular docking was applied in order to assess if the extensions to the structural similarity measures are suitable for virtual screening. For the HMGR experiments we did the docking ourselves, for the second experiment we used the docking scores as provided in the DuD database. We now describe the docking procedure applied in the by-hand HMGR experiment.

HMGR is a tetra-mer with four identical binding sites whereas two chains contribute residues to one binding site. In the PDB six co-crystallizations of HMGR are available, each with one statin: atorvastatin (PDB ID:1HWK), fluvastatin (1HWI), simvastatin (1HW9), compactin (1HW8), rosuvastatin (1HWL) and cerivastatin (1HWJ). A comparison of the CoA bound binding sites with the statin bound binding sites showed rearrangements. In the statin bound binding sites some residues are disordered which fold to an

α -helix when CoA is bound. In the presence of the α -helix, a narrow pantothenic acid-binding pocket is formed making it impossible for statins to bind. Instead a hydrophobic groove is formed that accommodates the hydrophobic moieties of the statins which accounts for a tighter binding of the statins [65]. Since we are interested in drug candidates with a similar binding ability as the statins, we focus on the statin bound HMGR structures. According to Istvan *et al.* [65] the orientation of the side chains in the binding sites does not differ among the statins. This was confirmed by a superposition of the six PDB structures with Pymol (<http://www.pymol.org/>). Due to this we chose to perform a rigid receptor cross-docking of the structural similar drug candidates to 1HWI with Glide 5.7 from the Maestro Suite of Schrödinger. If not indicated otherwise, the default settings were used. The first step in the docking process was the automatic preparation of the complete PDB structure of fluvastatin (1HWI) with the Protein Preparation Wizard of the Maestro Suite. Since there are four identical binding sites, the docking was performed with only one of them. At some binding sites ADP is bound nearby. Since ADP does not participate in statin binding [65] the binding site mainly formed by chain D with some contribution of chain C was chosen, which lacks ADP. In order to speed up the docking procedure, the multi-mer was simplified by removing the redundant chains A and B. The receptor preparation was completed by the manual removal of all waters, the ligand molecule and the ADPs of the other binding site formed by chain C and D. The selected drug candidates were prepared using Ligprep 2.5. In a preprocessing step of the docking procedure the receptor grid for the chosen binding site was pre-calculated using the Glide 5.7 Receptor Grid Generation. The co-crystallized fluvastatin in the chosen binding site was used as reference ligand. Subsequently the rigid receptor docking was performed with the extra precision mode of the Glide 5.7 Ligand Docking application.

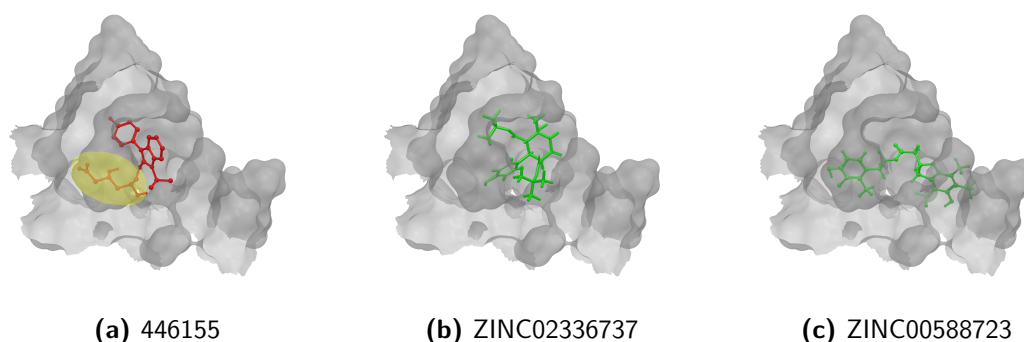


Figure 5.4: Original position of fluvastatin (Figure 5.4a) and docking of two non-statins with best docking score from MCS (Figure 5.4b) and MCS_{ext} (Figure 5.4c) similarity ranking docked to HMGR (1hwi). Only the active site of the receptor is shown. The hand-selected important fragment is marked in yellow in Figure 5.4a.

5.2 Results and Discussion

5.2.1 By-Hand Experiments

In the first set of experiments we extract the binding-relevant knowledge used to extend the structural similarity measures by literature review and support the process by MCS similarity ranking and docking calculations. We therefore rank the screening database (including decoys and statin ligands) with respect to fluvastatin using sim_{MCS} . Subsequently, we docked the top 25 compounds of the similarity ranking to the HMGR receptor. Looking at the docking results in Table 5.4 (and the long version Table B.1 in the supplement), it can be seen that only one compound (CID 60823) has a good docking score. This is atorvastatin, one of the two statins found in the top 25 of the MCS similarity ranking. All other compounds have rather weak docking scores. Four structures from this ranking are shown in Figures 5.5a - 5.5d and the docking of the best non-statin is shown in Figure 5.4b. It can clearly be seen that the highlighted part of the structure of fluvastatin (Figure 5.2b and Figure 5.4a) or something structurally similar, is not present in any of the structures (non-statins).

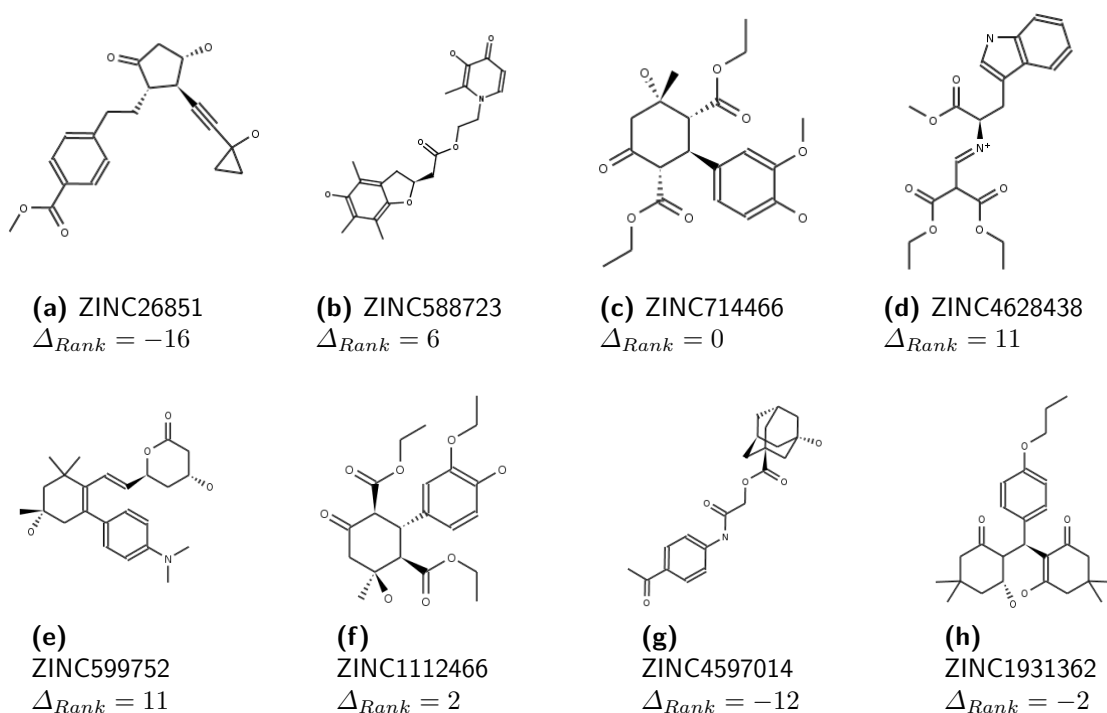


Figure 5.5: Structures from MCS ranking (Figures 5.5a-5.5d) and extended similarity ranking (Figures 5.5e-5.5h). PubChem CIDs and rank difference are given next to the structure.

According to Istvan *et al.* [65], this part mimics the original binding ligand and consequently is essential for binding. The hydrophobic part of the statins is responsible for the nano-molar affinity of the statins but not sufficient for inhibitory binding on its own.

Taking those facts into consideration, we decided to use the highlighted hydrophilic part of fluvastatin as background knowledge in our study. As described in the Material and Methods Section, the substructure was fragmented and used to derive a binary occurrence fingerprint of length 57 for the extended similarity measure (5.1).

We then calculated a similarity ranking with the extended MCS similarity measure and again docked the top 25 compounds. The results of docking the top 25 compounds of the extended MCS similarity ranking are shown in Table 5.5. The docking scores are clearly improved in comparison to those of the structures found by the MCS similarity ranking given in Table 5.4. This means that the compounds found will very likely have a higher binding affinity to the receptor. Figure 5.4 show the original position of fluvastatin and dockings of the two non-statins with the best docking score from the two similarity rankings. It can be seen that the ligand of the extended MCS similarity (in Figure 5.4c) enters the active site much better than the one of the MCS similarity (in Figure 5.4b).

As last experiment for the by-hand approach, we calculated similarity rankings with the ECFP similarity and also with an extended version of the ECFP similarity. We use the same binding-relevant substructure as for the MCS similarity. Comparing the differences in enrichment factors of the ligand structures in the ranked databases (MCS and ECFP similarity rankings) with the respective extended variants (see Table 5.6), it is clear that the extension is beneficial in all cases. Especially the MCS similarity, that shows a slightly weaker performance than the ECFP similarity, benefits from the similarity extension. Here an improvement of Δ_{EF} can be seen in all except one cases (if further improvement is possible). For ECFP a decrease in Δ_{EF} can be seen in all except four cases.

For the second data set we use for testing the by-hand approach, PPAR γ , we shorten the selection procedure. By visual inspection of the eight approved drugs shown in Table 5.2 and Figure 5.3 as well as binding information on Rosiglitazone given in by Liberato *et al.* [81] we select two binding relevant substructures as shown in Figure 5.6. As described in the Material and Methods Section, the substructures were fragmented and used to derive a binary occurrence fingerprint for the extended similarity measure (5.1). The results for the similarity rankings that are calculated in analogy to the HMGR by-hand experiments are given in Table 5.7. The results clearly show that the reduced effort to extract the binding-relevant information has direct impact on the ranking performance. Only in half of the settings (MCS lig vs. DUD_{set}, ECFP lig vs. DUD_{all} and ECFP lig vs. DUD_{set}) we see improvements of the extended similarity measures in comparison the base similarity measures. From that we conclude that it is of high importance to be very careful on selecting the binding-relevant structural information when using the presented approach A (by-hand selection).

We then calculated a similarity ranking with the extended MCS similarity measure and again docked the top 25 compounds. The results of docking the top 25 compounds of the extended MCS similarity ranking are shown in Table 5.5 (Table B.2 of the appendix). The docking scores are clearly improved in comparison to those of the structures found by the MCS similarity ranking given in Table 5.4 (Table B.1 of the appendix). This means

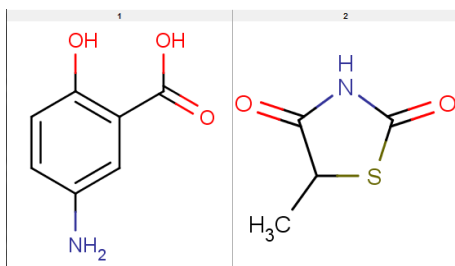


Figure 5.6: Binding relevant substructures used for calculating the bind_fp fingerprint for the PPAR γ by-hand experiments (approach A).

| Rank | CID | Score | Rank _{MCS} | Δ Rank _{MCS} |
|------|--------------|-----------|---------------------|------------------------------|
| 1 | 60823 | -10.564 | 2 | -1 |
| 2 | ZINC02336737 | -5.808526 | 13 | -11 |
| 3 | ZINC00026851 | -5.699634 | 19 | -16 |
| 4 | ZINC00588719 | -5.568737 | 11 | -7 |
| 5 | ZINC00599752 | -5.46502 | 5 | 0 |
| 6 | ZINC00588053 | -5.463745 | 16 | -10 |
| 7 | ZINC00864379 | -5.291673 | 15 | -8 |
| 8 | ZINC01253780 | -5.211104 | 14 | -6 |
| 9 | ZINC00714466 | -5.149133 | 9 | 0 |
| 10 | ZINC00588723 | -5.14689 | 4 | 6 |

Table 5.4: Results of the docking run (MCS top 25). Δ Rank = Rank_{docking} - Rank_{MCS}. A negative Δ Rank value means, in the MCS similarity the compound is ranked lower, a positive Δ Rank that it is ranked higher than by the docking procedure. For the complete table, refer to Table B.1 of the appendix.

that the compounds found will very likely have a higher binding affinity to the receptor. Figures 5.4a - 5.4c show the original position of fluvastatin and dockings of the two non-statins with the best docking score from the two similarity rankings. It can be seen that the ligand of the extended MCS similarity (in Figure 5.4c) enters the active site much better than the one of the MCS similarity (in Figure 5.4b).

| Rank | CID | Score | Rank _{MCS_{ext}} | Δ Rank _{MCS_{ext}} |
|------|--------------|------------|-----------------------------------|--|
| 1 | ZINC00588723 | -10.382184 | 16 | -15 |
| 2 | 24848419 | -7.980885 | 3 | -1 |
| 3 | ZINC01253780 | -7.385909 | 9 | -6 |
| 4 | ZINC00625939 | -7.157018 | 11 | -7 |
| 5 | ZINC01032240 | -7.104563 | 5 | 0 |
| 6 | ZINC00864379 | -7.052449 | 10 | -4 |
| 7 | ZINC00026851 | -6.910078 | 19 | -12 |
| 8 | ZINC00714466 | -6.702119 | 6 | 2 |
| 9 | ZINC01112466 | -6.667553 | 7 | 2 |
| 10 | 64715 | -6.654007 | 12 | -2 |

Table 5.5: Results of the docking run (MCS_{ext} top 25). Δ Rank = Rank_{docking} - Rank_{MCS_{or}ECFP}. A negative Δ Rank value means, in the extended similarity the compound is ranked lower, a positive Δ Rank that it is ranked higher than by the docking procedure. For the complete table, refer to Table B.1 of the appendix.

As last experiment for the by-hand approach, we calculated similarity rankings with the ECFP similarity and also with an extended version of the ECFP similarity. We use the same binding-relevant substructure as for the MCS similarity. Comparing the differences in enrichment factors and the mean ranks μ_{Rank} of the ligand structures in the ranked databases (MCS and ECFP similarity rankings) with the respective extended variants (see Table 5.6), it is clear that the extension is beneficial in all cases. Especially the MCS similarity, that shows a slightly weaker performance than the ECFP similarity, benefits from the similarity extension. Here an improvement of Δ_{EF} can be seen in all except one cases (if further improvement is possible). For ECFP a decrease in Δ_{EF} can be seen in all except four cases.

| query vs. DB | MCS | | | MCS _{ext} | | |
|----------------------------|-----------|----------|---------|---------------------|----------------|----------------|
| | 1% | 5% | 10% | 1% | 5% | 10% |
| stat vs DuD _{all} | 63.5±14.1 | 10.1±3.7 | 4.5±2.3 | 16.7±18.3 | 0.0±0.0 | 0.0±0.0 |
| stat vs DuD _{set} | 61.2±13.6 | 9.4±4.4 | 3.1±2.2 | 28.9±26.1 | 1.7±1.8 | 0.0±0.0 |
| lig vs DuD _{all} | 14.1± 1.7 | 1.1±0.3 | 0.0±0.1 | 3.1± 0.7 | 0.3±0.1 | 0.0±0.0 |
| lig vs DuD _{set} | 0.0± 0.0 | 2.1±0.5 | 0.0±0.0 | 0.0± 0.0 | 1.7±0.2 | 0.0±0.0 |
| | ECFP | | | ECFP _{ext} | | |
| | 1% | 5% | 10% | 1% | 5% | 10% |
| stat vs DuD _{all} | 50.0± 0.0 | 10.0±0.0 | 5.0±0.0 | 8.3±20.4 | 1.7±0.0 | 0.0±0.0 |
| stat vs DuD _{set} | 52.0± 0.0 | 10.1±0.0 | 5.0±0.0 | 20.2±20.2 | 0.0±0.0 | 0.0±0.0 |
| lig vs DuD _{all} | 6.1± 1.2 | 1.4±0.2 | 0.1±0.1 | 5.9± 1.5 | 1.4±0.3 | 0.0±0.0 |
| lig vs DuD _{set} | 0.1± 0.1 | 0.9±0.2 | 0.0±0.0 | 0.0± 0.0 | 0.4±0.1 | 0.0±0.0 |

Table 5.6: Δ_{EF} values for all four similarity methods for the hand-selection experiments with HMGR at 1%, 5% and 10% of the database. Improvements compared to the non-extended variant are marked in bold print. stat = statines.

| query vs. DB | MCS | | | MCS _{ext} | | |
|----------------------------|-----------|-----------|---------|---------------------|----------------|----------------|
| | 1% | 5% | 10% | 1% | 5% | 10% |
| ChBa vs DuD _{all} | 82.9± 6.5 | 15.0± 1.3 | 7.0±1.1 | 79.7±6.5 | 15.0±1.9 | 7.3±0.8 |
| ChBa vs DuD _{set} | 73.8±10.5 | 12.2± 3.4 | 5.2±1.6 | 80.0±6.5 | 14.4±1.8 | 7.0±0.9 |
| lig vs DuD _{all} | 10.3± 5.7 | 7.9± 3.3 | 4.1±2.7 | 11.0±6.1 | 8.2±7.5 | 2.9±2.8 |
| lig vs DuD _{set} | 8.2± 3.6 | 6.9± 2.8 | 3.0±2.3 | 8.0±6.1 | 7.0±9.5 | 1.9±2.6 |
| | ECFP | | | ECFP _{ext} | | |
| | 1% | 5% | 10% | 1% | 5% | 10% |
| ChBa vs DuD _{all} | 79.7± 9.3 | 13.8± 1.9 | 6.6±0.6 | 78.1±8.8 | 14.7±1.6 | 7.0±0.9 |
| ChBa vs DuD _{set} | 70.7±11.5 | 12.9± 1.6 | 5.5±1.9 | 78.5±8.9 | 14.1±2.3 | 6.7±1.1 |
| lig vs DuD _{all} | 6.9± 3.3 | 7.4±11.1 | 2.4±2.6 | 10.0±8.3 | 4.6±1.1 | 1.2±1.2 |
| lig vs DuD _{set} | 4.2± 2.1 | 6.1± 1.3 | 0.9±1.1 | 3.9±8.7 | 3.5±1.9 | 0.9±0.7 |

Table 5.7: Δ_{EF} value for all four similarity methods for the hand-selection experiments with PPAR γ at 1%, 5% and 10% of the database. Improvements compared to the non-extended variant are marked in bold print. ChBa = ChemBank ligands.

5.2.2 Mining-Based Experiments

In the following, we first assess for both data-mining based variants (B1: all known ligands used to calculate the fragment occurrence fingerprint or B2: only part of them used), if the extension of the MCS and the ECFP similarity measures with the data mining derived fingerprint improves the quality of the similarity ranking. Second we compare the data mining approach with the by-hand approach for the HMGR data set. The results for variant B1 are given in Tables 5.8 - 5.10. To see how the data mining based approach performs, when only few ligand structures are available as background knowledge, we reran the experiments with variant B2: using only ten per cent randomly chosen from the respective DuD ligand sets (20% due to smaller ligand set sizes in case of the HMGR, ADA and TK data sets) to extract background knowledge. The results using DuD_{set} as database are given in Tables B.3 - B.5.

| DuD set | MCS | | | MCS _{ext} | | |
|---------------|-----------|----------|---------|--------------------|------------------|-------------------|
| | 1% | 5% | 10% | 1% | 5% | 10% |
| HMGR | 8.5± 4.5 | 7.0±6.8 | 2.8±3.6 | 4.6± 9.8 | 2.0±1.0 | 0.5± 0.3 |
| ER | 13.6± 7.6 | 12.6±4.1 | 5.3±1.7 | 13.1± 7.0 | 11.4±2.8 | 3.7± 0.9 |
| PPAR γ | 4.6±10.6 | 1.2±5.4 | 1.7±2.8 | 4.6±11.0 | 3.8±5.5 | 1.5± 2.9 |
| P38 MAP | 9.6± 7.9 | 8.6±3.7 | 3.3±1.8 | 3.8± 5.4 | 4.8±4.2 | 2.4± 2.1 |
| TK | 20.1± 4.4 | 12.6±2.1 | 5.1±1.6 | 18.3± 5.3 | 12.3±2.7 | 4.0± 1.3 |
| FXa | 4.6±11.2 | 7.6±3.8 | 3.3±1.8 | 3.5±11.0 | 6.4±4.6 | 2.5± 2.5 |
| ADA | 10.1± 6.4 | 8.2±3.0 | 4.3±3.6 | 9.2± 4.8 | 7.7±2.0 | 2.3± 0.8 |
| DHFR | 10.9±10.6 | 11.7±2.9 | 4.7±1.1 | 3.1± 5.0 | 0.3±0.3 | 0.0± 0.0 |
| AChE | 10.3±12.5 | 11.3±4.7 | 4.8±2.5 | 10.0±11.8 | 9.5±5.8 | 4.4± 3.0 |
| COX-2 | 12.3± 9.2 | 11.7±2.2 | 5.3±1.1 | 10.7±10.3 | 10.1±3.8 | 2.2± 2.6 |
| w/d/l | | | | 10 / 0 / 0 | 9 / 0 / 1 | 10 / 0 / 0 |

Table 5.8: Mean Δ_{EF} and standard deviation for the MCS and MCS_{ext} similarity methods at 1%, 5% and 10% of the database (receptor specific decoy set DuD_{set}). The extension fingerprint is calculated from all ligands (approach B1). Improvements of MCS_{ext} compared to MCS are marked with bold print. w/d/l = wins/draws/losses.

Testing for the improvement of the extended similarity compared to the baseline similarity, on average, for a given data set, we find the following numbers of wins and losses for a fixed α coefficient of 0.3 weighting the contribution of the extension of the similarity measure in Table B.3 (MCS vs MCS_{ext}, approach B2): 8:2 (at 1%), 7:3 (at 5%), 8:2 (at 10%). Similar or even stronger results can be found for other settings, in particular for retrieving 10% of the compounds: 8:2 on Table 5.9 (ECFP vs. ECFP_{ext}, approach B2), 10:0 on Table 5.8 (MCS vs. MCS_{ext}, approach B1) and 8:2 on Table 5.9 (ECFP vs. ECFP_{ext}, approach B1).

Checking whether these results are statistically significant, we chose one of the weakest significance tests, the sign test [28], which is based on only one weak assumption, namely the independence of the measurements. The sign test has a p -value ≤ 0.109 for a result of 8 wins vs. 2 losses, a p -value ≤ 0.0215 for 9 wins vs. 1 loss, and even smaller for 10 wins vs. 0 losses. We apply the sign test to determine whether Δ_{EF} is on average greater for

one method compared to another for a given data set.

While the results already show improvements of the score for a fixed α of 0.3, one might be interested in the results for an optimal α , which we do not know beforehand. Also, it is interesting to know into which range optimal α s fall and whether 0.3 is a suitable default value. Results are shown in Tables 5.11 - 5.13 as well as in Figures 5.7 and 5.7. As it turns out, the statistics of the number of wins and losses can still be improved, e.g., from 8:2, 7:3, 8:2 to 10:0, 9:0, 9:1, respectively, and so forth. On the other hand, the optimal α s seem to vary somewhat, with a value of 0.3 not being too large for most data sets and most percentages of retrieved compounds (see Table 5.11).

To account for the variation of Δ_{EF} across different sets within a cross-validation (see the standard deviations in Tables 5.8 - 5.10 as well as Tables B.3 - B.5), we wanted to check whether the scores of two compared methods go up or down in a concerted fashion, or whether this is not the case. For this purpose, we present the win/loss statistics for a fixed α of 0.3 in Tables B.9 and 5.14. As can be seen in these tables, the proportion of 8:2 or 9:1 still holds when zooming in on the individual data sets from Tables 5.8, 5.8, B.3 and B.4. Unfortunately, the results are not independent anymore, thus, the sign test can no longer be applied.

To investigate if the extension similarity $\text{sim}_{\text{bind_fp}}$ on its own is better than the base similarity measures MCS and ECFP we provide Tables 5.10 and B.5. The results show that the bind_fp similarity in general is not better on its own in comparison to the base similarities. Only for 10% of the database in approach B1 the bind_fp similarity performs better in the ranking than MCS or ECFP.

| DuD set | ECFP | | | ECFP _{ext} | | |
|---------------|-----------|-----------|----------|---------------------|------------------|------------------|
| | 1% | 5% | 10% | 1% | 5% | 10% |
| HMGR | 8.7± 9.4 | 6.8± 8.5 | 4.2±5.6 | 0.0± 0.0 | 2.8±2.2 | 0.9± 0.5 |
| ER | 8.0± 4.0 | 7.4± 3.9 | 6.7±4.6 | 6.3± 4.8 | 9.2±1.6 | 3.3± 1.2 |
| PPAR γ | 1.3± 0.7 | 7.1±11.2 | 1.0±0.7 | 4.2±11.1 | 3.6±5.7 | 1.8± 2.8 |
| P38 MAP | 7.0± 5.9 | 5.9± 3.0 | 3.4±2.0 | 2.8± 5.7 | 5.0±4.1 | 2.4± 2.1 |
| TK | 9.8± 6.0 | 12.1± 8.9 | 10.9±6.4 | 16.5± 8.4 | 11.4±3.6 | 4.3± 2.0 |
| FXa | 7.4±11.3 | 2.4± 2.0 | 1.7±1.5 | 3.5±11.0 | 4.3±5.3 | 2.0± 2.7 |
| ADA | 6.3± 3.3 | 6.4± 4.5 | 8.9±6.0 | 8.3± 7.1 | 7.7±2.0 | 2.4± 1.0 |
| DHFR | 2.5± 2.0 | 1.8± 1.5 | 1.8±1.5 | 1.9± 0.9 | 0.1±0.1 | 0.0± 0.0 |
| AChE | 15.0±11.2 | 5.2± 2.3 | 6.8±3.8 | 11.0±12.0 | 9.4±5.6 | 4.0± 2.8 |
| COX-2 | 8.7±10.6 | 3.4± 1.9 | 3.4±2.5 | 6.7±10.0 | 5.5±5.0 | 2.1± 2.6 |
| w/d/l | | | | 7 / 0 / 3 | 5 / 0 / 5 | 8 / 0 / 2 |

Table 5.9: Mean Δ_{EF} and standard deviation for the ECFP and ECFP_{ext} similarity methods at 1%, 5% and 10% of the database (receptor specific decoy set DuD_{set}). The extension fingerprint is calculated from all ligands (approach B1). Improvements of ECFP_{ext} compared to ECFP are marked with bold print. w/d/l = wins/draws/losses.

Our final results on the DuD data sets concern the question whether the method is really sensitive against the choice of a suitable α . For this purpose, we present the win/loss statistics for a wide range of α values (from 0.0 to 1.0 with a step size of 0.1), across all the data sets from cross-validation in Tables B.10 and B.11. Quite surprisingly, the choice

of a value of α does not appear to have a strong influence on the win/loss statistics. The proportion of roughly 8:2 or 9:1 still holds in this experiment. Therefore, we may conclude that the method is reasonably robust regarding the choice of a suitable value for α .

Comparing the data mining based extension results for the HMGR data set (first rows denoted HMGR in Tables 5.8 and 5.9) with the by-hand results on HMGR in Table 5.6 (rows denoted "lig vs DuD_{set}"), we see that the Δ_{EF} values are slightly better for the by-hand extension, but both variants of the data mining based approach are quite competitive. The ECFP_{ext} results of variant B1 are even better than the by-hand results.

As final experiments to test our data-mining based approaches B1 and B2 we added 25 ChEMBL activity class data sets. The results for approach B1 and B2 are given in Tables B.7 and B.8 respectively. For those data sets the win counts over all data sets are 19, 21, 21 and 18, 22, 22 (of 25 maximum possible) for 1%, 5% and 10% of the database and MCS_{ext} and ECFP_{ext}. According to the sign test the difference between extended and non-extended similarities is significant at a level of 0.05 [28].

| DuD set | bind_fp | | |
|---------------|------------------------|------------------------|------------------------|
| | 1% | 5% | 10% |
| HMGA | 6.7±1.5 ^{• °} | 1.1±0.0 ^{• °} | 0.6±0.0 ^{• °} |
| ER | 62.3± 15.2 | 8.7± 4.1 [°] | 2.7±1.6 ^{• °} |
| PPAR γ | 13.2± 30.1 | 2.4± 6.1 | 1.2± 3.0 [°] |
| P38 MAP | 24.2± 22.2 | 4.8± 4.3 [°] | 2.4±2.1 ^{• °} |
| TK | 42.8± 24.3 | 0.9±1.5 ^{• °} | 0.0±0.0 ^{• °} |
| Fxa | 21.2± 26.9 | 3.8± 5.5 [°] | 1.8± 2.7 [°] |
| ADA | 26.1± 0.0 | 1.7±0.0 ^{• °} | 0.5±0.1 ^{• °} |
| DHFR | 0.0±0.0 ^{• °} | 0.0±0.0 ^{• °} | 0.0±0.0 ^{• °} |
| AchE | 47.4± 34.8 | 7.7± 6.0 [°] | 3.8±3.0 ^{• °} |
| COX-2 | 71.6± 22.6 | 10.2± 6.3 [°] | 2.2±2.6 ^{• °} |

Table 5.10: Mean Δ_{EF} and standard deviation for the bind_fp similarity method at 1%, 5% and 10% of the database (receptor specific decoy set DuD_{set}). The extension fingerprint is calculated from all ligands (approach B1). Cases where bind_fp is better than ECFP or MCS are marked with a [•] or [°], respectively.

| DuD set | ECFP _{ext} | | | MCS _{ext} | | |
|---------------|---------------------|-----|-----|--------------------|-----|-----|
| | 1% | 5% | 10% | 1% | 5% | 10% |
| HMGR | 0.2 | 0.5 | 0.9 | 0.3 | 0.7 | 0.8 |
| ER | 0.1 | 0.3 | 0.2 | 0.6 | 0.6 | 0.5 |
| PPAR γ | 0.1 | 0.4 | 0.0 | 0.0 | 0.4 | 0.0 |
| P38 MAP | 0.3 | 0.5 | 1.0 | 0.7 | 0.8 | 1.0 |
| TK | 0.0 | 0.1 | 0.3 | 0.3 | 1.0 | 0.2 |
| Fxa | 0.0 | 0.3 | 0.2 | 0.2 | 0.7 | 0.6 |
| ADA | 0.3 | 1.0 | 1.0 | 0.3 | 1.0 | 1.0 |
| DHFR | 0.1 | 0.3 | 0.5 | 0.4 | 1.0 | 0.6 |
| AchE | 0.2 | 0.3 | 0.1 | 0.2 | 0.3 | 0.4 |
| COX-2 | 0.1 | 0.2 | 0.2 | 0.1 | 1.0 | 0.6 |

Table 5.11: The best α coefficients for the MCS_{ext} and ECFP_{ext} similarity methods. α has been increased from 0.0 to 1.0 in steps of 0.1. The coefficient giving the best Δ_{EF} value is reported. If two values are identical the smaller α is reported.

| DuD set | MCS _{ext} | | | | ECFP _{ext} | | | |
|---------------|--------------------|------------------|-------------------|-----|---------------------|------------------|------------------|-----|
| | 1% | 5% | 10% | | 1% | 5% | 10% | |
| HMGR | 5.8 ±10.0 | 1.7 ±0.7 | 0.6 ± 0.3 | 0.3 | 0.6 ± 1.9 | 2.6 ±3.2 | 0.6 ± 0.1 | 0.8 |
| ER | 12.1 ± 6.0 | 9.3 ±3.5 | 3.2 ± 1.1 | 1.1 | 6.0 ± 5.3 | 8.5±2.1 | 3.1 ± 1.2 | 0.5 |
| PPAR γ | 4.5 ±10.6 | 3.8±5.5 | 1.5 ± 2.9 | 2.9 | 4.1±10.7 | 3.6 ±5.6 | 1.7± 2.5 | 0.0 |
| P38 MAP | 2.8 ± 6.9 | 4.8 ±4.2 | 2.4 ± 2.1 | 2.1 | 2.7 ± 6.0 | 4.8 ±4.2 | 2.4 ± 2.1 | 1.0 |
| TK | 18.3 ± 5.3 | 11.1 ±3.8 | 3.7 ± 1.7 | 1.7 | 16.5± 8.4 | 11.1 ±3.8 | 4.2 ± 1.9 | 0.2 |
| FXa | 3.5 ±11.0 | 4.3 ±5.4 | 2.0 ± 2.7 | 2.7 | 3.5 ±11.0 | 4.2±5.4 | 2.0± 2.7 | 0.6 |
| ADA | 9.2 ± 4.6 | 5.2 ±0.0 | 2.2 ± 0.8 | 0.8 | 7.8 ± 7.5 | 5.2 ±0.0 | 2.2 ± 0.7 | 1.0 |
| DHFR | 2.7 ± 5.1 | 0.0 ±0.0 | 0.0 ± 0.0 | 0.0 | 1.9 ± 0.9 | 0.0 ±0.0 | 0.0 ± 0.0 | 0.6 |
| ACHE | 10.0 ±11.8 | 9.0 ±6.0 | 4.3 ± 2.8 | 2.8 | 11.0 ±12.0 | 9.0±6.1 | 4.0 ± 2.8 | 0.5 |
| COX-2 | 9.9 ± 9.8 | 9.8 ±3.7 | 2.1 ± 2.6 | 2.6 | 6.7 ±10.2 | 5.3±5.0 | 2.1 ± 2.6 | 0.6 |
| w/d/l | 10 / 0 / 0 | 9 / 0 / 1 | 10 / 0 / 0 | | 8 / 0 / 2 | 6 / 0 / 4 | 8 / 0 / 2 | |

Table 5.12: Mean Δ_{EF} and standard deviation using the best α coefficients for extended similarities MCS_{ext} and ECFP_{ext} for the receptor specific decoy sets DuD_{set} at 1%, 5% and 10% of the database. The extension fingerprint is calculated from all ligands (approach B1). Improvements of MCS_{ext} compared to MCS as well as ECFP_{ext} compared to ECFP are marked in bold print. w/d/l = wins/draws/losses.

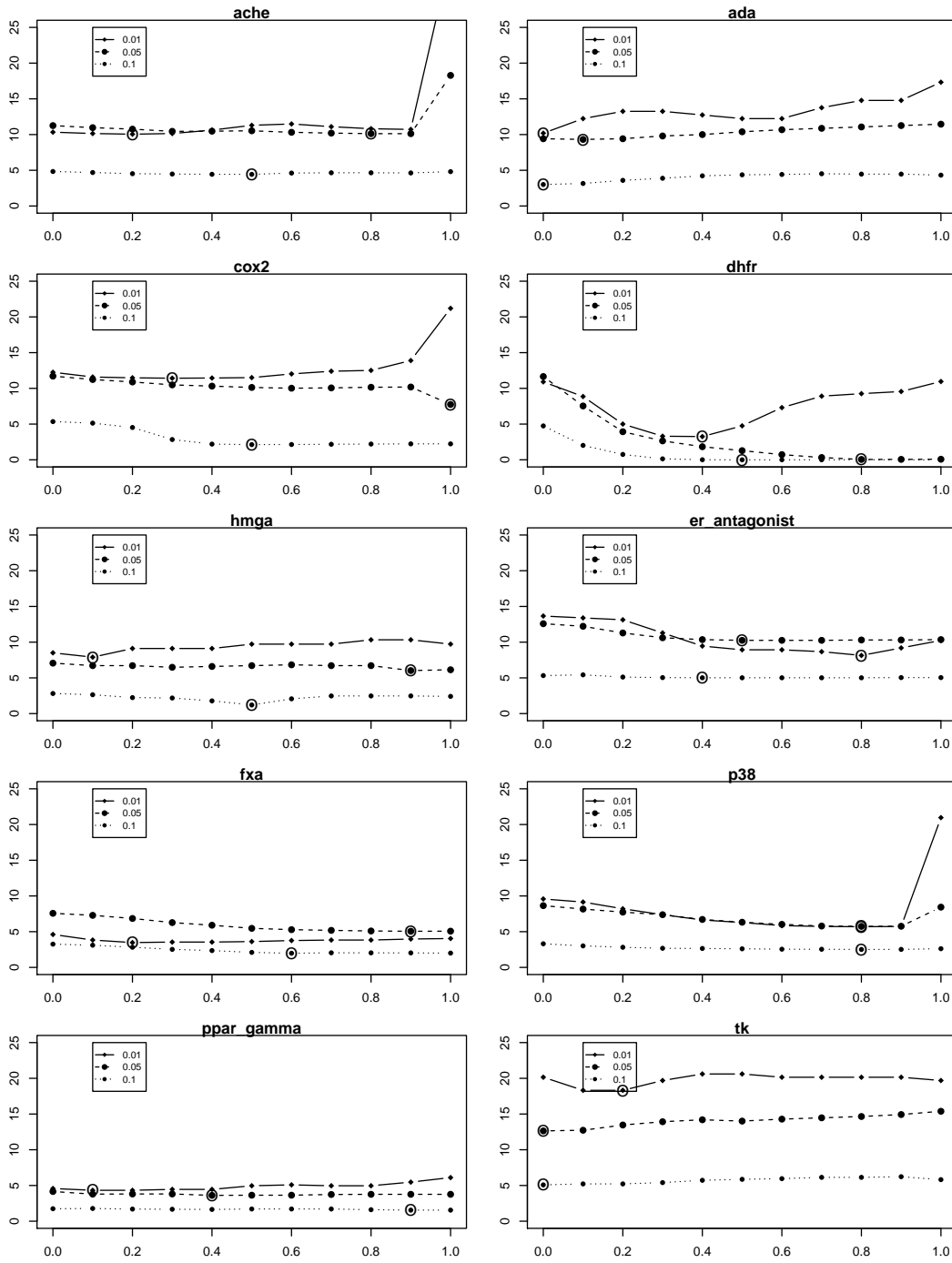


Figure 5.7: Plot of α vs. Mean Δ_{EF} for MCS_{ext} . On the x-axis the values of the combining factor α is plotted versus the mean Δ_{EF} for MCS_{ext} on the y-axis. (approach B2)

| DuD set | MCS _{ext} | | | ECFP _{ext} | | |
|-------------------|--------------------|------------------|------------------|---------------------|------------------|------------------|
| | 1% | 5% | 10% | 1% | 5% | 10% |
| HMGR | 6.1 ±10.5 | 2.1 ±0.9 | 0.8 ± 0.4 | 2.7 ± 6.8 | 4.3 ±5.6 | 1.5 ± 2.5 |
| ER | 8.1 ± 6.1 | 10.4 ±2.9 | 4.6 ± 1.5 | 6.3 ± 5.4 | 10.0±2.0 | 4.2 ± 1.3 |
| PPAR _γ | 4.6±10.6 | 3.9±5.5 | 1.7± 2.8 | 4.1±10.7 | 3.5 ±5.6 | 1.7± 2.5 |
| P38 MAP | 5.4 ± 6.4 | 5.4 ±3.4 | 1.4 ± 0.1 | 3.9 ± 5.7 | 6.7±3.8 | 1.4 ± 0.1 |
| TK | 17.4 ± 5.2 | 11.4 ±4.8 | 4.7 ± 1.6 | 16.5± 8.4 | 11.4 ±3.5 | 4.6 ± 2.1 |
| FXa | 3.5 ±11.2 | 5.7 ±5.1 | 2.5 ± 2.6 | 3.5 ±11.0 | 5.0±5.2 | 2.2± 2.6 |
| ADA | 9.7 ± 5.4 | 7.2 ±2.6 | 2.4 ± 1.1 | 7.3 ± 7.2 | 6.9±2.7 | 2.4 ± 1.0 |
| DHFR | 3.0 ± 6.0 | 0.8 ±1.6 | 0.0 ± 0.0 | 2.4 ± 1.2 | 0.4 ±0.9 | 0.0 ± 0.0 |
| ACHE | 10.1 ±12.0 | 9.6 ±5.8 | 4.5 ± 2.9 | 11.2 ±12.2 | 9.4±5.9 | 4.4 ± 2.8 |
| COX-2 | 12.0 ±10.3 | 10.8 ±6.3 | 2.8 ± 2.8 | 6.7 ±10.3 | 5.5±4.9 | 2.2 ± 2.6 |
| w/d/l | 9 / 1 / 0 | 9 / 0 / 1 | 9 / 1 / 0 | 8 / 0 / 2 | 4 / 0 / 6 | 8 / 0 / 2 |

Table 5.13: Mean Δ_{EF} and standard deviation using the best α coefficients for extended similarities MCS_{ext} and ECFP_{ext} for the receptor specific decoy sets DuD_{set} at 1%, 5% and 10% of the database. The extension fingerprint is calculated from 10% (20% for HMGR, TK and ADA) of the ligands (approach B2). Improvements of MCS_{ext} compared to MCS as well as ECFP_{ext} compared to ECFP are marked in bold print. w/d/l = wins/draws/losses.

| | ECFP _{ext} | | | | | | MCS _{ext} | | | | | |
|-------------------|---------------------|------|-----|------|-----|------|--------------------|------|-----|------|-----|------|
| | 1% | | 5% | | 10% | | 1% | | 5% | | 10% | |
| | win | loss | win | loss | win | loss | win | loss | win | loss | win | loss |
| HMGR | 10 | 0 | 9 | 1 | 9 | 1 | 5 | 0 | 5 | 0 | 5 | 0 |
| ER | 8 | 2 | 9 | 1 | 7 | 3 | 10 | 0 | 9 | 1 | 7 | 3 |
| PPAR _γ | 10 | 0 | 9 | 1 | 9 | 1 | 6 | 4 | 10 | 0 | 8 | 2 |
| P38 MAP | 9 | 1 | 10 | 0 | 9 | 1 | 9 | 1 | 9 | 1 | 10 | 0 |
| TK | 7 | 3 | 9 | 1 | 9 | 1 | 9 | 1 | 6 | 4 | 7 | 3 |
| FXa | 10 | 0 | 7 | 3 | 8 | 2 | 9 | 1 | 8 | 2 | 8 | 2 |
| ADA | 10 | 0 | 6 | 4 | 9 | 1 | 6 | 3 | 8 | 1 | 6 | 3 |
| DHFR | 7 | 3 | 10 | 0 | 10 | 0 | 10 | 0 | 10 | 0 | 10 | 0 |
| ACHE | 8 | 2 | 9 | 1 | 9 | 1 | 8 | 2 | 10 | 0 | 9 | 1 |
| COX-2 | 8 | 2 | 9 | 1 | 7 | 3 | 7 | 3 | 8 | 2 | 8 | 2 |
| sum | 87 | 13 | 87 | 13 | 86 | 14 | 79 | 15 | 83 | 11 | 78 | 16 |

Table 5.14: Win/Loss counts for all ten random folds for extended similarities MCS_{ext} and ECFP_{ext} versus their base similarities MCS and ECFP for the receptor specific decoy sets DuD_{set} at 1%, 5% and 10% of the database. The extension fingerprint is calculated from 10% (20% for HMGR, TK and ADA) of the ligands (approach B2).

5.3 Conclusions

Structural similarity measures, especially the ECFP fingerprints, have been reported to be superior to non-substructural fingerprints [58]. This chapter shows that and how such structural similarity methods used in virtual screening can be improved further by integrating background knowledge on binding-relevant structural features. We presented an approach based on by-hand selection of the background knowledge as well as an approach working with fragment-based data mining. From our experimental evaluation we conclude that the addition of only one binding-relevant sub-structural feature of a known ligand can substantially improve the enrichment factors in the virtual screening. We additionally show that using data mining based knowledge extraction instead of time consuming by-hand selection of relevant features gives competitive results.

CHAPTER 6

Adapted Transfer of Distance Measures for Quantitative Structure-Activity Relationships and Data-Driven Selection of Source Datasets

Quantitative structure-activity relationships (QSARs) are models quantitatively correlating chemical structure with biological activity or chemical reactivity. In technical and statistical terms, QSARs are often regression models on graphs (molecular structures being modeled as graphs). QSARs and small molecules are subject of very active research in data mining [60, 123]. The task is often tackled by instance-based and distance-based methods, which predict biological activity based on the similarity of structures. As the success of those methods critically depends on the availability of a suitable distance measure, it would be desirable to automatically determine a measure that works well for a given dataset and endpoint¹⁹. Recently proposed solutions for other, related problems (general classification problems instead of domain-specific regression problems as discussed here) include distance learning methods [48] and methods from inductive transfer [32]. In distance learning, the distance measures (e.g., parameterized distances like the Mahalanobis distance) are directly learned from labeled training distances. Inductive transfer is concerned with transferring the bias of one learning task to another, related task.

In this chapter, we propose adapted transfer, a combination of distance learning and inductive transfer. We learn the contributions of the distances on a task related to our problem and then transfer them to our learning task at hand. The approach is evaluated specifically for QSAR problems (regression on graphs). In the experiments, we investigate how adapted transfer performs compared to distance learning or inductive transfer alone, depending on the size of the available training set. These questions are studied using five pairs of distinct datasets, each consisting of two datasets of related problems.

In addition, we present an approach that rids us of the assumption that we know a related task that we can use as source task for transfer. We select the related dataset in a data-driven way from the PubChem BioAssay [147] database. The activity overlap similarity of two datasets is applied to find a suitable source task. This approach is

¹⁹ In pharmacology and toxicology, an endpoint constitutes the target outcome of a trial or experiment.

evaluated on five distinct datasets, for which we first find an appropriate source dataset and then use the same set of experiments that we use when hand-selecting the source task.

For the distance measures, our starting point is the observation by Raymond and Willett [105] that maximum common subgraph (MCS) based measures and fingerprint-based measures provide orthogonal information and thus should be considered as complementary. The reason for this may be that MCS-based measures aim to quantify the *global* similarity of structures, whereas fingerprint-based measures rather quantify *local* similarity in terms of smaller, common substructures. We devised an approach that optimally combines the contributions of the two types of measures, and thus balances the importance of global and local similarity for chemical structures.

This chapter is organized as follows: In the next section, we present the technical details of learning and adapting distance measures for QSAR problems. Then the datasets, preprocessing steps and the experimental setup are explained before we present the results of the experimental evaluation. This is followed by the introduction and evaluation of our approach to select the source datasets in a data-driven way. We experimentally compare the approach to the *Boosting for Regression Transfer (TrAdaBoost.R2)* [98] method before we give conclusions in the last section.

6.1 Distance Learning, Inductive Transfer and Adapted Transfer

We frame the learning problem as follows. We are given a set of n labeled examples $X := \{(x_1, y_1), \dots, (x_n, y_n)\}$, where the examples $x_i \in \mathcal{X}$ are arbitrary objects taken from an instance space \mathcal{X} and the $y_i \in \mathbb{R}$ are real-valued target labels. For the learning setting, we aim at finding a regression function $r : \mathcal{X} \rightarrow \mathbb{R}$ that predicts the target label well on new unseen data. We measure the accuracy of a predictor, by taking the squared difference between the predicted target label y' and the true target label y . In other words, we evaluate a prediction using the *squared loss* $l_2(y, y') = (y - y')^2$. We also assume that we have a *distance function* $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ at our disposal, which quantifies the distance between two instances. More precisely, we demand that $d(x, x) = 0$ and that $d(x_1, x_2) < d(x_1, x_3)$, if x_2 is more similar to x_1 than x_3 . For ease of notation, we store the distances between all training examples in one $n \times n$ matrix D , so that $D = [d_{ij}] = d(x_i, x_j)$. One well known way to perform regression with distance functions is the *k-nearest neighbor* rule. Given an unlabeled example x , one determines the k nearest neighbors in the training data according to the distance function and then predicts the average over the k neighbors. Let $Y := (y_1, \dots, y_n)^T$ denote the target label vector and let $W = [w_{ij}]$ be a $n \times n$ *neighbor matrix* that has

$$w_{ij} = \begin{cases} \frac{1}{k} & \text{if } x_j \in k \text{ nearest neighbors of } x_i \\ 0 & \text{otherwise.} \end{cases}$$

With this, the vector of predicted target labels of the training instances is simply $\hat{Y} := WY$.

Our main contribution, the adapted transfer, is based on the two building blocks: distance learning and inductive transfer. Therefore, we introduce the building blocks first, and describe our main contribution subsequently. In the following we deal with settings, where we have more than one distance function to rate the distance between examples. Rather than restricting ourselves to one fixed function, we would like to use all available information for the prediction by combining the m distance functions d_1, \dots, d_m . One simple way to do so is to take the average: $\hat{Y} = \frac{1}{m} \sum_{i=1}^m W_i Y$. In practice, however, one will often encounter settings, in which some distances provide better information than the others. In such settings it makes sense to use a *weighted average* $\hat{Y} = \sum_{i=1}^m \alpha_i W_i Y$, where the weight vector $\alpha = (\alpha_1, \dots, \alpha_m)^T \in \mathbb{R}^m$ with $\sum_{i=1}^m \alpha_i = 1$ specifies to which extent each distance function contributes to the prediction. If we aim at low empirical error on the training set, we can determine the optimal α by minimizing the squared error on the training set:

$$\begin{aligned} \alpha^* &:= \operatorname{argmin}_{\alpha} \left\| \sum_{i=1}^m \alpha_i W_i Y - Y \right\|^2 & (6.1) \\ \text{subject to } & \sum_{i=1}^m \alpha_i = 1 \\ & 0 \leq \alpha_i \leq 1 \text{ for } i = 1, \dots, m \end{aligned}$$

This is a standard quadratic program with linear constraints and can be solved efficiently by standard convex optimization software.

To extend this setting, we use a different nearest neighbor matrix W than that of the standard k -nearest neighbor approach. In the original definition this matrix makes a hard cut: the first k neighbors contribute equally to the prediction, whereas the remaining examples are ignored. This appears to be a somewhat arbitrary choice and one can envision many more fine-grained and less restrictive prediction schemes. In principle, a matrix W must fulfill two properties in order to lead to reasonable predictions: its rows must sum to one and it must assign larger weights to more similar instances. In our experiments we used a nearest neighbor approach with a distance threshold. Instead of choosing a fixed number k of nearest neighbors, one selects a distance threshold t and determines the set T of all neighbors whose distance to the test example is less than t . Each example in T influences the prediction with weight $\frac{1}{|T|}$.

Our second building block is *inductive transfer*. Inductive transfer is suitable for settings where the amount of available training data is too small to determine a good weight vector. Instead of learning a completely new weight vector α from the (limited) target training data, we make use of an additional dataset, which is assumed to have similar characteristics as the target data. We call this additional dataset the *source dataset* to distinguish it from the *target training set* so that the inductive transfer takes place from source to target. In

the *Simple Transfer* setting, one induces a weight vector β only from the source data (by solving (6.1) for the source dataset) and uses this β without modification for the actual prediction. The actual training data provides the neighbors for the prediction, but is not used for the computation of α .

Enhancing this *Simple Transfer* setting, the *Adapted Transfer* setting allows for the transferred weight vector β to be adapted to the target training data. This can be done in two ways:

- ▷ *Bounded Adaptation*. One induces a weight vector β from the source data, but adapts it in a second step slightly to the target training data. For the adaptation step, we would like to avoid overfitting on the (limited) training data. Thus, we extend the optimization criterion (6.1) with the additional criterion that the α may not differ too much from the transferred β . More precisely, for a fixed $\epsilon > 0$ we compute

$$\alpha^* := \operatorname{argmin}_{\alpha} \left\| \sum_{i=1}^m \alpha_i W_i Y - Y \right\|^2 \quad (6.2)$$

subject to $\sum_{i=1}^m \alpha_i = 1$

$|\alpha_i - \beta_i| \leq \epsilon$ for $i = 1, \dots, m$

$0 \leq \alpha_i \leq 1$ for $i = 1, \dots, m$

- ▷ *Penalized Adaptation*. In this approach, we also adapt the weight vector β induced from the source data. Instead of limiting the interval from which the α can be taken, we add a regularization term to the optimization criterion that penalizes α s that deviate too much from β . Formally, for $C > 0$, we solve

$$\alpha^* := \operatorname{argmin}_{\alpha} \left\| \sum_{i=1}^m \alpha_i W_i Y - Y \right\|^2 + C \|\alpha - \beta\|^2 \quad (6.3)$$

subject to $\sum_{i=1}^m \alpha_i = 1$

$0 \leq \alpha_i \leq 1$ for $i = 1, \dots, m$

In the following sections, these variants will be evaluated and tested experimentally.

6.2 Data and Experimental Setup

All of the datasets used in Section 6.3 were taken from the data section of the cheminformatics web repository²⁰. Since we are interested in adapted transfer between different

²⁰ <http://www.cheminformatics.org>

datasets, we put a special focus on finding pairs of datasets with similar or identical endpoints. Note that due to the wealth of data produced in all areas of science and industry today, the existence of related datasets is frequently occurring and thus practically relevant. In fact, even in computational chemistry, the five pairs used here are just a selection from a wider range of possibilities.

For the first pair of datasets [131, 132] abbreviated DHFR_4q (361 compounds) and DHFR_S. (673), the goal is to predict the dihydrofolate reductase inhibition of compounds as measured by the pIC_{50} value that indicates how much of a given substance is needed to block a biological activity by half. We had to remove a number of instances, which were considered to be inactive in the original publication and marked with default values. Overall the compounds in this pair of datasets share a high similarity. Consequently, there are often only local changes to the molecular graph structure and the graphs are very similar on a global level. The second pair, CPDB_m (444, mouse) and CPDB_r (580, rat) are generated from data obtained by the carcinogenic potency project²¹. The compounds' carcinogenicity is rated according to the molar TD_{50} value $\text{TD}_{50}^{\text{m}}$, where a low value indicates a potent carcinogen. The two datasets contained several instances where the actual structure of the compound was missing. If the molecule could be identified uniquely, we downloaded the structure from the NCBI PubChem database²². If this was not possible, the molecule was removed from the set. The third pair of datasets [132], ER_TOX (410) and ER_LIT (381), measure the logarithmized relative binding affinities (RBA) of compounds to the estrogen receptor with respect to β -estradiol. All inactive compounds were removed from the datasets as they all have the same value. The fourth pair, ISS_m (318, mouse) and ISS_r (376, rat)[9], is similar to the second pair. The target value under consideration is again the carcinogenic potency of a compound as measured by the molar TD_{50} value. The two datasets contained several instances where the actual structure of the compound was missing. If the molecule could be identified uniquely via the given CAS number, we downloaded the structure from the NCBI PubChem database. If this was not possible, the molecule was removed from the set. The fifth and last pair of datasets [132, 131], COX2_4q (282) and COX2_S. (414), are used to predict the cyclooxygenase-2 inhibition of compounds as measured by the pIC_{50} value. We had to remove a number of instances, which were considered to be inactive in the original publication and marked with default values. As in the first pair of datasets, the compounds contained in this dataset pair are highly similar. The preprocessed and cleaned datasets used in our experiments are available for download on the authors' website²³.

21 <http://potency.berkeley.edu/chemicalsummary.html>

22 <http://pubchem.ncbi.nlm.nih.gov/>

23 <http://infosys.informatik.uni-mainz.de/research>

6.2.1 Distances

For all datasets, we generated three different distance matrices. The first and the second matrix are based on a distance measure for binary fingerprints closely related to the Tanimoto similarity measure (2.1).

The first set of fingerprints are occurrence fingerprints for frequently occurring substructures. The substructures are closed free trees that were calculated with the Free Tree Miner (FTM) [113] software. The frequency threshold was set individually for each dataset so that approximately 1000 free trees were found (see Table 6.1). Moreover, we reduced the solution space of the free trees by computing closed free trees. A free tree is closed if any free tree that is more specific, meaning that it is larger and contains the other, has a lower frequency. We calculated closed features to reduce the number of free trees, to remove redundancy, and finally to make the solution space sparser in order to remove dependencies among its elements. For the calculation of closed free trees, we applied the iSAR software package written by Selina Sommer [125]. The second fingerprint set is built of pharmacophoric (binary) fingerprints computed with the cheminformatics library JOELIB2²⁴. In this way, we obtained a chemical description in the form of a pharmacophoric (binary) fingerprint containing more than 50 chemical descriptors. For each instance, the existence of single atoms, functional subgroups or stereochemical features is tested. Binary fingerprint vectors are calculated as for the free trees and used in the above Tanimoto-based distance measure. The third distance matrix is based on a Tanimoto-like maximum common subgraph (MCS) based distance measure (2.5). JChem Java classes were used for computing the maximum common subgraph (MCS), JChem 5.1.3_2, 2008, ChemAxon (<http://www.chemaxon.com>).

Table 6.1: Overview of the datasets and the minimum support threshold *minsup* set for FTM and of the number of free trees and closed free trees. size: number of instances in the dataset, fts: frequent free trees, cfts: frequent closed free trees.

| Dataset | size | minsup | fts | cfts |
|----------------|------|--------|-----|------|
| <i>COX2_4q</i> | 282 | 0.35 | 957 | 254 |
| <i>COX2_S.</i> | 414 | 0.32 | 914 | 251 |
| <i>DHFR_4q</i> | 361 | 0.27 | 866 | 231 |
| <i>DHFR_S.</i> | 673 | 0.24 | 953 | 265 |
| <i>ISS_m</i> | 318 | 0.04 | 764 | 412 |
| <i>ISS_r</i> | 376 | 0.05 | 742 | 315 |
| <i>CPDB_m</i> | 444 | 0.04 | 727 | 420 |
| <i>CPDB_r</i> | 580 | 0.04 | 866 | 477 |
| <i>ER_TOX</i> | 410 | 0.19 | 818 | 368 |
| <i>ER_LIT</i> | 381 | 0.67 | 943 | 194 |

²⁴ <http://www-ra.informatik.uni-tuebingen.de/software/joelib>

6.3 Experiments

We consider a QSAR learning task given by a *target* training and test set. Additionally, we assume that we can transfer information from a *source* dataset containing related training data. The task is to induce a predictor from the target training set and the source dataset, which features good predictive accuracy on the test set. To solve this task we propose the strategy *adapted transfer*. This approach combines adaptation and inductive transfer, as outlined in the second section. We start by identifying the weight vector α^* that optimizes (6.1) on the source dataset. Instead of using this weight vector directly, we adapt it to better match with the target training data. This is done either with bounded adaptation by optimizing (6.2) or with penalized adaptation by solving (6.3). The resulting weight vector is then applied to the target training data in a nearest-neighbor classifier.

To get reliable results, we repeat our experiments one hundred times, where each run consists of a ten-fold cross-validation. This means we estimate the methods' success on one thousand different configurations of training- and test-folds. To quantify predictive accuracy, we choose mean squared error, a standard measure in regression settings. We evaluate the adapted transfer approaches against three baseline strategies:

- ▷ **Best single distance.** We perform an internal 5-fold cross-validation on the target training set to determine the best of the three distances. This distance is then used to predict the target values for the target test set. The source dataset is not used.
- ▷ **Distance learning.** We compute the solution to the optimization problem (6.1) on

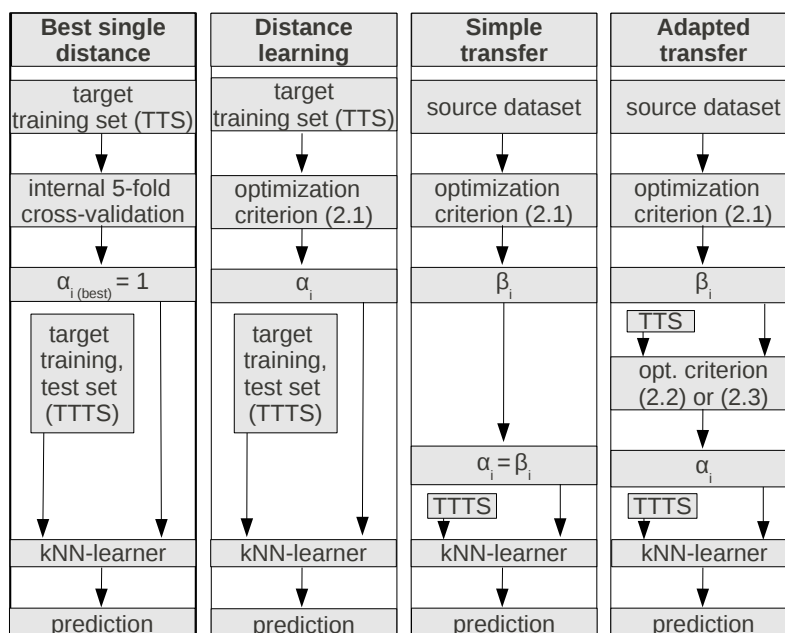


Figure 6.1: Graphical overview of the four strategies used in the experiments. Abbreviations: opt. = optimization, $\alpha_{i(\text{best})} = \alpha_i$ for best single distance.

determine the best linear combination α on the target training set. The weighted combination of distances is then used as new distance for the prediction on the test data. Again, the source dataset is not used.

- ▷ **Simple transfer.** Here, we optimize (6.1) on the source dataset instead of the target training data. The weighted combination of distances is then used as new distance for the prediction on the test data. Here, the target training data is only used for the nearest-neighbor classifier, not for the adaptation of the distance measure.

All four strategies are illustrated in Figure 6.1. All algorithms and methods were implemented in MATLAB Version 7.4.0.336 (R2007a). We applied the MOSEK²⁵ Optimization Software (Version 5.0.0.60) that is designed to solve large-scale mathematical optimization problems. The Matlab source code used for our experiments is available for download on the authors' website²⁶.

6.3.1 Learning Curves

At their core, distance adaptation and inductive transfer methods are approaches to improve predictive accuracy by fine-tuning the learning bias of a machine learning scheme. Both can be expected to make a difference only if there is not enough target training data available to obtain a good predictor. If this is not the case and there is sufficient training data available, most reasonable learning approaches will find good predictors anyway, and distance adaptation or inductive transfer cannot improve its predictive accuracy significantly. To evaluate this trade-off between the amount of available training data and the applicability of transfer and adaptation approaches, we first present *learning curves* rather than point estimates of a predictor's accuracy for a fixed training set size. More precisely, we repeat each experiment with increasing subsets of the original target training data. We start by using only the first 10%, then 20%, and so on until the complete training data is available. The corresponding learning curves are given in Figures 6.2, 6.3 and 6.4 for one representative parameter setting producing typical results (nearest neighbor with distance threshold $t = 0.2$, $\epsilon = 0.2$ for the bounded adaptation and $C = 10.0$ for the penalized adaptation; No in-depth, systematic analysis of the impact of changing those parameters on the performance has been conducted). While the differences appear to level off for increasing training set sizes, there are clearly differences at the beginning of the learning curves. The single best distance is outperformed by other methods (outside the scale of the y-axis for CPDB and ISS), and distance learning does not work well yet for small training set sizes (except for ISS_r). For the ISS_r dataset the MSE at 10% and 20% is unusually low. This could be attributed to overfitting. For a more principled comparison, we now examine under which circumstances one approach outperforms another significantly.

²⁵ MOSEK ApS, Denmark. <http://www.mosek.com>

²⁶ <http://infosys.informatik.uni-mainz.de/research>

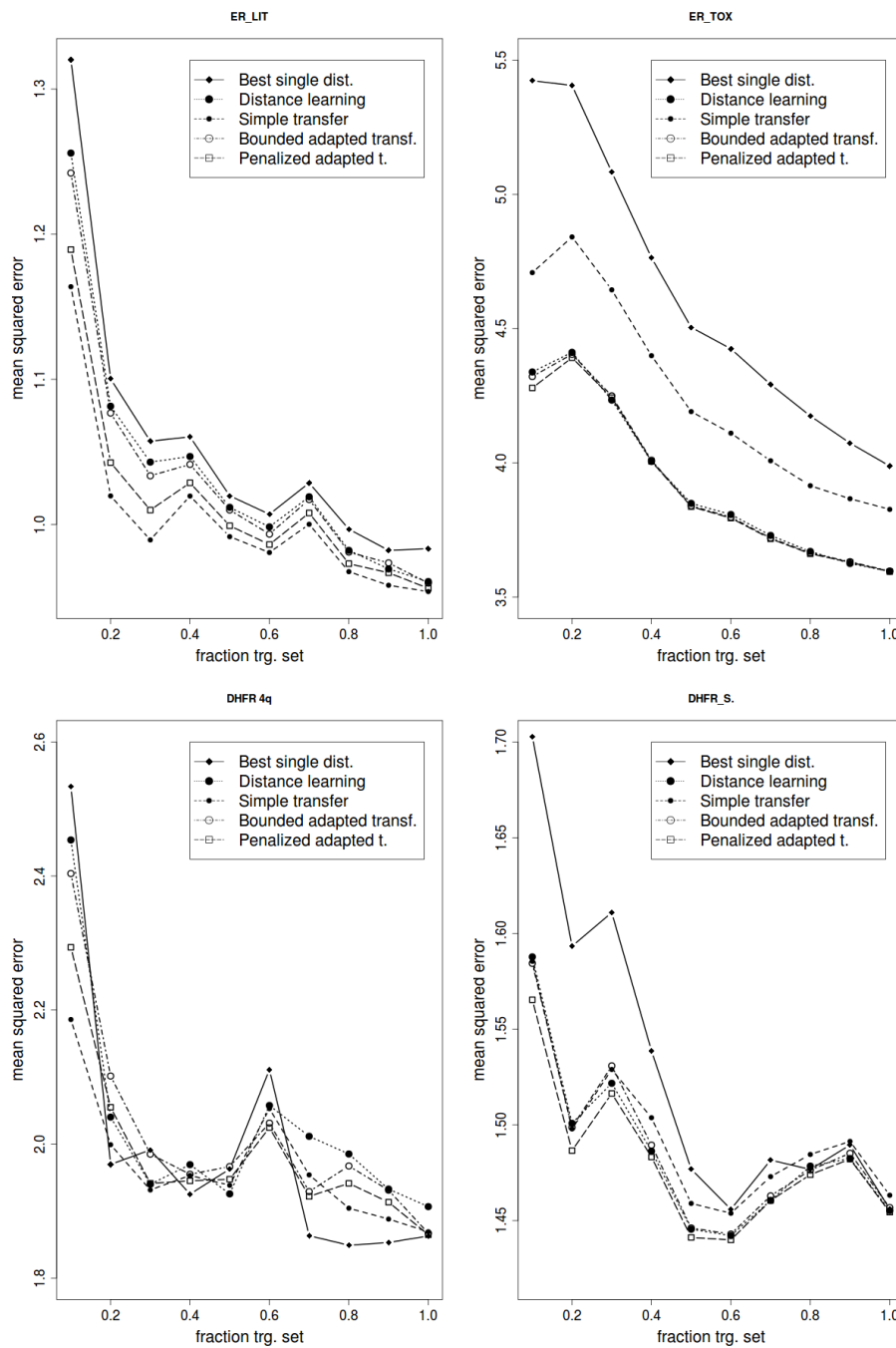


Figure 6.2: Learning curves for nearest neighbor with dist. thr. $t = 0.2$.

6.3.2 Comparison of Approaches

To investigate whether our adapted transfer strategy outperforms the presented baseline approaches we first evaluate how the baseline approaches perform compared to each other. The corresponding results are shown in Table 6.2. Second, we investigate the performance of the adapted transfer strategy (results shown in Table 6.3). We conducted one hundred runs of tenfold cross-validation. For each run, we noted whether the first or the second

method performed better. We tested if the resulting set of predictive accuracies were statistically significant improvements or deteriorations at a significance level of 5% using Matlab’s implementation of the paired-sample t-test.

As a first experiment, we would like to address the question whether distance learning improves predictive accuracy. More precisely, we compare whether having a linear combination of the distances’ contributions optimizing criterion (6.1) (distance learning strategy) outperforms the single best distance. The results are given in the first section of Table 6.2. It shows that distance adaptation using a linear combination significantly outperforms the single best distance in nine out of ten cases for 10% and eight out of ten cases for 100%. Thus, there is some empirical evidence that a learning method, which adjusts its bias to better accommodate for the underlying data, is more successful than an approach with a single fixed bias (i.e. distance). The result also supports the observation by Raymond and Willett [105] that maximum common subgraph based measures and fingerprint-based measures provide orthogonal information.

For the second experiment we would like to investigate whether inductive transfer improves predictive accuracy. To do so, we compare the best single distance strategy with the simple transfer where the weights for the linear combination are computed on the source dataset rather than the target training data. Our experiments indicate (second section of Table 6.2) that inductive transfer using a linear combination significantly outperforms the single best distance in all cases for 10% and seven out of ten cases for 100%. Apparently, inductive transfer has the same effect as distance learning.

Since both building blocks of our adapted transfer strategy, distance learning and inductive transfer, improve predictive accuracy, the next experiment deals with the question under which circumstances one approach outperforms the other. The experiments indicate (third section of Table 6.2) that inductive transfer is significantly better than distance learning, if only few training data are available, but the opposite is true, if all the available training data is used. Hence, one can say that one should resort to inductive transfer methods, whenever there is comparably few training data available and when the source data for the transfer is of sufficiently good quality (as appears to be the case on all datasets except for ER_TOX, ISS_r and COX2_Sutherland). Unfortunately, it is often hard to tell in advance, whether the source data is good enough for successful transfer and how the size of the available target data compares to the size of source data of unknown quality.

In order to avoid this problem, we introduced a “mixed strategy”, which transfers weights from the source dataset, but ensures that the actual weights differ not too much from the ones which can be obtained by distance adaptation on the target data. We now compare the penalized adapted transfer approach to its two building block baseline methods. The results in the first section of Table 6.3 show that adapted transfer outperforms distance learning on small training data, but leads to no further improvement, if there is sufficient amount of training data. On the other hand, the mixed strategy performs better than simple transfer in settings with large amounts of training data. When only few training data is present, its performance is sometimes better and sometimes worse than

Table 6.2: Distance learning vs. Best single distance. A vs. B: ● = A significantly better than B, ○ = A significantly worse than B; w = how often out of 100 times A “wins” against B.

| Distance learning vs. Best single distance | | | | |
|---|-----|--------------|------|--------------|
| frac. trg. set | 10% | | 100% | |
| Short-hand | w | p-value | w | p-value |
| <i>DHFR_4q</i> | 39 | 0.2445 | 17 | 8.9928e-15 ○ |
| <i>DHFR_S.</i> | 74 | 7.3586e-10 ● | 50 | 0.9916 |
| <i>CPDB_m</i> | 100 | 3.8397e-50 ● | 100 | 5.0502e-62 ● |
| <i>CPDB_r</i> | 100 | 5.1532e-51 ● | 100 | 3.1704e-67 ● |
| <i>ER_TOX</i> | 96 | 3.1118e-30 ● | 99 | 8.1943e-37 ● |
| <i>ER_LIT</i> | 69 | 2.3297e-06 ● | 66 | 5.3560e-06 ● |
| <i>ISS_m</i> | 100 | 1.4792e-50 ● | 100 | 8.8650e-76 ● |
| <i>ISS_r</i> | 100 | 1.5046e-46 ● | 100 | 3.5386e-44 ● |
| <i>COX2_4q</i> | 83 | 6.5791e-10 ● | 59 | 1.4442e-05 ● |
| <i>COX2_S.</i> | 56 | 0.0070 ● | 93 | 3.3759e-24 ● |
| Simple transfer vs. Best single distance | | | | |
| <i>DHFR_4q</i> | 68 | 1.9913e-06 ● | 40 | 0.1496 |
| <i>DHFR_S.</i> | 72 | 5.6524e-09 ● | 47 | 0.1856 |
| <i>CPDB_m</i> | 100 | 1.7442e-59 ● | 100 | 2.4231e-61 ● |
| <i>CPDB_r</i> | 100 | 3.0277e-56 ● | 100 | 6.3283e-69 ● |
| <i>ER_TOX</i> | 80 | 1.8104e-12 ● | 79 | 1.5431e-10 ● |
| <i>ER_LIT</i> | 92 | 1.7692e-22 ● | 80 | 1.7956e-06 ● |
| <i>ISS_m</i> | 99 | 1.4792e-50 ● | 100 | 4.7435e-26 ● |
| <i>ISS_r</i> | 97 | 1.5046e-46 ● | 55 | 1.3118e-22 ● |
| <i>COX2_4q</i> | 91 | 6.5791e-10 ● | 34 | 7.5261e-04 ○ |
| <i>COX2_S.</i> | 44 | 0.0070 ● | 78 | 0.0050 ● |
| Distance learning vs. Simple transfer | | | | |
| <i>DHFR_4q</i> | 14 | 2.9153e-17 ○ | 21 | 1.2097e-11 ○ |
| <i>DHFR_S.</i> | 44 | 0.8970 | 55 | 0.1030 |
| <i>CPDB_m</i> | 13 | 1.5582e-38 ○ | 69 | 8.3637e-04 ● |
| <i>CPDB_r</i> | 2 | 2.4788e-19 ○ | 26 | 1.1582e-04 ○ |
| <i>ER_TOX</i> | 73 | 5.3856e-09 ● | 84 | 1.0532e-12 ● |
| <i>ER_LIT</i> | 12 | 6.5350e-20 ○ | 44 | 0.0067 ○ |
| <i>ISS_m</i> | 29 | 8.8116e-07 ○ | 100 | 1.4392e-42 ● |
| <i>ISS_r</i> | 85 | 1.4217e-19 ● | 100 | 4.6961e-78 ● |
| <i>COX2_4q</i> | 15 | 6.0041e-13 ○ | 85 | 4.6009e-18 ● |
| <i>COX2_S.</i> | 52 | 0.0178 ● | 87 | 5.5511e-16 ● |

the simple transfer (see second section of Table 6.3), possibly depending on the quality of the source data and the representativeness of the few training examples. In summary, these results indicate that adapted transfer is a good compromise, which keeps the high predictive accuracy of distance adaptation on small and large training datasets, and improves on simple transfer in settings with large amounts of training data. This holds for both variants of the adapted transfer (bounded and penalized), which perform comparably with a slight advantage for the penalized version.

Table 6.3: Penalized adapted transfer vs. Distance learning. A vs. B: ●/○ = A significantly better/worse than B; w = “wins” of A.

| Penalized adapted transfer vs. Distance learning | | | | |
|---|-----|--------------|------|--------------|
| frac. trg. set | 10% | | 100% | |
| Short-hand | w | p-value | w | p-value |
| <i>DHFR_4q</i> | 75 | 3.6350e-11 ● | 84 | 1.7402e-18 ● |
| <i>DHFR_S.</i> | 61 | 0.0363 ● | 54 | 0.6914 |
| <i>CPDB_m</i> | 94 | 3.5594e-29 ● | 39 | 0.0160 |
| <i>CPDB_r</i> | 50 | 0.9865 | 69 | 4.7849e-04 ○ |
| <i>ER_TOX</i> | 49 | 0.0244 ● | 48 | 0.8390 |
| <i>ER_LIT</i> | 76 | 6.8167e-11 ● | 55 | 0.1088 |
| <i>ISS_m</i> | 69 | 1.3269e-05 ● | 58 | 0.1663 |
| <i>ISS_r</i> | 30 | 4.2098e-05 ○ | 81 | 2.6523e-13 ● |
| <i>COX2_4q</i> | 87 | 3.5753e-12 ● | 7 | 5.2729e-29 ○ |
| <i>COX2_S.</i> | 55 | 0.4762 | 22 | 7.1220e-11 ○ |
| Penalized adapted transfer vs. Simple transfer | | | | |
| <i>DHFR_4q</i> | 21 | 2.1467e-06 ○ | 54 | 0.1907 |
| <i>DHFR_S.</i> | 61 | 0.0014 ● | 54 | 0.0229 ● |
| <i>CPDB_m</i> | 17 | 9.5657e-13 ○ | 52 | 0.2353 |
| <i>CPDB_r</i> | 5 | 1.3033e-28 ○ | 42 | 0.3879 |
| <i>ER_TOX</i> | 82 | 8.5636e-12 ● | 81 | 3.5649e-13 ● |
| <i>ER_LIT</i> | 28 | 5.6761e-06 ○ | 44 | 0.3060 |
| <i>ISS_m</i> | 41 | 0.1454 | 100 | 1.1372e-41 ● |
| <i>ISS_r</i> | 81 | 3.9968e-15 ● | 100 | 1.3256e-79 ● |
| <i>COX2_4q</i> | 21 | 5.9618e-09 ○ | 60 | 0.0033 ● |
| <i>COX2_S.</i> | 75 | 3.4792e-08 ● | 64 | 7.0540e-05 ● |

6.3.3 Analysis of Optimized Weights

Figure 6.5 shows horizontally stacked bar-plots of the weights α_i optimized in the distance learning approach and of the weights α_i^p optimized in the penalized adaptation approach (mean over the hundred repetitions of ten-fold cross-validation). The weights α_1 based on the sub-structural features are shown in white, the pharmacophoric fingerprint based weights α_2 in gray and the MCS-based α_3 in black. A general observation is that the strength of the adaptation of the α_i s is consistent with the learning curves in Figures 6.2 and 6.4. Strong adaptation can, for example, be seen, e.g., in the *DHFR_4q* and *COX2_S.* datasets at 10% and at 100%. This effect can be clearly seen in the learning curve. Especially notable is that the MCS weights α_3 (black) are significantly lower for the DHFR and COX2 datasets. This reflects very nicely the fact that the compounds in those four datasets are much less diverse. Less diverse compounds can be distinguished more easily with local than with global differences as represented by the MCS-based weights α_3 .

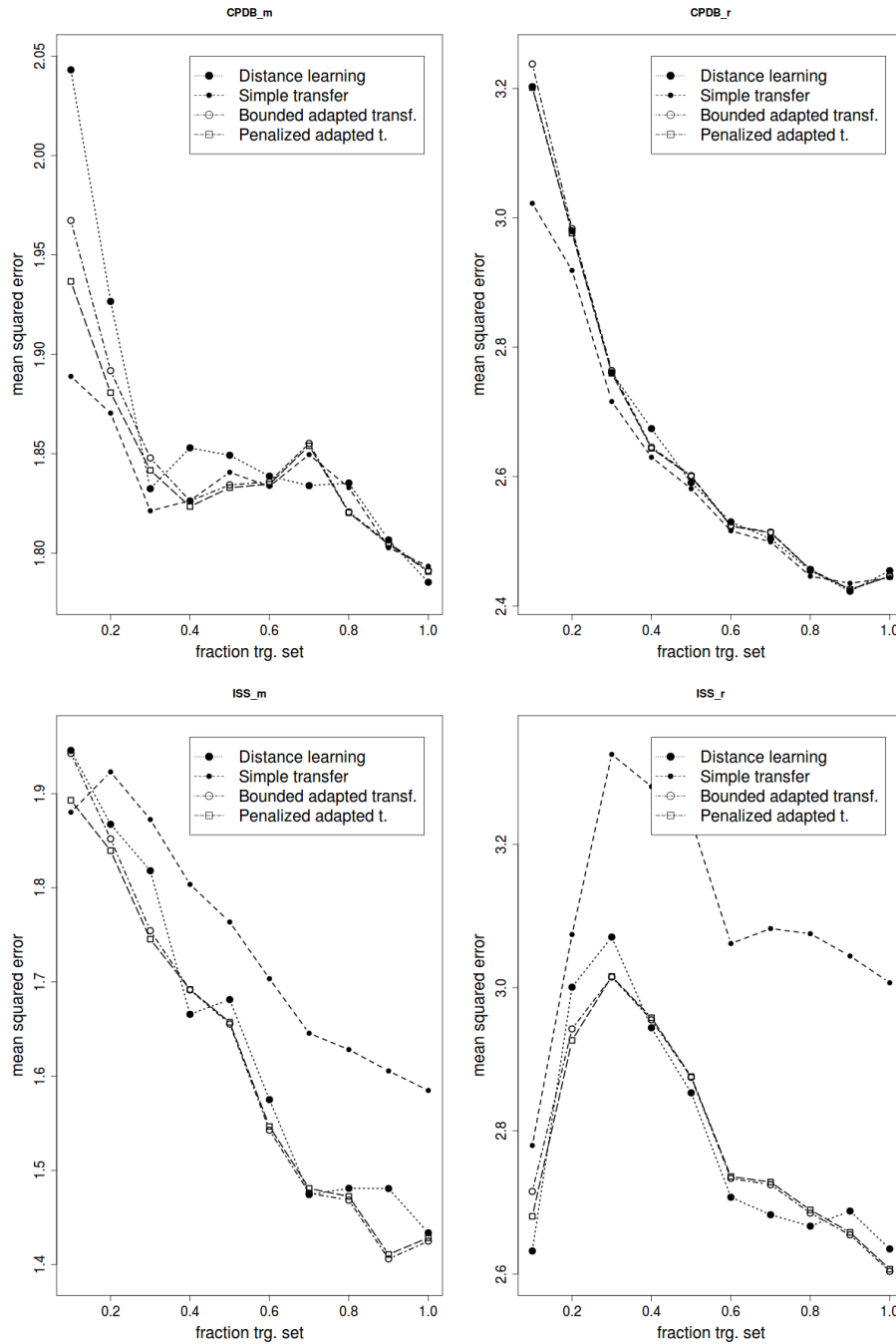


Figure 6.3: Learning curves for nearest neighbor with distance threshold $t = 0.2$.

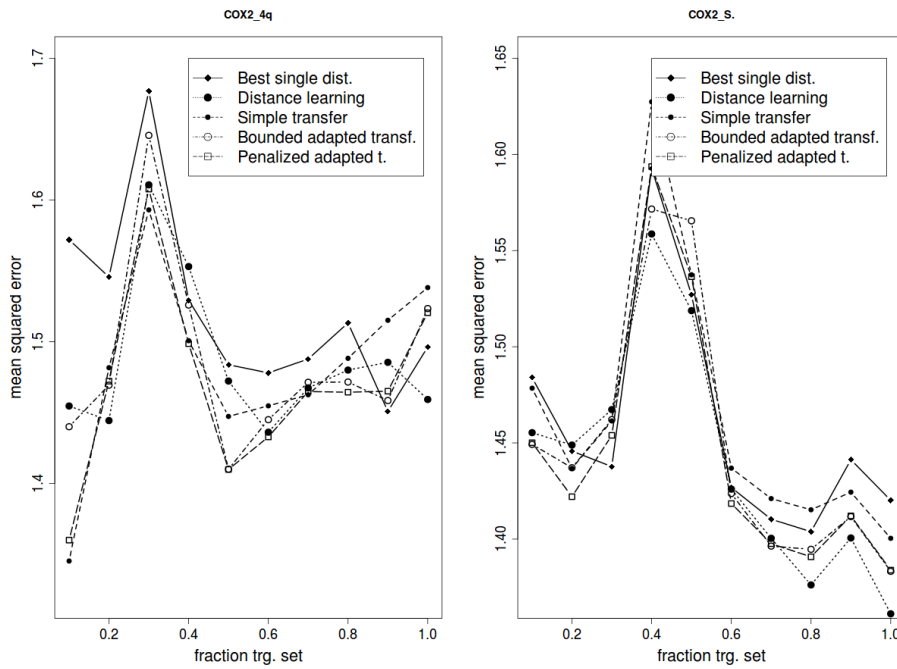


Figure 6.4: Learning curves for nearest neighbor with distance threshold $t = 0.2$.

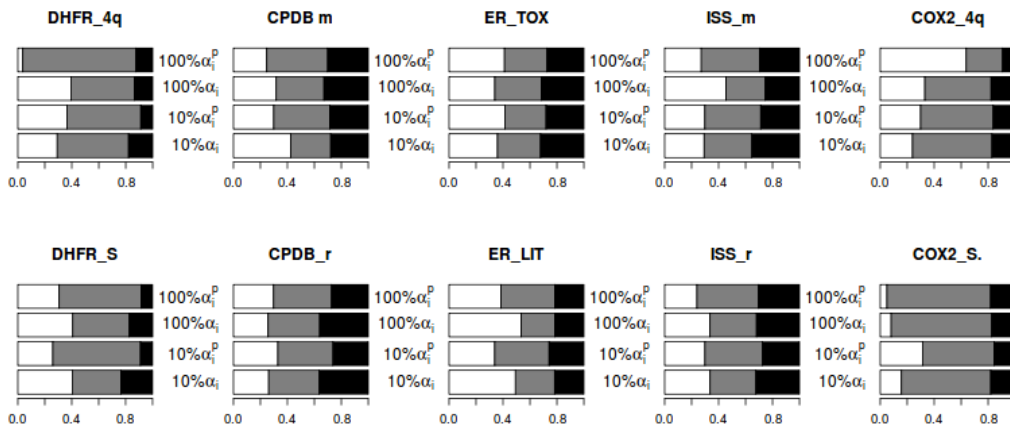


Figure 6.5: Graphical representation of the α_i and α_i^p at 10% and 100% of the training data. α_1 (cFTs) = white, α_2 (joelib) = gray and α_3 (MCS) = black.

6.4 Data-driven Selection of Source Datasets

In Sections 6.2 and 6.3 of this study we worked with the assumption that we have two datasets at hand that are related and that we can use one of them to learn the bias for the other (and vice versa). In this section we discard this assumption and present an approach to select the source dataset in a data-driven way, given a pool of datasets that contains a potentially related one. As pool of available datasets we use the PubChem BioAssay database [147], which contains more than 500k datasets (bio assays)²⁷ from toxicological and biochemical experiments.

6.4.1 Source Dataset Selection

Of the datasets used in the first part of this work, only the CPDB mouse (PubChem BioAssay identifier: 1199) and CPDB rat (AID: 1208) datasets were deposited in the PubChem BioAssay database by the respective data providers. The PubChem datasets are eight (mouse) and four compounds (rat) smaller, due to the validation and standardization process that is performed when a depositor electronically submits a dataset to PubChem. The remaining datasets used so far are only partially or incompletely available in the database. To get a reasonable number of target datasets we added three more datasets (bio assays) that are of importance in drug research and development. The first one results from a primary biochemical high throughput screening (HTS) assay for agonists of the steroid receptor coactivator 1 (SRC-1) recruitment by the peroxisome proliferator-activated receptor gamma PPAR γ (AID: 631). The second one, ER_p, was generated from a HTS of estrogen receptor- α coactivator binding potentiators (AID: 639). The third additional target dataset is an antagonist confirmation screen of the M1 muscarinic receptor (AID: 677), abbreviated M1_c. In the PubChem BioAssay database, compounds in an assay are categorized either as *active* or *inactive* with respect to the biological assay at hand (some can also be categorized as *inconclusive*, but this can be due to technical experimental problems). The categorization is usually done with a threshold on the real-

Table 6.4: Datasets used for data-driven selection of source datasets. *sim* represents the PubChem activity similarity.

| Target dataset | | | Source dataset | | | <i>sim</i> |
|----------------|------|------|----------------|------|------|------------|
| Name | AID | Size | Name | AID | Size | |
| mouse | 1199 | 436 | rat | 1208 | 576 | 0.311 |
| rat | 1208 | 576 | mouse | 1199 | 436 | 0.323 |
| SCR-1 | 631 | 811 | SCR-2 | 1297 | 410 | 0.412 |
| ER_p | 639 | 1151 | ER_i | 629 | 1442 | 0.046 |
| M1_c | 677 | 723 | M1_p | 628 | 2179 | 0.332 |

²⁷ accessed Sept. 26, 2011

valued target variable (experimental endpoint/measurement). For all five target datasets we used the *activity overlap* measure (OV_{act}) to retrieve a similarity ranking of related assays available in the database. Hereby, the similarity of two sets A and B is calculated with a Tanimoto-like similarity coefficient, using the categorization of the compounds:

$$OV_{act}(A,B) = \frac{act(A \wedge B)}{act(A) + act(B) - act(A \wedge B)}, \quad (6.4)$$

where $act(\cdot)$ returns the number of active compounds in an assay. From this ranking we use the first assay of sufficient size to be used as source dataset (size > 100) that is not a superset of the considered target dataset and that has a meaningful relation to the considered target dataset. Unfortunately, the last constraint is important for a meaningful and valid transfer of learning bias, but can not be ensured automatically. However, this might be a problem specific to the domain the data stems from and is due to the lack of meta-data, e.g. ontology data, relating the different toxicological and biochemical target variables. This selection process yields for the CPDB mouse and CPDB rat datasets the respective other one as source dataset, as previously also done via hand-selection. For SCR-1, a HTS assay for agonists of the steroid receptor coactivator 2 (SRC-2) recruitment by the PPAR γ (AID: 1297) is found, for ER_p a HTS of Estrogen Receptor- α coactivator binding inhibitors (ER_i; AID: 629). Note that assays 639 and 629 have inverse meaning of the target variable (ER potentiators and ER inhibitors). We kept this pair of datasets to investigate the outcome of such an “inverse” setting and if the manual part of the source dataset selection could be reduced or automated completely (in case the transfer works well regardless of the inverted meaning of the target variable). For M1_c an antagonist primary screen of the M1 muscarinic receptor (M1_p; AID: 628) was found. An overview of the datasets and the similarities is given in Table 6.4. The distance matrices for those ten target and source datasets are calculated as described in Section 6.2.1. We performed the same experiments as described in Section 6.3.

6.4.2 Discussion and Results

To evaluate how well our presented semi-automatic source data selection works with the adapted transfer strategy, we use the same step-wise comparison of approaches as done in Section 6.3.2. Table 6.5 shows the associated results. We also provide learning curves in Figure 6.6 and Figure 6.7. Our first experiment is again the comparison of the predictive accuracy of the distance learning setting with the best single distance. The results show that learning the distance combination outperforms the single best distance in 5 out of 5 cases for 10% and for 100% significantly. This strengthens the findings in Section 6.3.2. However, no statement on the source data selection can be made at this point, as no transfer is involved. In the second experiment we investigate if inductive transfer from a source dataset selected with our data-driven approach can improve the predictive performance compared to the best single distance. Considering the results in section two of Table 6.5

we can say that the simple inductive transfer has the same effect as distance learning. This effect is apparent for hand-selected and semi-automatically selected source datasets. For this discussed comparison, the “inverse” source dataset for assay 639 seems to have no influence. We attribute this tendency to the weak performance of the best single distance.

After finding that the simple transfer works very well with respect to the best single distance, we now assess if the circumstances under which one approach outperforms the other are the same as those apparent with the hand-selected source datasets. The respective results are shown in the third section of Table 6.5. Inductive transfer is significantly better on three of the five datasets (three out of four if we assume the “inverse” dataset is not suited well as source dataset), when there is few training data available. This effect is lost when more training data (100%) is used. However, for the five datasets at hand, the distance learning method outperforms the simple transfer only in 1 of 5 cases at 100% of the training data (compared to 5/10 in the hand-selected case).

We now compare the penalized adapted transfer method with its building blocks. The fourth and fifth sections of Table 6.5 show the relevant results. We discuss the performance on assay 639 separately, because of the “inverse” source dataset. As in Section 6.3, we see that adapted transfer can give an improvement (two of four cases) if there is only few training data available, but no further improvement can be made if there are sufficient training data at hand. The mixed strategy performs slightly worse than the simple transfer at 10% but can improve the results if given more training data (100%). For assay 639 it seems irrelevant, if we perform any transfer, as there is no improvement at any point. This shows that the selection of a meaningful source dataset is very important, and at least for the problem domain at hand a well-defined ontology linking different biochemical assays is needed to facilitate an automated selection process. Comparing the results compiled for the two sets of CPDB mouse and CPDB rat data, the results are consistent although some compounds were removed. Summing up, we can say that the tendency of the results of the data-driven selection is the same as for the hand-selected. This fact means that our approach is a successful example of how source datasets for inductive or adaptive transfer can be selected more automatically.

6.4.3 Comparison with Boosting for Regression Transfer

For a comparison with an existing method for transfer learning that uses transfer in a regression setting, we chose *Boosting for Regression Transfer (TrAdaBoost.R2)* [98], which is a regression variant of the well-established *TrAdaBoost* method [25] that was developed for classification settings. A conceptual difference of the adapted transfer methodology presented in this work and *TrAdaBoost.R2* is that the latter uses the source dataset directly in combination with the target training set. In each boosting iteration, the relative target instance weights are increased if the instance is misclassified. The source instance weights are decreased in such a case. This process identifies the source data instances that are most informative for learning with the target dataset. Another difference is

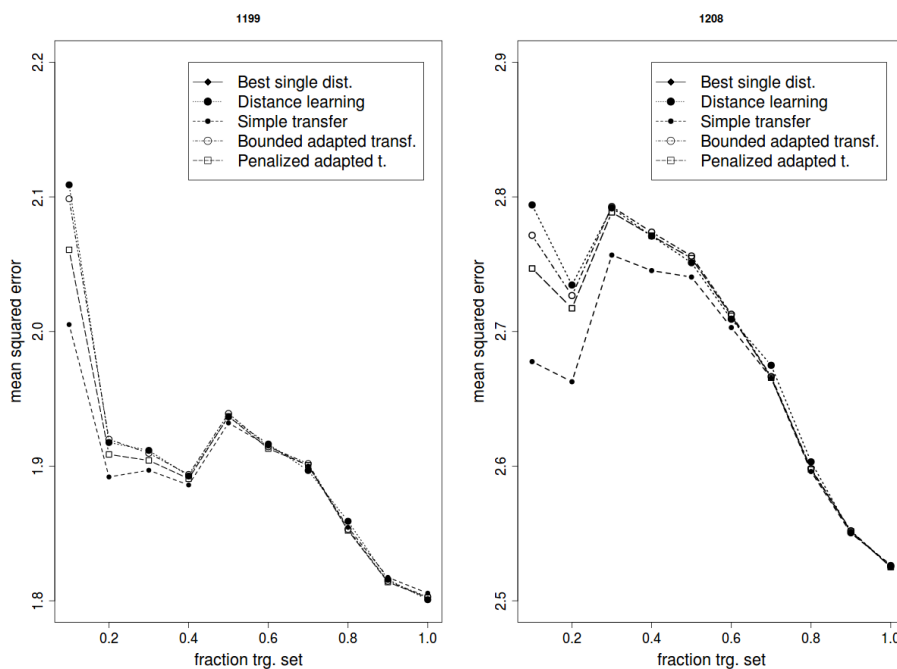


Figure 6.6: Learning curves for the data-driven selection of source datasets for AID 1199 and AID 1208 and nearest neighbor with dist. thr. $t = 0.2$.

that we use an optimized combination of distance contributions as input for a k -NN learner whereas TrAdaBoost.R2 uses the input features directly. An advantage of using the distance contributions as input for the learning problem is that we achieve additional interpretability as we obtain information on the importance of the different representations of the input instances (see Figure 6.5).

In our comparison experiments, we used the fingerprints generated for distance calculation directly as features. The closed FTM binary occurrence features were combined with the JOELIB2 pharmacophoric fingerprints as input features for the TrAdaBoost.R2 method. The third source of information (MCS) was not included, as here the distances were calculated directly from the input molecules (instances). The TrAdaBoost.R2 algorithm was used as provided by the authors. As base learner for TrAdaBoost.R2, we employed the M5P model tree algorithm from the WEKA workbench [49]. As suggested by the authors, we conducted 10 iterations of boosting and also left the other parameters at the provided defaults. We performed ten-fold cross-validation and report the root mean squared error results in Table 6.6. We did not perform these experiments for the “inverse” source data target data combination (629 and 639).

The results show that adapted transfer outperforms TrAdaBoost.R2 in only one of four cases. However, we have to keep in mind that the model-based TrAdaBoost.R2 base learner M5P is more powerful than the lazy k -NN learner used in conjunction with the Adapted Transfer algorithm, especially for smaller datasets. Consequently, the comparison of the transfer methods is per se slightly uneven. Another factor that attenuates the difference of the results is the higher interpretability of the Adapted Transfer results. The contributions

Table 6.5: Distance learning vs. Best single distance. A vs. B: ● = A significantly better than B, ○ = A significantly worse than B; w = how often out of 100 times A “wins” against B.

| Distance learning vs. Best single distance | | | | |
|---|-----|--------------|------|--------------|
| frac. trg. set | 10% | | 100% | |
| Short-hand | w | p-value | w | p-value |
| 1199 | 100 | 3.8966e-18 ● | 100 | 3.8963e-18 ● |
| 1208 | 99 | 4.0162e-18 ● | 100 | 3.8966e-18 ● |
| 631 | 90 | 2.3540e-14 ● | 83 | 1.1879e-14 ● |
| 639 | 99 | 4.0162e-18 ● | 98 | 5.2674e-18 ● |
| 677 | 100 | 3.8966e-18 ● | 100 | 3.8961e-18 ● |
| Simple transfer vs. Best single distance | | | | |
| 1199 | 100 | 3.8966e-18 ● | 100 | 3.8966e-18 ● |
| 1208 | 99 | 4.0162e-18 ● | 100 | 3.8966e-18 ● |
| 631 | 97 | 5.5962e-18 ● | 86 | 8.8028e-14 ● |
| 639 | 100 | 3.8966e-18 ● | 99 | 4.0159e-18 ● |
| 677 | 100 | 3.8966e-18 ● | 100 | 3.8966e-18 ● |
| Distance learning vs. Simple transfer | | | | |
| 1199 | 8 | 2.3391e-15 ○ | 52 | 0.4331 |
| 1208 | 13 | 1.9266e-14 ○ | 40 | 0.5191 |
| 631 | 38 | 6.5073e-05 ○ | 57 | 0.4509 |
| 639 | 54 | 0.1896 | 41 | 0.1680 |
| 677 | 99 | 4.3971e-18 ● | 77 | 2.0655e-11 ● |
| Penalized adapted transfer vs. Distance learning | | | | |
| 1199 | 78 | 1.5961e-08 ● | 49 | 0.4556 |
| 1208 | 69 | 2.1072e-05 ● | 49 | 0.7181 |
| 631 | 46 | 0.6214 | 53 | 0.4798 |
| 639 | 44 | 0.0755 | 44 | 0.1690 |
| 677 | 52 | 0.8151 | 41 | 0.1299 |
| Penalized adapted transfer vs. Simple transfer | | | | |
| 1199 | 18 | 3.1127e-11 ○ | 50 | 0.9712 |
| 1208 | 21 | 1.7386e-09 ○ | 47 | 0.9178 |
| 631 | 32 | 5.8773e-05 ○ | 54 | 2.145e-02 ● |
| 639 | 53 | 0.9233 | 38 | 0.0031 ○ |
| 677 | 99 | 4.3969e-18 ● | 76 | 8.0233e-11 ● |

Table 6.6: Result comparison with TrAdaBoost.R2 for three datasets. Shown are RMS error values for 10% and 100% of the target training dataset.

| | TrAdaBoost.R2 | | Penalized Adaptation | |
|------|---------------|--------|----------------------|--------|
| | 10% | 100% | 10% | 100% |
| 1199 | 1.139 | 1.003 | 1.436 | 1.343 |
| 1208 | 1.477 | 1.185 | 1.657 | 1.589 |
| 631 | 26.293 | 27.264 | 36.067 | 34.207 |
| 677 | 137.684 | 92.251 | 80.231 | 74.228 |

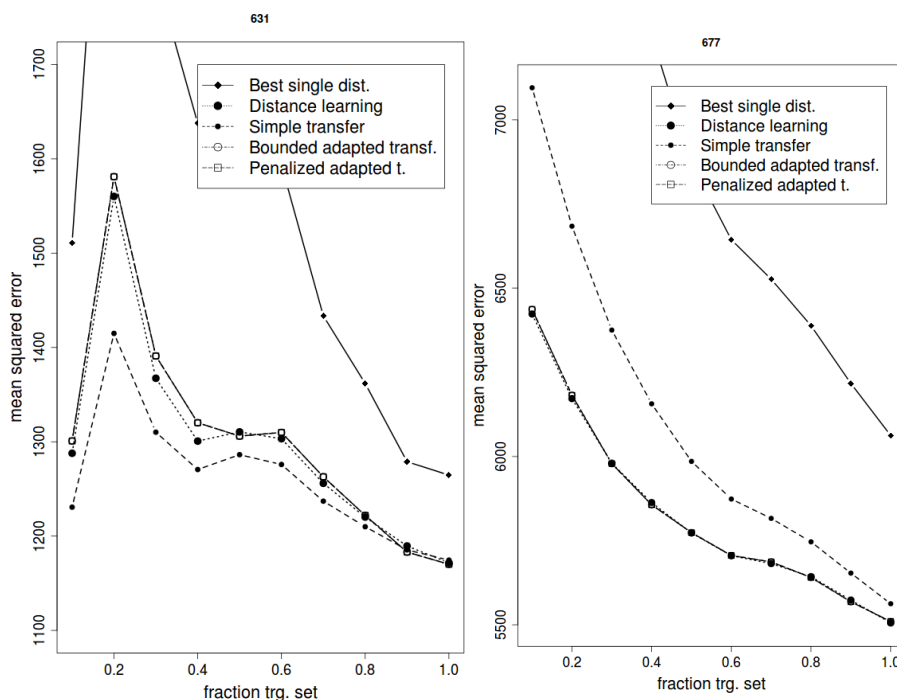


Figure 6.7: Learning curves for the data-driven selection of source datasets for AID 631 and AID 677 and nearest neighbor with dist. thr. $t = 0.2$.

of the used distance measures are especially useful in the domain of Quantitative Structure Activity Relationships (QSARs) and cheminformatics, for which the Adapted Transfer approach was developed in the first instance. The contributions also make it possible to gain insight into the diversity of a chemical dataset as the diversity influences the weights of the three chosen distance measures.

6.5 Conclusion

In this chapter, we proposed adapted transfer, a method combining inductive transfer and distance learning, and evaluated its use for quantitative structure-activity relationships. The method derives linear combinations of contributions of distance measures for chemical structures. Compared to inductive transfer and distance learning alone, the method appears to be a good compromise that works well both with large and small amounts of training data. Technically, the method is based on convex optimization and combines the contributions from representatives of two distinct families of distance measures for chemical structures, MCS-based and fingerprint-based measures. In a last step, we got rid of the assumption that we have a source task by default. We presented an approach for a data-driven selection of the source task from a database of datasets, using a similarity based on a categorization of the endpoint.

In further work, it would be interesting to see if using multiple source datasets instead of one could offer added value when the different source datasets contain complementary information relevant for the target dataset. Second, it would be intriguing to embed the

transfer and adaptation of distance measures into a Bayesian framework. Third, the quantification of the relatedness of source and target datasets could further be improved. For instance, one could think of a scenario where the strength of the adaptation is dependent on the distance between the target and the source datasets.

CHAPTER 7

Relations Between the Presented Approaches

In the preceding three chapters we presented approaches that work with the concept of small molecule similarity to enhance applications in cheminformatics and QSAR modeling. Chapter 4 introduced similarity boosted QSAR, Chapter 5 presented a concept of structural similarity measures extended with background knowledge and Chapter 6 introduced Adapted Transfer of Distance Measures. In the following sections we discuss the pairwise relations of the three approaches to stress where synergy effects through combinations of the approaches can be achieved. We also think about possible options to plug one approach directly into the other and discuss why certain combinations do not make sense or are not possible due to technical or conceptual differences. Figure 7.1 gives an overview of the binary relations of the approaches.

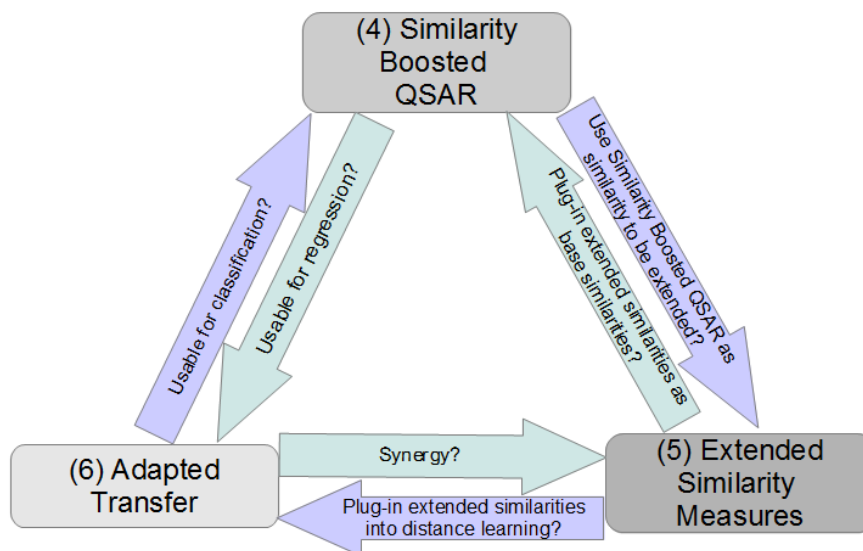


Figure 7.1: Possibilities discussed for plug-in or synergy scenarios of the three presented approaches. The respective chapter number is given in brackets.

7.1 Similarity Boosted QSAR and Improved Structural Similarities

The main contribution of the work presented in Chapter 4 are molecular descriptors constructed from similarities with respect to reference compounds. The similarity is a consensus similarity of five base similarities. The reference compounds can be selected either by literature review (\mathcal{R}_{LIT}), by clustering the active structures and using cluster representatives as reference compounds (\mathcal{R}_{ACT}) or by clustering a set representative of the chemical space to get reference structures (\mathcal{R}_{DB}). Chapter 5 presents two possibilities to significantly improve structural similarity measures by an extension that incorporates background knowledge. The structural similarity measures used are a maximum common subgraph (MCS) based similarity and an ECFP circular fingerprint based similarity. The extensions are denoted MCS_{ext} and $ECFP_{ext}$. Plainly speaking two similarity measures are presented.

Keeping those facts in mind, it seems obvious that the first way to use both approaches in combination is to plug the extended similarity measures into the similarity boosted QSAR method as further base similarities constituting the similarity with respect to the reference compounds. This plug-in scenario only makes sense if the additional similarities encode further information or even information complementary to that already encoded by the five base similarity measures in use: MACCS keys similarity, topological fingerprints similarity, ECFP and FCFP circular fingerprints similarity and atom pairs fingerprints similarity. Consequently, only an addition of the MCS_{ext} similarity is an attractive option. The $ECFP_{ext}$ would have to be used instead of the plain ECFP circular fingerprint similarity. However, there remains the issue of a meaningful extension in the given problem setting. In Chapter 5 we use a small number of known high-quality ligands to automatically generate features for the extension fingerprint or literature information on a single binding-relevant substructure. Basically, the used dataset is unlabeled. In the QSAR setting of Chapter 4 we have a completely labeled dataset with 50% active and 50% inactive compounds (in our experiments). Those active compounds are of mixed quality with respect to their binding affinity, as a structure labeled “active” can be a strong a medium or a weak binder²⁸. Thinking along the lines of the similarity boosted QSAR approach one could consider using only \mathcal{R}_{LIT} or the \mathcal{R}_{ACT} reference compounds instead of all active compounds to generate the extension fingerprint. Nevertheless, at least in the clustering variant the quality issue remains. If we were confronted with a regression setting, one could easily choose the five, ten or k “best” instances looking at the target values. The \mathcal{R}_{DB} reference compounds contain no binding-relevant information at all and thus can not be used. If we use the \mathcal{R}_{LIT} reference compounds to generate the fingerprint extension, this is first of all not satisfying from a data mining perspective, as in principle we would like to use the computational methods to reduce – or in an optimal case completely avoid – having to use

²⁸ In a regression setting one could consider to use the best binders but in the given classification setting the distinction between strong and less strong binders is impossible.

manual feature extraction. Second, we would have to use the \mathcal{R}_{ACT} or the \mathcal{R}_{DB} variant of the similarity vector composition to avoid encoding redundant information by using the \mathcal{R}_{LIT} reference compounds twice in the feature generation process. Furthermore, potential overlaps of the \mathcal{R}_{LIT} reference compounds with the \mathcal{R}_{ACT} or the \mathcal{R}_{DB} reference structures would have to be removed. Concluding we can state that we expect slight improvement in the base similarity measure, but the effect will be leveled out to a great extent in the combined approaches that give the best performances in our experiments.

Looking at the constellation the other way around, we note that the molecular descriptors derived in the similarity boosted QSAR approach constitute no direct structural representation of the molecules which was a motivation to use ECFP and MCS based similarities in Chapter 5. Nevertheless, we use the similarity boosted QSAR descriptors in an experiment to test whether they can be useful not only in a predictive QSAR setting but also in virtual screening. The experiment shares the setup with the ranking experiments in Chapter 5 and a more detailed description of the setup can be found there. In addition to the similarities sim_{MCS} (see (2.5)), sim_{ECFP} and their respective extended variants we calculate rankings based on the \mathcal{R}_{ACT} descriptors. As the values in the \mathcal{R}_{ACT} descriptor vectors are numerical in the interval (0.0,1.0) it suggests itself to apply the standard Euclidean distance to compare two instances and invert the distance ranking afterwards. The DuD datasets [62] used in Chapter 5 are basically unlabeled. The categorization as ligand and decoy could be used, but to get a meaningful clustering as basis for the \mathcal{R}_{ACT} descriptors the number of ligands is too small (23 – 349 ligands; see Table 5.3). Consequently, we use datasets from Chapter 4 in the experiments. Of the datasets used in Chapter 4 (see left hand side of Table 7.3) only the ones with AID 631 and 639 have a categorization in active and inactive compounds. The remaining datasets only provide regression data. We need the regression data to be able to select a subset of the 50% of the datasets that are labelled active to get a virtual screening setting. Compounds labelled active may be strong or weak inhibitors. We are only interested in strong inhibitors that can be used in analogy to the DuD ligand compounds. Therefore, we select the ten compounds with the best values in the regression endpoint. The compounds are referred to as “ligand compounds” and are listed in Table 7.1. One of those compounds is selected randomly and used as query compound in the virtual screening experiments.

The results of the six similarity rankings are shown in Table 7.2. As evaluation measures we use the Δ_{EF} (1.3) measure and the mean ranks (μ_{Rank}) of the ligands in the ranking.

First, we analyze the performance of the SimBoosted QSAR derived ranking, then we analyze the extended variants compared to the non-extended variants before we compare the different ranking methods amongst each other. Considering the Δ_{EF} values at 1%, 5% and 10% of the database, \mathcal{R}_{ACT} performs worse for dataset 631 and competitive for dataset 639. Looking at the μ_{Rank} values, however, the performance is worse than that of both structural similarity measures.

In all three cases (MCS, ECFP and \mathcal{R}_{ACT}) the extended variant either gives the same performance as the base variant or improves it. The Δ_{EF} values at 1%, 5% and 10% of

| index | AID 631 | AID 639 |
|-------|----------------|----------------|
| 1 | 2446271 | 1252572 |
| 2 | 5702697 | 2012958 |
| 3 | 6099 | 650090 |
| 4 | 3096032 | 664182 |
| 5 | 1103487 | 1119214 |
| 6 | 4050208 | 2561590 |
| 7 | 12871037 | 5708073 |
| 8 | 1779937 | 2871821 |
| 9 | 2917291 | 1283398 |
| 10 | 664285 | 740743 |

Table 7.1: Top 10 inhibitors in assay 631 and 639. Given are the PubChem compound identifiers (CID). The randomly chosen query compound is marked with bold print.

the database are improved in 5 of 6 cases for MCS_{ext} , in only 1 of 6 cases for $ECFP_{ext}$ and in all cases for \mathcal{R}_{ACT} . The μ_{Rank} values are improved in all cases. From those numbers we conclude that extending the similarity measures is beneficial in all three cases, however, the \mathcal{R}_{ACT} descriptor based ranking benefits the most from the extension.

Comparing the Δ_{EF} values of the three extended similarity measures MCS_{ext} and $\mathcal{R}_{ACT_{ext}}$ perform equally and have a slight advantage in comparison to $ECFP_{ext}$. Looking at the μ_{Rank} values MCS_{ext} is better than $ECFP_{ext}$ which is better than $\mathcal{R}_{ACT_{ext}}$. However, in practical applications the enrichment factor values will be of more importance than the mean ranks.

Concluding, we can sum up that on the one hand the similarity boosted QSAR descriptors with \mathcal{R}_{ACT} reference compounds without extension add no accuracy to the similarity ranking procedure. The extended version on the other hand can be utilized not only in predictive QSAR applications but also in virtual screening scenarios and give competitive results to existing structural similarity measures.

| method | 631 | | | | 639 | | | |
|---------------------------|---------------------|---------------------|--------------------|--------------------|---------------------|--------------------|--------------------|--------------------|
| | 1% | 5% | 10% | μ_{Rank} | 1% | 5% | 10% | μ_{Rank} |
| MCS | 70.962 | 18.247 | 8.010 | 373.6 | 94.173 | 18.174 | 8.007 | 414.1 |
| MCS_{ext} | 70.962 | 16.220 [•] | 7.009 [•] | 245.9 [•] | 83.709 [•] | 8.007 [•] | 4.003 [•] | 254.7 [•] |
| ECFP | 70.962 | 16.220 | 8.010 | 322.6 | 94.173 | 14.135 | 7.006 | 400.7 |
| $ECFP_{ext}$ | 70.962 | 16.220 | 8.010 | 254.4 [•] | 94.173 | 14.135 | 6.005 [•] | 284.9 [•] |
| \mathcal{R}_{ACT} | 91.237 | 18.022 | 8.010 | 564.4 | 90.078 | 14.012 | 7.006 | 697.4 |
| $\mathcal{R}_{ACT_{ext}}$ | 70.962 [•] | 16.220 [•] | 7.009 [•] | 279.8 [•] | 83.709 [•] | 8.007 [•] | 4.003 [•] | 353.7 [•] |

Table 7.2: Δ_{EF} values at 1%, 5% and 10% of the database. μ gives the mean rank of the top 10 ligands, respectively. Improvements of the extended compared to the non-extended variant are marked with a [•].

7.2 Adapted Transfer of Distance Measures and Improved Structural Similarities

When we consider the second binary relation of the presented approaches, it is possible to plug the extended similarity measures from Chapter 5 into the Adapted Transfer of Distance Measures approach (Chapter 6). More precisely, distance variants of the extended similarities MCS_{ext} and $ECFP_{ext}$ could be used in the distance learning part of the adapted transfer approach. As an MCS based distance measure is already used in the original Adapted Transfer approach, MCS_{ext} can be used instead of MCS. However, as the combination of the contributions of the three distance measures using MCS, JOELib2 and FTM feature information is optimized, the effect on the predictive performance will probably be rather small, as the extension is realized via FTM features. $ECFP_{ext}$ could be used as additional fourth distance measure.

As discussed for the combination of the extended similarity with the similarity boosted QSAR approach in Section 7.1, the critical part of the usage of the extended similarity as plug-in is to find a meaningful way to generate the similarity extension. In the adapted transfer approach we experiment with regression datasets, and in consequence the compounds with the highest activity could be used as input for the similarity extension method. A problem with the extended similarity as plug-in for the Adapted Transfer of Distance Measures approach is that we already use the contribution of FTM calculated structural features in the distance learning procedure. We suspect that the benefit of the extension encoding background knowledge that has been shown for the virtual screening scenario is mitigated by the fact that we already encode a lot of structural information by the closed FTM features. An addition of the ECFP distance (without extension) might make sense as the ECFP distance also encodes information about the atomic invariants that are not encoded by other structural features. This theoretical improvement, however, is not attributed to the combination of the presented approaches.

Another argument against the usage of the extended similarity measures as plug-in for the adapted transfer approach is that the incorporation of data-dependent features will increase the CPU runtime of the approach. Distance matrices based on features that are not set-dependent can be calculated before the validation process and outside of the cross-validation. They have to be calculated only once per dataset. If the distance matrix has to be calculated for each training fold of the repeated cross-validation procedure it has to be calculated ten thousand times (100×10 -fold cross-validation and 10 increments of training set size from 10% to 100%), or even fifty thousand times in case of the *Best single distance* experimental setting (additional 5-fold internal cross-validation)

As for Section 7.1, we conclude that a utilization of the Adapted Transfer of Distance Measures procedure in the similarity extension approach has no practical use.

7.3 Similarity Boosted QSAR and Adapted Transfer of Distance Measures

The approaches similarity boosted QSAR and Adapted Transfer of Distance Measures have in common that they basically solve the same problem: QSAR modeling. The difference is that the first of the two approaches solves classification problems, the latter solves regression problems. Another difference is that the adapted transfer approach is especially designed for QSAR problems where only a limited amount of training data is available and thus the performance can be improved by transferring and adapting a bias from another, related problem. In consequence to those differences, we did not select a common set of datasets for the experimental evaluations of the approaches. However, there is an overlap between the datasets used to evaluate the two approaches. Of the seven PubChem BioAssay [147] datasets used in the similarity boosted QSAR approach, three are also used in the second, data-driven source selection part of the adapted transfer approach as they provide also a real-valued endpoint values and are of relatively limited size: the estrogen receptor assay (ER; AID 639), the steroid receptor coactivator 1 assay (SRC-1, PPAR γ ; 631) and the M1 muscarinic receptor assay (M1; 677). An overview of the datasets used in the similarity boosted QSAR approach and the adapted transfer approach is given in Table 7.3 (only the datasets from the data-driven selection experiments of the adapted transfer approach are shown, as there is no overlap with the ten hand-selected datasets).

| similarity boosted QSAR | | | Adapted Transfer | | |
|-------------------------|----------|-------|------------------|-------|------|
| AID | Endpoint | n | AID | Name | m |
| 639 | ER | 2302 | 639 | ER_p | 1151 |
| 631 | SCR-1 | 1622 | 631 | SCR-1 | 811 |
| 677 | M1 | 1446 | 677 | M1_c | 723 |
| 1511 | hERG | 3104 | 1199 | mouse | 436 |
| 2796 | AhR | 15980 | 1208 | rat | 576 |
| 1479 | THR | 1632 | - | - | - |
| 2156 | KCNQ2 | 6814 | - | - | - |

Table 7.3: Summary of the used similarity boosted QSAR and Adapted Transfer datasets (data-driven source selection only). The number of examples n for the similarity boosted QSAR approach consists of 50% active and 50% inactive structures. The number of examples m for the Adapted Transfer approach only consists of active structures with a real-valued endpoint variable. AID corresponds to the PubChem BioAssay identifier.

Another question to be discussed is if the adapted transfer approach could also be used to solve classification problems. A direct usage in classification is not possible, as the approach is based on the optimization criteria (6.1), (6.2) and (6.3) that are optimized with respect to the squared error on the training set. The optimization criteria would have to be adapted and the quality measure with respect to which the weights are optimized would have to be replaced by a classification quality measure. Overall, this would heavily alter the approach. Alternatively, the classification problem could be interpreted as regression

problem with target values being only 0.0 and 1.0 and thus use the adapted transfer approach to solve it. However, this seems to be an artificial construct and we suspect that it will result in poor predictive performance. In contrast, the use of the similarity boosted QSAR methodology to learn regression models is a straight forward experiment. Basically the only thing that changes is that the target variable is real-valued instead of nominal and thus the quality measures used for evaluation change.

If we think about a plug-in scenarios with the two discussed approaches, the first thing that comes to mind is using distance learning to optimize the contributions of base similarities of the similarity boosted QSAR approach. We refrain from experiments assessing if this combined approach has additional predictive power at this point, as it is not a combination of our contributions but rather a conceptional addition to the similarity boosted QSAR approach.

CHAPTER 8

Application: Distributed REST Web Services for Toxicity Prediction

With the growing significance of web services as tools and interfaces in the scientific community, it is standing to reason to offer services particularly tailored for specific problem domains like the one at hand: predictive toxicology. Web services introduce and offer not only a lot of flexibility to software, but also enable users and developers to easily contribute to them. Another driving force of today's web-based technologies is the open source concept. Open source software gains its quality, flexibility and diversity from community efforts. It has been discussed in the literature and on the web for some years now, if the open source concept - not limited to software - can help innovation in drug discovery²⁹ [27, 126, 68, 90, 97, 43]. B. Munos at Eli Lilly & Co. asks, if open source R&D can reinvigorate drug research, as the low number of novel therapeutics approved by the US Food & Drug Administration (FDA) in recent years continues to cause great concern [88]. In his words, the resulting model is a hybrid in which a part of the R&D process is open-sourced while the rest is outsourced. To function, however, it needs strong project leadership and expertise in the minutia of drug R&D, which mostly exist in big pharmaceutical firms. This suggests that, far from being a threat to conventional drug R&D, open-source could be a way to leverage big pharma's capabilities in order to tackle challenges that the blockbuster model cannot address economically, such as neglected diseases. Although the first inroads to open source software in biological and chemical research were made in bioinformatics [27, 43], there exists a huge number of valuable open source software suites and tools for cheminformatics and predictive toxicology applications today. Examples are the chemistry development kit (CDK) [128], RDKit³⁰, AZOrange³¹ [127], openbabel [94], gSpan' [67, 161], FTM [113], BBRC [87] or LAST-PM [86] to name just a few. This versatile set of tools allows to conduct nearly all operations necessary in daily predictive toxicology, especially when combined with open source data mining and machine learning software packages, like the WEKA workbench [49]. However, the different tools use different data formats, are available in different programming languages

²⁹ Open source drug discovery: http://p2pfoundation.net/Open_Source_Drug_Discovery.

³⁰ <http://www.rdkit.org>

³¹ <https://github.com/AZCompTox/AZOrange>

for different platforms and consequently make a lot of conversion work, script writing and patching up software necessary.

This chapter focuses on OpenTox [52, 45], a project I and my colleagues have worked on for two and a half years as developers and researchers. OpenTox is a distributed, REST-based web service framework for predictive toxicology. It has been developed in an EC FP7 project³², with the goal of promotion, development, validation, acceptance and implementation of QSAR for toxicology. Building an integration framework for predictive toxicology makes it easier to compare multiple models, merge data from different sources or find all models available for a certain endpoint. Such a framework also addresses every day computational toxicology challenges like multiple data formats, implicit semantics that are often buried in human readable documentation, multiple software solutions that are mostly incompatible and hard to achieve prediction reproducibility. The framework has been designed according to the OECD validation principles³³, REACH regulatory guidance requirements for *in silico* models³⁴ and user requirements.

The focus of OpenTox is to provide an open and extensible framework rather than a closed, rigid software bundle. It provides an Application Programming Interface (API) to handle toxicology data, descriptor calculation and learning algorithms, prediction models, reporting and validation procedures as well as visualization tools. It makes use of the semantic web technologies RDF (Resource Description Framework³⁵) and OWL (Web Ontology Language³⁶) to underline and link its building blocks with ontologies (see Figure C.1). Very recently, following the 240th National Meeting of the American Chemical Society (ACS) in Boston, USA, the Journal of Cheminformatics has started to publish a thematic series [19, 117, 70, 53, 153, 20] on *RDF Technologies in Chemistry*, edited by E. Willighagen and M.P. Braendle. This shows that OpenTox has a sound grasp of contemporary technological developments and makes use of them.

In addition to the API development, the project launched several example applications to show the capabilities, usability and flexibility of the framework. The first prototype application is ToxPredict³⁷. This application enables a user to submit a set of chemical structures and get a prediction for one or more toxicological endpoints with one or more pre-trained prediction models. ToxPredict will be explained in more detail later in this chapter. The second application is ToxCreate³⁸. It allows the user to build her own models for toxicity prediction. The third prototype application is ToxDesc³⁹. ToxDesc is a simple descriptor calculation web interface. Q-edit⁴⁰ provides functionality to automatically fill

32 Project Reference Number: Health-F5-2008-200787 in the HEALTH-2007-1.3-3 program

33 <http://www.oecd.org/dataoecd/33/37/37849783.pdf>,

http://www.oecd.org/document/40/0,3343,en_2649_34377_37051368_1_1_1_1,00.html

34 http://ec.europa.eu/environment/chemicals/reach/reviews_en.htm#annex11

35 www.w3.org/RDF/

36 <http://www.w3.org/TR/owl2-overview/>

37 <http://www.toxpredict.org>

38 <http://www.toxcreate.org>

39 <http://opentox-dev.informatik.tu-muenchen.de:8080/ToxDesc>

40 <https://github.com/alphaville/Q-edit>

out and to edit QPRF (QSAR Prediction Reporting Format) reports. Further applications that demonstrate the functionalities of special algorithms that are integrated in the framework are MaxTox⁴¹ and MakeMNA⁴².

The remainder of this chapter is organized as follows: in the next section the philosophy and rationale of the OpenTox project is explained. Section 8.2 introduces the technology of REST web services, before the OpenTox API is described in Section 8.3. The prototype application ToxPredict is topic of Section 8.4 before the chapter is concluded in Section 8.5.

| Organization | Country |
|--|-------------|
| Douglas Connect | Switzerland |
| Ideaconsult | Bulgaria |
| David Gallagher | UK |
| Superior Health Institute (ISS) | Italy |
| Seascape Learning & JNU | India |
| Technische Universität München | Germany |
| Albert Ludwigs University Freiburg | Germany |
| Fraunhofer Institute for Toxicology & Experimental Medicine | Germany |
| Institute of Biomedical Chemistry of the Russian Academy of Medical Sciences | Russia |
| National Technical University of Athens | Greece |
| In Silico Toxicology | Switzerland |

Table 8.1: Overview of the OpenTox project partners.

41 <http://www.maxtox.org>

42 <http://195.178.207.160/opentox/MakeMNA>

8.1 OpenTox Philosophy and Background

The new European Union (EU) REACH chemical legislation requires the chemical industry to provide information on substances that are produced on the European market or imported to it from December 1st, 2010. The first phase from 2010 to 2018 includes substances above a threshold of 1,000 tons per year, the second phase drastically reduces the threshold to 1 ton per year. The responsibility to generate and submit the requested information lies with the manufacturers and importers. This information should include, for example, human safety and environmental toxicity. Instead of supporting an increased use of test animals, REACH fosters the development of new *in vitro* and *in silico* methods. Additionally, high costs and ethical concerns for laboratory animals have led to a much increased importance of QSAR studies within the drug discovery process [141]. To address this challenge, the European Commission has funded the OpenTox⁴³ project to develop an open source framework that provides unified access to experimental toxicity data, *in silico* models, and validation and reporting procedures. A listing of the partner organizations from academia and industry that were involved in the OpenTox project is given in Table 8.1.

OpenTox relies on open source software to optimize its impact, allow for inspection and review and to attract external contributors. One of the main contributions of OpenTox is that it offers a uniform interface to open source cheminformatics software. And, as it is open source, the list of available tools can easily and quickly be extended. For example, OpenTox integrates open source software and very recent software developments made by the consortium partners like FCDE [14], ToxTree [99], lazar [57] or LoMoGraph [15]. This flexibility and extensibility is a clear advantage over tools like OCHEM [130], a recently published web based QSAR development platform, which is closed-system and relies heavily on proprietary software packages. Very important guidelines in the development process of OpenTox have been the OECD Guidelines for (Q)SAR Validation⁴⁴. These guidelines have been developed in the OECD (Q)SAR project with the goal to enable (Q)SAR application in regulatory context by industry and governments and to improve the regulatory acceptance of QSAR. A brief overview of the contents of the guidelines is given in Table 8.2.

To minimize integration efforts for existing software packages, maximize flexibility and to be able to maintain the framework extensible and in a distributed environment, it was decided to make use of REST web services as the fundamental technology of the OpenTox framework. This technology will be explained in the next section.

⁴³ www.opentox.org

⁴⁴ http://www.oecd.org/document/2/0,3746,en_2649_34377_42926338_1_1_1_1,00.html

| OECD principle | Explanation |
|--|--|
| Defined Endpoint | providing a unified source of well defined and documented toxicity data with a common vocabulary |
| Unambiguous Algorithm | providing transparent access to well documented models and algorithms as well as to the source code |
| Defined Applicability Domain | integrating tools for the determination of applicability domains during the validation of prediction models |
| Goodness-of-Fit, Robustness and Predictivity | providing scientifically sound validation routines for the determination of errors and confidences |
| Mechanistic Interpretation | integrating tools for the inference, correlation or prediction of toxicological mechanisms and the recording of opinions and analysis in reports |

Table 8.2: Overview: OECD Guidelines for (Q)SAR Validation

8.2 REST Web Services

The technological foundation for the OpenTox framework are REST web services. The Representational State Transfer (REST) - introduced in 2000 by Fielding [37] in his dissertation - is a software architectural style that is based on the biggest distributed application, the world wide web and its basic transfer protocol HTTP. It adopts the basic HTTP operations GET, POST, PUT and DELETE (CRUD operations - create, read, update, delete) to enable communication between service and client in a uniform but still generic way. The focus of REST is to enable access to named resources through the consistent interface of the CRUD operations.

If you had to explain how REST works to a non-IT person, you would probably tell her that basically, REST is a language for machines that works in a similar way like our human language. It uses nouns and verbs. All objects correspond to nouns, e.g. web pages, pictures, data sets, algorithms, and the four HTTP operations GET, POST, PUT and DELETE correspond to the verbs. Every object can be addressed via a unique name (URI) and can have several representations, e.g. HTML to be shown in a web browser, XML for service communication, JPG if it is an image, SDF if it is chemical structure data. This setting enables services on the web to talk to each other in a standardized way. Web services that conform to the REST constraints, are referred to as being RESTful [107]. One of the main features of REST web services is that they are stateless. This aspect makes them scalable even for big service constructs.

The W3C defines a *web service* as

a software system designed to support inter-operable machine-to-machine interaction over a network. It has an interface described in a machine-processable format [...]⁴⁵.

⁴⁵ <http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/>

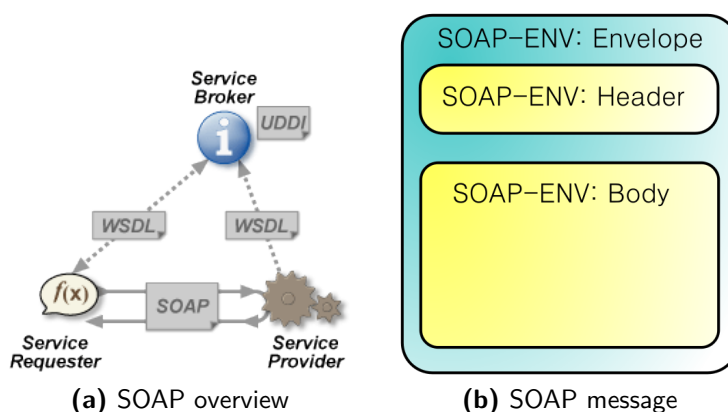


Figure 8.1: SOAP web service process overview and message structure⁴⁶

Up to now, there exist two major technological approaches for web services: SOAP and REST. SOAP (Simple Object Access Protocol) is the older technology that has its own protocol and packs information in a so-called envelope (see: Figure 8.1b). A graphical overview how SOAP services work is given in Figure 8.1a. The focus - in comparison to REST - lies on exposing applications as services where each application can have a different interface. In O'Reilly's [107] "RESTful WebServices", the authors distinguish between *Web Services* (also called "Big Web Services") and *web services* (REST). They say that, in practice, SOAP services are mainly used to implement Remote Procedure Call applications via HTTP. One advantage of "Big Web Services" is that there are a lot of development tools with which one can generate RPC-style web service code automatically. In that case, using the much simpler REST web services makes less sense. Advantages of REST web services are, amongst others, their scalability, the independent deployment of single components and the composition of services. Single REST services can easily be used together. More clearly, there is nothing like REST services per se. If we want to be exact, there are only resources that are made available. Through the universal address space of URIs, it is very easy to cross application borders. A document just links a resource in a different organization. Amongst others that are reasons why OpenTox decided to make use of REST web services.

The four HTTP operations are the main features common to all building blocks of the REST-based API (see Figure 8.2 for a graphical overview). An overview of this API is given in the following section.

⁴⁶ <http://en.wikipedia.org/wiki/File:Webservices.png> and <http://en.wikipedia.org/wiki/File:SOAP.svg>

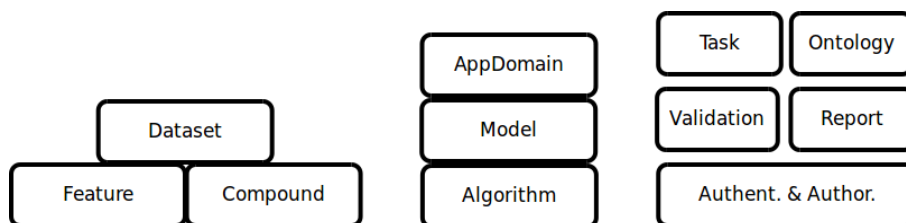


Figure 8.2: OpenTox API building blocks. Every block corresponds to a service type and can be addressed via the methods GET, POST, PUT and DELETE. Authent. & Author. is short for Authentication and Authorization.

8.3 The OpenTox Application Programming Interface

To assure reliable interoperability between the various OpenTox web services, a well-defined API is required. The OpenTox API specifies, how each OpenTox web service can be used, how interfaces of new services have to be implemented and how the returned resources should look like. It further specifies the HTML status codes returned in case of successful operations as well as HTML errors codes. The specifications for the OpenTox API are available on the OpenTox website⁴⁷. The different development stages of the API are also documented there. The most recent version is 1.2, which is (by June 2011) in the transition to become the stable version. The choice of employing web services allows the complete framework to operate in different locations, independent of operating systems and underlying implementation details like the programming language or platform.

Figure 8.6 shows the OpenTox resources modeled in the OpenTox Ontology. These resources are provided by the various OpenTox web services. The links between the components reflect interaction between the respective web services. The model web service

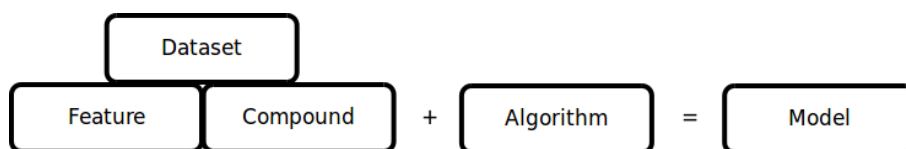


Figure 8.3: OpenTox API building blocks. Blocks combined to learn QSAR prediction models.

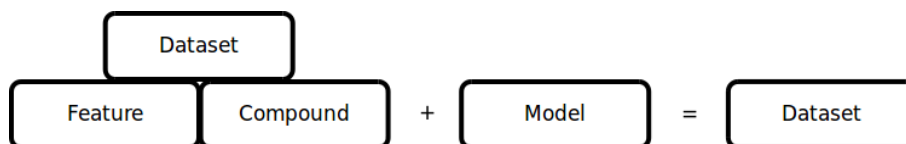


Figure 8.4: OpenTox API building blocks. Blocks combined to make toxicology predictions.

⁴⁷ <http://www.opentox.org/dev/apis>

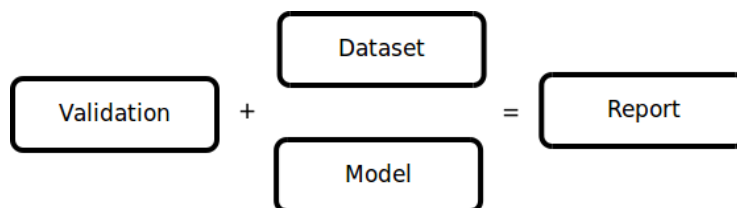


Figure 8.5: OpenTox API building blocks. Blocks combined to create prediction report.

provides access to prediction models. To create a prediction model, a dataset instance linking compounds and associated features is fed to a learning algorithm (compare Figure 8.3). Datasets are stored in the dataset web service. A dataset contains data entries, which are chemical compounds, as well as their feature values. Features are defined as objects representing a property of a compound, including descriptors, endpoints and predictions. Different representations and conformations of a chemical compounds can be accessed from the compound web service. The feature web service provides access to the available features. The API building blocks used for a prediction workflow are a dataset (compounds plus features) for which the predictions should be made and a prediction model. The result of the prediction process is again a dataset. It now contains a prediction feature that stores the prediction value for each compound (see Figure 8.4). To ensure comparability and thorough validation, the OpenTox API has its own validation building block. To validate a learned prediction model, the dataset and model building blocks are necessary in addition to the validation block (see Figure 8.5). The validation at the moment allows for cross-validation, bootstrap validation and train-test-split validation. The validation results are available as reports in various formats, e.g. HTML or pdf. The task web service supports long-running, asynchronous processes. The ontology web service provides meta information from relevant ontologies (which can be accessed using SPARQL queries⁴⁸), as well as lists of available services. Since API version 1.2 an approach to Authentication and Authorization is also specified in the API. The approach is based on the OpenSSO/OpenAM⁴⁹ technology and coupled to the OpenLDAP server used for maintaining OpenTox user accounts. All OpenTox resources have representations providing information about the type of resource, and what the service accepts as input such as tuning parameters. Most algorithms and model resources in OpenTox are available in multiple representations. The RDF representation, and in particular its XML formatted variant, was chosen as the master data exchange format. Figure C.2 shows the Algorithm API in a tabular overview, as an example of how the API is structured in detail.

48 <http://www.w3.org/TR/rdf-sparql-query/>

49 <http://forgerock.com/downloads.html>

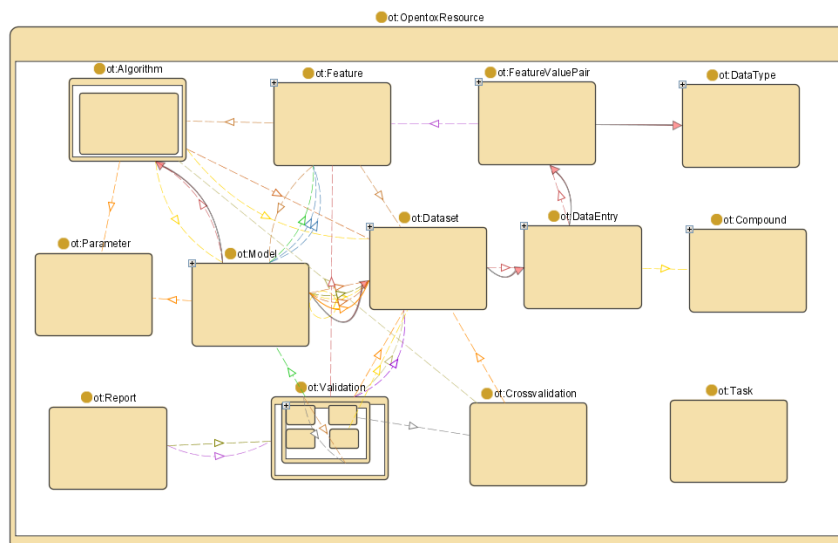


Figure 8.6: Relationships between OpenTox Resources modeled in the OpenTox Ontology

8.4 Prototype Application: ToxPredict

To show the usability, flexibility and how a real-world application that is built on top of the OpenTox framework API could look like, OpenTox - especially IdeaConsult - developed the ToxPredict prototype application located at www.toxpredict.org. ToxPredict is aimed at the user having no or little experience in QSAR predictions. It offers an easy-to-use, web-based user interface, allowing to enter a chemical structure and to obtain in return a toxicity prediction for one or more target values.

The ToxPredict web application, offers several top-level functionalities. First, the user can log in, using a valid OpenTox user account. The user account is identical with the one obtained from a registration on the OpenTox web site and the OpenTox mailing list. It also enables the user to save progress to his personal account. Still, the application is usable without login using a default guest user id. The second functionality is to predict toxicological hazard, the third to browse available datasets and models. The web application is started by default with the prediction site (see Figure C.3) asking the user to enter or search for a structure for which he would like to apply some OpenTox models. This can be done by structure drawing, or search. Any identifier (CAS, name, EINECS), SMILES, InChi or OpenTox compound or dataset URL can be entered. The input type will be guessed automatically by the system. The user can also choose between exact structure search, substructure search or similarity search. Once the structure is found, the system switches to the *View results* page. Figure 8.7 shows this page for benzoic acid. The top of the page shows a two-dimensional image of the structure and gives, amongst others, information about identifiers, IUPAC name, SMILES string and REACH registration date. The bottom half of the page offers two tabs, one for predictions and one to view datasets containing the query compound. In the predictions tab, a list of available

prediction models is listed and the user can either choose single models that shall be used to predict properties of the compound or use the *Run All* button to use all available prediction models at once. A symbol next to each model indicates that a calculation is in progress and as soon as the calculation is finished the corresponding section of the page is updated. Everything is done dynamically on the same web page using web 2.0 technology to reduce navigation between pages and create a more desktop application-like user feeling. In Figure 8.7, for example, the pKa for benzoic acid has been predicted to be 3.52.

While current functionality may appear to an end-user not much different from a stand-alone prediction application like ToxTree [99], the back-end technology provides a very flexible means for integrating datasets, models and algorithms, developed by different software technologies and organizations and running at remote locations. Especially in combination with the ToxCreate application that has been developed to build OpenTox prediction models, ToxPredict shows the versatility and flexibility of the OpenTox API. Approaches like ToxPredict are hopefully able to improve the acceptance of *in silico* or (Q)SAR toxicology models for regulatory purposes and make everyday toxicology prediction easier.

8.5 Conclusion

This chapter introduced the distributed, REST web service based framework OpenTox. Because of the highly flexible and decentralized architecture that is based on REST web services, it is easy to integrate state-of-the-art methods and algorithms from machine learning, predictive toxicology and other relevant fields. The positive impact of the project is documented, e.g., by the broad interest of industry, academia and government representatives to attend the final OpenTox InterAction meeting on “Innovation in Predictive Toxicology” hosted at the Technische Universität München, in August 2011. The conference program features industry speakers from Proctor & Gamble, Novozymes, Astra Zeneca, RIVM, Cyprotex Discovery Ltd., Sanofi-Aventis, Pharmatroppe Ltd., Leadscope Inc. and Biowisdom Ltd.. Industry and Government organizations represented are the Fraunhofer Institute, the EBI, Health Canada, the US EPA, the Cambridge Cell Networks, the Helmholtz Centre Munich, OpenSource Drug Discovery and additionally several European universities. Furthermore, one of the main contributions of the OpenTox project, the developed application programming interface, has already been used to integrate OpenTox webservices with the Bioclipse framework [154]. This enables a more thorough and versatile computational toxicology analysis.

Nevertheless, there are still improvements to be made. One option for improvement would be to either add another mandatory exchange format to the now only mandatory format, RDF/XML, or try to find speed up possibilities. In my opinion, this exchange format is too bulky and slows down the data transfer between web services. Another possible step for improvement or enhanced acceptance would be to abandon the Authentication & Authorization (A&A) functionalities and focus on easily downloadable and installable

services instead. A&A for distributed web services is, up to now, a more or less unsolved technical problem that would by itself be worth a project the size of OpenTox. Also, pharmaceutical companies still are very careful when it comes to transferring confidential structure data over the web. Most of the time transferring confidential data is not at issue at all for those companies. Furthermore, removing the A&A functionality would decrease the frameworks complexity drastically.

Overall, the OpenTox project was a success. It showed that collaborative work using only open source software can build a powerful, extensible and flexible framework for predictive toxicology. Hopefully, the project will be maintained and updated after the project finishes in August 2011. This could be done by the open source community, a further collaborative project that enhances the frameworks infrastructure, one or more of the involved project partners or a combination of those possibilities.

[Help](#)

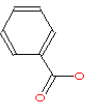
ToxPredict

WELCOME
Log in
PREDICT

Search structure
Upload structure
View results
BROWSE
Datasets
Models
MY WORKSPACE
My uploads

Please select the structure(s) for which you would like to apply some OpenTox models.

Draw



JME Editor courtesy of Peter Ertl, Novartis

Search

Query:

Search mode

- Auto detect
- Exact structure
- Substructure search
- Similarity search

Enter any identifier (CAS, Name, EINECS) or SMILES or InChI or URL of OpenTox compound or dataset. ToxPredict will guess the input type automatically. SMILES may be entered manually into the text field or alternatively use the JME editor to draw the structure.

Developed by IdeaconSult Ltd. 2011

More...

Figure 8.7: ToxPredict entry page showing a structure search by using the drawing editor

CHAPTER 9

Summary and Outlook

In this final chapter, I first give a summary of my main contributions to small molecule similarities in QSAR research and predictive toxicology before I discuss possible further directions for research in this area.

9.1 Summary

In this thesis I introduced three enhancements to small molecule similarity for quantitative structure-activity relationship modeling and cheminformatics applications. This includes molecular descriptors (features) based on small molecule similarity, incorporating background knowledge into structural similarity measures and learning, transferring and adapting dissimilarities from related learning problems.

After giving a brief overview of quantitative structure-activity relationships and similarity measures used in cheminformatics and after discussing related work in Chapters 2 and 3, I discuss the usage of similarity descriptor vectors in QSAR modeling. The similarities that constitute the descriptor vector are calculated with respect to a set of reference compounds that can be derived with three variants: literature review, clustering the set of actives, and clustering a set that is representative of the chemical space. In the case of the clustering variants, cluster representatives are randomly selected and used as reference compounds. The similarities themselves are consensus similarities built from diverse fingerprint similarities. In an experimental comparison with structural descriptors (BBRC and ECFP) the similarity descriptors perform quite well. In addition, I showed that the similarity descriptors encode information that is complementary to that of the structural descriptors and a combination of both further enhances the predictive performance of the derived QSAR models.

My second contribution aims at improving structural similarity measures by incorporating background knowledge. I tested my approach in a virtual screening setting using the DuD database as background dataset. To incorporate the background knowledge on binding-relevant structural moieties, I extend the structural similarity measure with a weighted fingerprint similarity. In a first experiment I derive the important structural

features by hand and visual analysis, in a second experiment I use a data mining approach to automatically derive those substructural features. The experimental evaluation showed that the incorporation of background knowledge significantly improves the enrichment factors and mean ranks in the resulting similarity rankings in both scenarios: by-hand and data mining based knowledge extraction.

My third approach, Adapted Transfer of Distance Measures, aims at problem settings where only few data (target dataset) for training a prediction model is available but sufficient data (source dataset) for a related problem is present. I extended the idea of inductive transfer that I combine with distance learning, by two variants that adapt the transferred bias in the form of weights to the characteristics of the target dataset: bounded and penalized adaptation. In the bounded variant the adaptation is constrained by an ϵ -environment around the transferred weights, in the penalized variant the adaptation is constrained by penalizing large deviations from the transferred bias. My first set of experiments uses hand-selected pairs of related datasets to evaluate my approach. In a comparison with inductive transfer and distance learning alone, the method appears to be a good compromise that works well both with large and small amounts of training data. In a second set of experiments I derived the related datasets in a semi-automatic way using a similarity measure on biochemical assays. The most similar, relevant assay is used for transfer. The results are comparable to the results of the first experiments.

After presenting my main contributions I discuss the pairwise relations between the three contributions and if or how they can be combined in Chapter 7. Finally in Chapter 8 I gave an overview of the OpenTox project. In the scope of the OpenTox project an open source distributed REST webservice based framework for toxicology prediction was developed. It provides a flexible API for learning and descriptor calculation algorithms, data storage and retrieval, validation, visualization and reporting. Several prototype applications have been built upon this API of which the ToxPredict web-application is presented in greater detail.

9.2 Outlook

My work on enhanced small molecule similarity for QSAR modeling and cheminformatics applications improved small molecule similarity measures or their usage in three application relevant areas. Nevertheless, there is no doubt that there is room for improvement not only in the design of similarity measures for small molecules but also in the way similarity measures are used. Considering the usage of similarities as descriptors, as in my Similarity Boosted QSAR approach, incorporation of further similarity measures could be beneficial. Pharmacophore-based or maximum common substructure (MCS) based similarities are two examples that could contain additional information that is to some extent complementary to the information encoded so far. In practice however, there will always be a trade-off between improvements in predictive accuracy or performance and usability and runtime. Especially the MCS features are computationally very intensive. A

lot of development opportunities can also be expected for the selection of reference compounds based on clustering. Here, the random selection of cluster representatives could be either replaced by a structural analog to a median or by a mere structural core instead of a complete structure. Also, the clustering itself could be optimized by the adoption of hierarchical clustering ideas and by simple parameter optimization.

If I think about possible improvements to the incorporation and mining of background knowledge approach, the first thing that comes to mind is an optimization of the weights of the combined similarities. It is improbable that the arbitrary choice of $\frac{1}{3}$ as weight for the similarity extension is the optimal choice. Moreover, the data mining based extension can further be optimized. One way to do this could be to mine the binding information in the PDB files of complexes of the known binders with the target protein for information which substructural moieties are relevant, if such complexed structure information is available. A visualization of such information encoded in the PDB file for 1HWI and fluvastatin is shown in Figure 9.1. Especially H-bond donor-acceptor sites or disulfide sites are of interest in this case. Checking if such information on complexes is available in the PDB [10] can be done automatically via the search functionalities that allow for searches using a ligand as query.

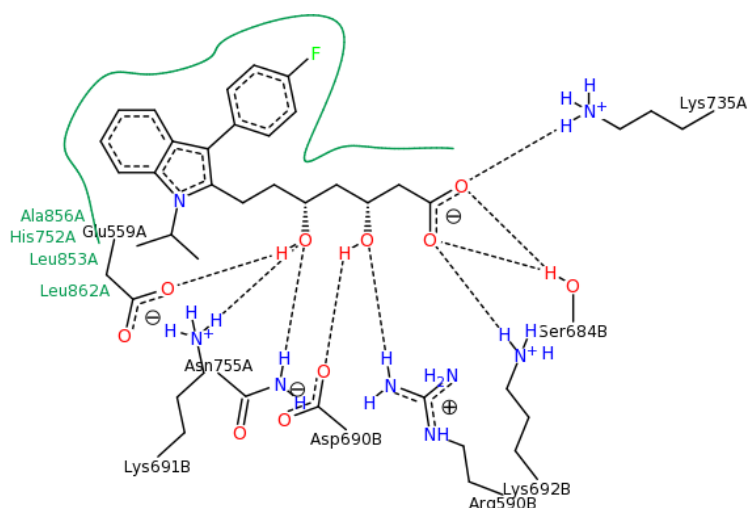


Figure 9.1: Interactions for PDB instance 1HWI:115:B:1. The image was created with the PoseView software [129].

There has been done a lot of research in the areas of distance learning and inductive transfer that leads us to the conclusion that there is no room for great improvements for my adapted transfer approach considering the transfer and adaptation process. I believe, however, that there is room for improvement when it comes to the selection of related tasks or problems in the domain of biological, biochemical or chemical data. Especially the construction and exploitation of ontological information could be highly beneficial in relating datasets with biological target variables to each other. A first example how ontology information can be incorporated into a toxicology prediction framework is given by the OpenTox project. To enable an automatic usage the vast amount of knowledge

available for those biological assays would have to be encoded in a clearly defined ontology. This could render the visual inspection of proposed related assays obsolete, as the relations defined on the ontological entities could be used for this process.

A final aspect that could lead to great improvements in QSAR and cheminformatics research in general is the promotion of the open data idea. A very good example for scientific progress enabled by open data is bioinformatics. The immense amounts of genetics and proteomics data available for bioinformaticians not only in industry but also in academia has allowed this relatively young scientific discipline to flourish over the last ten years. Chemical data, particularly when coupled to pharmaceutical research, has never been made public in comparable amounts so that academic research is condemned to use small often medium quality datasets, and knowledge sharing in between institutions and companies is highly complex or even impossible. I think that changes in this way of thinking could be to the benefit of all especially to patients.

APPENDIX A

Additional Material for Chapter 4

A.1 \mathcal{R}_{LIT} Reference Compounds

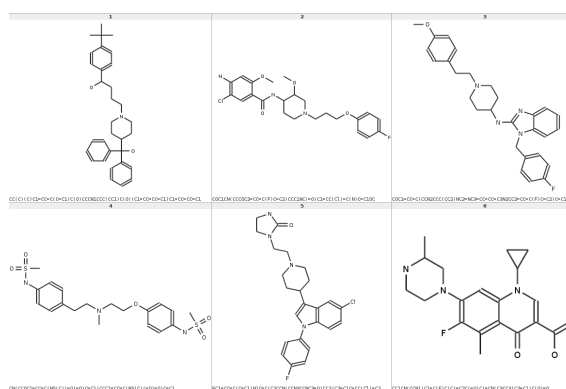


Figure A.1: Reference compounds for the hERG data.

- 1 CC(C)(C)C1=CC=C(C=C1)C(CCCN2CCC(CC2)C(C3=CC=CC=C3) ...
... (C4=CC=CC=C4)O)O
- 2 COC1CN(CCC1NC(=O)C2=CC(=C(C=C2OC)N)Cl)CCCOC3=CC=C(C=C3)F
- 3 COC1=CC=C(C=C1)CCN2CCC(CC2)NC3=NC4=CC=CC=C4N3C ...
... C5=CC=C(C=C5)F
- 4 CN(CCC1=CC=C(C=C1)NS(=O)(=O)C)CCOC2=CC=C(C=C2)NS(=O)(=O)C
- 5 C1CN(CCC1C2=CN(C3=C2C=C(C=C3)Cl)C4=CC=C(C=C4)F) ...
... CCN5CCNC5=O
- 6 CC1CN(CCN1)C2=C(C(=C3C(=C2)N(C=C(C3=O)C(=O)O)C4CC4)C)F

Table A.1: hERG reference compounds in SMILES format.

```

1  C1=C2C(=CC(=C1Cl)Cl)OC3=CC(=C(C=C3O2)Cl)Cl
2  C1=CC(=C(C(=C1)Cl)C=NN=C(N)N)Cl
3  C1=CC=C2C3=C4C(=CC2=C1)C=CC5=C4C(=CC=C5)C=C3
4  C1=CC(=C(C=C1C2=CC(=C(C(=C2)Cl)Cl)Cl)Cl)Cl
5  CC1=C2CCC3=C2C(=CC4=C3C=CC5=CC=CC=C54)C=C1
6  C1=CC=C(C=C1)C2=CC(=O)C3=C(O2)C=CC4=CC=CC=C43
7  C1=CC=C2C(=C1)NC(=N2)C3=CSC=N3
8  CC1=CN=C(C(=C1OC)C)CS(=O)C2=NC3=C(N2)C=C(C=C3)OC
9  C1C(C(C2=CC=CC=C21)N)O
10 SC1=CC=CC=C1N
11 CN(C1=CC=CC=C1)N
12 C1=CC2=C(C=CC=C2N)C(=C1)N

```

Table A.2: SMILES representation of AhR reference compounds.

```

1  C[C@]12CCC3c4ccc(cc4CCC3C1CC[C@@H]2O)O
2  c1cc(ccc1c2c(c3ccc(cc3s2)O)C(=O)c4ccc(cc4)OCCN5CCCCC5)O
3  c1cc(ccc1c2cc3cc(cc(c3o2)Br)O)O
4  c1cc(ccc1C2C3=C(CCOc4c3ccc(c4)O)c5ccc(cc5O2)F)OCCN6CCCCC6
5  CN(C)CCOc1ccc(cc1)[C@H]2[C@H](Sc3cc(ccc3O2)O)c4ccc(c4)O
6  c1cc(ccc1C2c3cc(ccc3Cc4c2c5ccc(cc5cc4)O)O)OCCN6CCCCC6
7  CC/C(=C(/CC)\c1ccc(cc1)O)/c2ccc(cc2)O

```

Table A.3: SMILES representation of ER reference compounds.

```

1  CN(CCOc1ccc(cc1)C[C@@H](C(=O)O)Nc2ccccc2C(=O)c3ccccc3)c4ccccc4
2  c1ccc(cc1)NC(=O)c2cc(ccc2Cl)N(=O)=O
3  Cc1c(nc(o1)c2ccccc2)CCOc3ccc(cc3)CC4C(=O)NC(=O)S4
4  Cc1cccc(c1)C(=O)c2ccccc2NC(Cc3ccc(cc3)OCCN(C)c4nc5ccccc5o4)C(=O)O
5  COC(=O)C(Cc1ccc(cc1)OCCn2c3ccc(cc3sc2=O)C(=O)c4ccccc4)C(=O)O
6  Cc1c(nc(o1)c2ccccc2)CCOc3ccc(cc3)CC(C)(C(=O)O)Oc4ccccc4
7  CN(CCOc1ccc(cc1)CC2C(=O)NC(=O)S2)c3ccccc3

```

Table A.4: SMILES representation of PPAR γ reference compounds.

```

1  c1cc(c(cc1O)c2c(cc(cc2I)CC(=O)O)I)I)O
2  CC1(C2CCC(C1C2)NC(=O)c3cc(ccc3O)Oc4c(cc(cc4Cl)n5c(=O) ...
... [nH]c(=O)cn5)Cl)C
3  COc1ccccc1CCNC(=O)c2cc(ccc2O)Oc3c(cc(cc3Br)CC(=O)O)Br
4  c1ccc(cc1)C(CNC(=O)c2cc(ccc2O)Oc3c(cc(cc3Br)CC(=O)O)Br)c4ccccc4
5  c1cc(c(cc1O)c2c(cc(cc2Cl)n3c(=O)[nH]c(=O)cn3)Cl)C(=O)N4CCOCC4)O

```

Table A.5: SMILES representation of THR reference compounds.

```

1  c1ccc2c(c1)C(=O)c3ccccc3C2(Cc4cnccc4)Cc5cnccc5
2  CCN1CCOc2c1cc(cc2)C(C)NC(=O)/C=C/c3ccccc3F
3  CC(c1ccc(c(c1)N2CCOCC2)F)NC(=O)/C=C/c3ccc(cc3)F
4  CC(c1ccc2c(c1)OCO2)NC(=O)/C=C/c3ccccc3Cl
5  CCN1CCCc2c1cc(cc2)C(C)NC(=O)/C=C/c3cc(ccc3F)F

```

Table A.6: SMILES representation of KCNQ2 reference compounds.

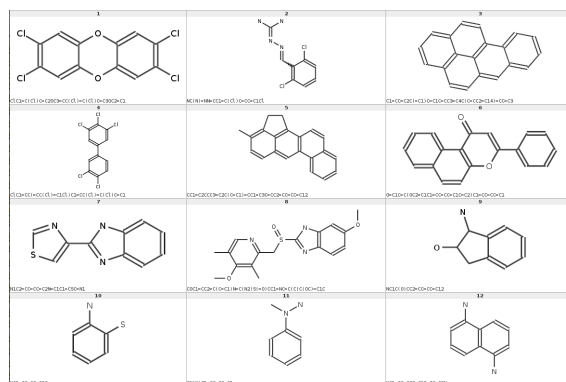


Figure A.2: Reference compounds for the AhR data.

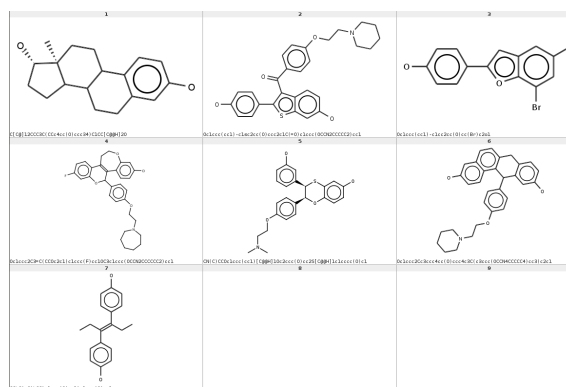
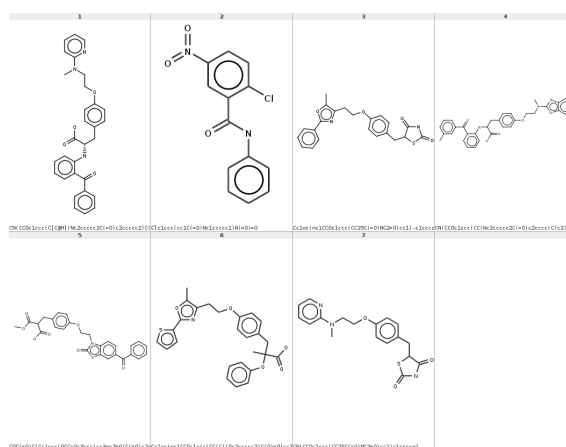


Figure A.3: Reference compounds for the ER data.

Figure A.4: Reference compounds for the PPAR γ data.

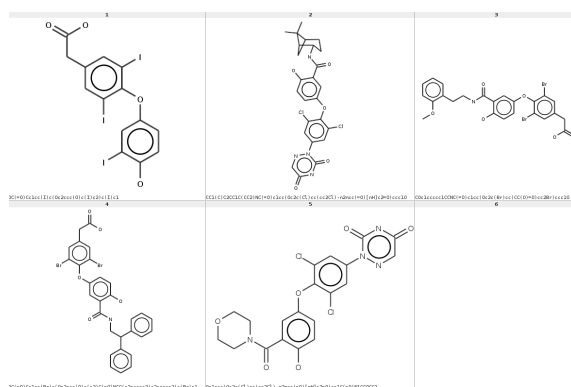


Figure A.5: Reference compounds for the THR data.

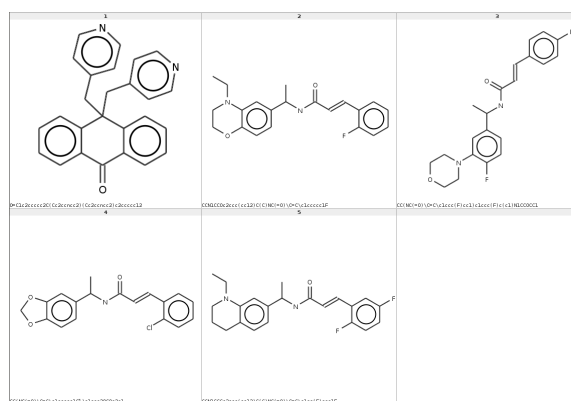


Figure A.6: Reference compounds for the KCNQ2 data.

```

1  c1ccc(cc1)C(c2ccccc2)([C@@H]3CCN(C3)CCc4ccc5c(c4)CCO5)C(=O)N
2  c1cc2c(cc1)Sc3c(cccc3)N2C[C@H]4[C@H]5CCN(C4)CC5
3  CC(C)N(CCC(c1ccccc1)c2cc(ccc2O)CO)C(C)C
4  c1ccc(cc1)c2ccccc2C3=CN4CCC3CC4
5  c1ccc2c(c1)C(=O)Nc3ccnc3N2C(=O)CN4CCNCC4
6  C[N+]1(C2CCC1CC(C2)CC(CO)(c3ccccc3)c4ccccc4)C
7  CCCCCSc1c(ncn1)O[C@@H]2CN3CCC2C3
8  CN1CCN(CC1)C2=Nc3cc(ccc3Nc4c2cccc4)Cl
9  Cc1ccc(c(c1)[C@@H](CCN(C(C)C)C(C)C)c2ccccc2)O
10 c1ccc(cc1)C(c2ccccc2)(C(=O)OC3CC4CCC(C3)[N+]45CCCC5)O
11 CCN(CC)CC#CCOC(=O)C(c1ccccc1)(C2CCCC2)O
12 CN1CCC=C(C1)c2c(nsn2)OCCCCCOc3c(nsn3)C4=CCCN(C4)C
13 CN1[C@@H]2CC[C@H]1CC(C2)OC(=O)C(CO)c3ccccc3
14 CC(CC(=O)O)N1CCC(=C2c3ccccc3OCc4c2ccc(c4)F)CC1

```

Table A.7: SMILES representation of M1 reference compounds.

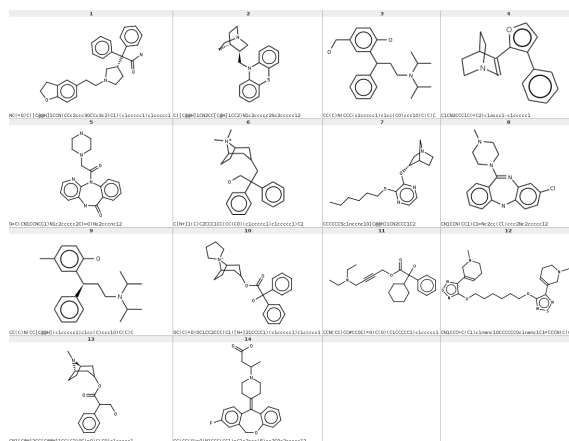


Figure A.7: Reference compounds for the M1 data.

A.2 Additional Result Tables

| Dataset | RF | SVM |
|---------|--------|--------|
| hERG | 34737 | 30713 |
| AhR | 143038 | 558594 |
| ER | 9263 | 7528 |
| SRC-1 | 6921 | 5362 |
| THR | 5456 | 5637 |
| KCNQ2 | 26322 | 57498 |
| M1 | 6103 | 5167 |

Table A.8: Example of CPU runtimes in seconds. Shown are ten-fold cross-validation running times for the descriptor set $SD(\mathcal{S}_{ALL}, \mathcal{R}^{act})$

| Dataset | Random Forests | | | | | |
|---------|----------------|----------------|----------------|----------------|--------------|---------------------|
| | $\{s_{MK}\}$ | $\{s_{topo}\}$ | $\{s_{ECFP}\}$ | $\{s_{FCFP}\}$ | $\{s_{AP}\}$ | \mathcal{S}_{ALL} |
| hERG | 0.592 | 0.551 | 0.566 | 0.566 | 0.566 | 0.613 |
| AhR | 0.738 | 0.632 | 0.739 | 0.711 | 0.717 | 0.772 |
| ER | 0.627 | 0.645 | 0.672 | 0.662 | 0.661 | 0.711 |
| SRC-1 | 0.625 | 0.599 | 0.625 | 0.627 | 0.596 | 0.696 |
| THR | 0.537 | 0.572 | 0.583 | 0.569 | 0.571 | 0.650 |
| KCNQ2 | 0.589 | 0.594 | 0.625 | 0.650 | 0.632 | 0.686 |
| M1 | 0.653 | 0.592 | 0.653 | 0.625 | 0.608 | 0.653 |

Table A.9: RandomForest (RF) ten-fold cross-validation results for using single similarity measures and their combination. The reference set is always \mathcal{R}^{lit} .

| Dataset | SVM | | | | | |
|---------|--------------|----------------|----------------|----------------|--------------|---------------------|
| | $\{s_{MK}\}$ | $\{s_{topo}\}$ | $\{s_{ECFP}\}$ | $\{s_{FCFP}\}$ | $\{s_{AP}\}$ | \mathcal{S}_{ALL} |
| hERG | 0.596 | 0.548 | 0.577 | 0.568 | 0.580 | 0.602 |
| AhR | 0.701 | 0.618 | 0.725 | 0.698 | 0.725 | 0.772 |
| ER | 0.640 | 0.663 | 0.698 | 0.676 | 0.676 | 0.725 |
| SRC-1 | 0.640 | 0.624 | 0.614 | 0.629 | 0.638 | 0.694 |
| THR | 0.527 | 0.601 | 0.564 | 0.580 | 0.607 | 0.633 |
| KCNQ2 | 0.597 | 0.597 | 0.664 | 0.653 | 0.640 | 0.697 |
| M1 | 0.646 | 0.588 | 0.638 | 0.649 | 0.653 | 0.680 |

Table A.10: Support Vector Machine (SVM) ten-fold cross-validation results for using single similarity measures and their combination. The reference set is always \mathcal{R}^{lit} .

| Dataset | Random Forests | | | | | \mathcal{S}_{ALL} |
|---------|----------------|--------------|----------------|----------------|----------------|---------------------|
| | $\{s_{MK}\}$ | $\{s_{AP}\}$ | $\{s_{FCFP}\}$ | $\{s_{ECFP}\}$ | $\{s_{topo}\}$ | |
| hERG | 0.587±0.011 | 0.565±0.013 | 0.559±0.012 | 0.559±0.013 | 0.551±0.012 | 0.608±0.013 ● |
| AhR | 0.735±0.004 | 0.736±0.005 | 0.705±0.005 | 0.714±0.005 | 0.629±0.005 | 0.766±0.004 ● |
| ER | 0.627±0.015 | 0.668±0.013 | 0.659±0.015 | 0.663±0.015 | 0.646±0.014 | 0.708±0.013 ● |
| SRC-1 | 0.614±0.019 | 0.615±0.018 | 0.613±0.018 | 0.583±0.017 | 0.600±0.015 | 0.681±0.018 ● |
| THR | 0.538±0.019 | 0.573±0.017 | 0.571±0.017 | 0.564±0.017 | 0.563±0.020 | 0.631±0.017 ● |
| KCNQ2 | 0.587±0.009 | 0.627±0.009 | 0.650±0.007 | 0.623±0.009 | 0.589±0.008 | 0.686±0.009 ● |
| M1 | 0.646±0.018 | 0.645±0.018 | 0.624±0.018 | 0.611±0.018 | 0.581±0.020 | 0.651±0.017 ● |

●/○ statistically significant improvement in all five cases/some of the five cases.

Table A.11: Statistical significance analysis of single similarities and their combination. Null hypothesis is that the combination is not better than a single similarity. Shown are mean accuracy values \pm standard deviations over 100 hold-out runs^a. The reference set is always \mathcal{R}^{lit} .

^a Please note that standard deviations only quantify the scatter among the values and do not allow for any conclusions on the statistical significance of the difference of the means [24].

| Dataset | BBRC | Random Forests | | |
|---------|-------------|---------------------|---------------------|--------------------|
| | | \mathcal{R}^{lit} | \mathcal{R}^{act} | \mathcal{R}^{db} |
| hERG | 0.633±0.014 | 0.608±0.013 ○ | 0.634±0.015 | 0.633±0.013 |
| AhR | 0.782±0.005 | 0.766±0.004 ○ | 0.779±0.008 ○ | **± ** |
| ER | 0.710±0.014 | 0.708±0.013 | 0.731±0.014 ● | 0.722±0.013 ● |
| SRC-1 | 0.720±0.017 | 0.681±0.018 ○ | 0.728±0.018 ● | 0.725±0.020 ● |
| THR | 0.650±0.017 | 0.631±0.017 ○ | 0.669±0.016 ● | 0.636±0.020 ○ |
| KCNQ2 | 0.677±0.009 | 0.686±0.009 ● | 0.738±0.010 ● | 0.727±0.009 ● |
| M1 | 0.645±0.022 | 0.651±0.017 ● | 0.669±0.017 ● | 0.663±0.020 ● |

●/○ statistically significant improvement/degradation wrt. column BBRC

Table A.12: Statistical significance analysis of BBRC and the variants for finding reference molecules. Null hypothesis is, that there is no improvement compared to column BBRC. Shown are mean accuracy values \pm standard deviations over 100 hold-out runs^a. $\mathcal{S} = \mathcal{S}_{ALL}$.

^a Please note that standard deviations only quantify the scatter among the values and do not allow for any conclusions on the statistical significance of the difference of the means [24].

| Dataset | ECFP _{r1} | Random Forests | | |
|---------|--------------------|---------------------|---------------------|--------------------|
| | | \mathcal{R}^{lit} | \mathcal{R}^{act} | \mathcal{R}^{db} |
| hERG | 0.661±0.012 | 0.608±0.013 ○ | 0.634±0.015 ○ | 0.633±0.013 ○ |
| AhR | 0.792±0.015 | 0.766±0.004 | 0.779±0.008 | **± ** |
| ER | 0.749±0.014 | 0.708±0.013 ○ | 0.731±0.014 ○ | 0.722±0.013 ○ |
| SRC-1 | 0.770±0.016 | 0.681±0.018 ○ | 0.728±0.018 ○ | 0.725±0.020 ○ |
| THR | 0.680±0.018 | 0.631±0.017 ○ | 0.669±0.016 ○ | 0.636±0.020 ○ |
| KCNQ2 | 0.763±0.009 | 0.686±0.009 ○ | 0.738±0.010 ○ | 0.727±0.009 ○ |
| M1 | 0.679±0.021 | 0.651±0.017 ○ | 0.669±0.017 ○ | 0.663±0.020 ○ |

●/○ statistically significant improvement/degradation wrt. column ECFP

Table A.13: Statistical significance analysis of ECFP_{r1} and the variants for finding reference molecules. Null hypothesis is, that there is no improvement compared to column ECFP_{r1}. Shown are mean accuracy values \pm standard deviations over 100 hold-out runs^a. $\mathcal{S} = \mathcal{S}_{ALL}$.

^a Please note that standard deviations only quantify the scatter among the values and do not allow for any conclusions on the statistical significance of the difference of the means [24].

| Dataset | Random Forests | | | | | |
|---------|----------------|-------------|------------------|--------------------|-------------|------------------|
| | BBRC | | | ECFP _{r1} | | |
| | <i>SN</i> | <i>SP</i> | Δ_{SN-SP} | <i>SN</i> | <i>SP</i> | Δ_{SN-SP} |
| hERG | 0.626±0.020 | 0.642±0.020 | -0.016 | 0.619±0.012 | 0.699±0.041 | -0.080 |
| AhR | 0.779±0.006 | 0.787±0.011 | -0.008 | 0.788±0.010 | 0.797±0.020 | -0.009 |
| ER | 0.696±0.018 | 0.726±0.020 | -0.030 | 0.739±0.020 | 0.760±0.023 | -0.021 |
| SRC-1 | 0.713±0.023 | 0.730±0.026 | -0.017 | 0.755±0.020 | 0.789±0.027 | -0.034 |
| THR | 0.645±0.028 | 0.657±0.024 | -0.012 | 0.677±0.027 | 0.685±0.030 | -0.008 |
| KCNQ2 | 0.679±0.012 | 0.674±0.010 | 0.005 | 0.764±0.011 | 0.760±0.013 | 0.004 |
| M1 | 0.641±0.029 | 0.649±0.028 | -0.008 | 0.676±0.032 | 0.684±0.031 | -0.008 |

Table A.14: Mean sensitivity (*SN*) and specificity (*SP*) values including standard deviations over 100 hold-out runs^a. Δ_{SN-SP} is the difference of sensitivity and specificity.

^a Please note that standard deviations only quantify the scatter among the values and do not allow for any conclusions on the statistical significance of the difference of the means [24].

| Dataset | Random Forests | | | | | |
|---------|---------------------|-------------|------------------|---------------------|-------------|------------------|
| | \mathcal{R}^{lit} | | | \mathcal{R}^{act} | | |
| | <i>SN</i> | <i>SP</i> | Δ_{SN-SP} | <i>SN</i> | <i>SP</i> | Δ_{SN-SP} |
| hERG | 0.609±0.013 | 0.608±0.020 | 0.001 | 0.621±0.021 | 0.651±0.024 | -0.030 |
| AhR | 0.787±0.008 | 0.747±0.007 | 0.040 | 0.777±0.009 | 0.782±0.006 | -0.005 |
| ER | 0.698±0.020 | 0.719±0.019 | -0.021 | 0.727±0.020 | 0.736±0.024 | -0.009 |
| SRC-1 | 0.682±0.028 | 0.681±0.025 | 0.001 | 0.729±0.025 | 0.728±0.026 | 0.001 |
| THR | 0.636±0.031 | 0.627±0.025 | 0.009 | 0.665±0.011 | 0.673±0.013 | -0.008 |
| KCNQ2 | 0.686±0.012 | 0.686±0.011 | 0.000 | 0.727±0.021 | 0.744±0.013 | -0.017 |
| M1 | 0.653±0.028 | 0.650±0.028 | 0.003 | 0.666±0.028 | 0.674±0.028 | -0.008 |

Table A.15: Mean sensitivity and specificity values including standard deviations over 100 hold-out runs^a. Δ_{SN-SP} is the difference of sensitivity and specificity. $\mathcal{S} = \mathcal{S}_{ALL}$.

^a Please note that standard deviations only quantify the scatter among the values and do not allow for any conclusions on the statistical significance of the difference of the means [24].

| Dataset | Random Forests | | | |
|---------|--------------------|-------------|------------------|-----------|
| | \mathcal{R}^{db} | | Δ_{SN-SP} | |
| | <i>SN</i> | <i>SP</i> | <i>SN</i> | <i>SP</i> |
| hERG | 0.624±0.019 | 0.643±0.024 | -0.019 | |
| AhR | **± ** | **± ** | ** | |
| ER | 0.715±0.018 | 0.731±0.023 | -0.016 | |
| SRC-1 | 0.721±0.031 | 0.731±0.027 | -0.010 | |
| THR | 0.641±0.029 | 0.633±0.029 | 0.008 | |
| KCNQ2 | 0.738±0.015 | 0.717±0.012 | 0.021 | |
| M1 | 0.660±0.032 | 0.669±0.030 | -0.009 | |

Table A.16: Mean sensitivity and specificity values including standard deviations over 100 hold-out runs^a. Δ_{SN-SP} is the difference of sensitivity and specificity. $\mathcal{S} = \mathcal{S}_{ALL}$.

^a Please note that standard deviations only quantify the scatter among the values and do not allow for any conclusions on the statistical significance of the difference of the means [24].

A.3 Diversity measures

The used measures of classifier diversity are based on the cross-classification table (see A.17) and defined as listed in Kuncheva and Whitaker [76]:

| | | BBRC | | |
|---------------------|-----------|-----------|-----------|---------------------|
| | | correct | incorrect | |
| \mathcal{R}_{ACT} | correct | a | b | $(a + b)$ |
| | incorrect | c | d | $(c + d)$ |
| | | $(a + c)$ | $(b + d)$ | $n = a + b + c + d$ |

Table A.17: Cross-classification table

Yule's Q :

$$Q = \frac{ad - bc}{ad + bc} \quad (\text{A.1})$$

Correlation Coefficient ρ :

$$\rho = \frac{ad - bc}{\sqrt{(a + c)(b + d)(a + b)(c + d)}} \quad (\text{A.2})$$

The double-fault measure DF :

$$DF = \frac{d}{n} \quad (\text{A.3})$$

APPENDIX B

Additional Material for Chapter 5

| Rank | CID | Docking Score | Rank _{MCS} | $\Delta_{Rank_{MCS}}$ |
|------|--------------|---------------|---------------------|-----------------------|
| 1 | 60823 | -10.564 | 2 | -1 |
| 2 | ZINC02336737 | -5.808526 | 13 | -11 |
| 3 | ZINC00026851 | -5.699634 | 19 | -16 |
| 4 | ZINC00588719 | -5.568737 | 11 | -7 |
| 5 | ZINC00599752 | -5.46502 | 5 | 0 |
| 6 | ZINC00588053 | -5.463745 | 16 | -10 |
| 7 | ZINC00864379 | -5.291673 | 15 | -8 |
| 8 | ZINC01253780 | -5.211104 | 14 | -6 |
| 9 | ZINC00714466 | -5.149133 | 9 | 0 |
| 10 | ZINC00588723 | -5.14689 | 4 | 6 |
| 11 | ZINC00590911 | -5.135349 | 25 | -14 |
| 12 | ZINC04128931 | -5.101469 | 22 | -10 |
| 13 | ZINC01032240 | -5.094384 | 8 | 5 |
| 14 | ZINC00658975 | -5.038167 | 20 | -6 |
| 15 | ZINC00590434 | -4.973652 | 21 | -6 |
| 16 | ZINC00625939 | -4.972097 | 18 | -2 |
| 17 | ZINC01112466 | -4.918515 | 10 | 7 |
| 18 | ZINC04628438 | -4.916212 | 7 | 11 |
| 19 | ZINC02049068 | -4.914307 | 17 | 2 |
| 20 | ZINC00803728 | -4.669317 | 24 | -4 |
| 21 | ZINC03273040 | -4.569581 | 23 | -2 |
| 22 | 24848419 | -4.425088 | 3 | 19 |
| 23 | ZINC03837410 | -4.29318 | 12 | 11 |
| 24 | ZINC00588941 | -4.144152 | 26 | -2 |
| 25 | ZINC02129514 | -4.095075 | 6 | 19 |

Table B.1: Results of the first docking run. $\Delta_{Rank} = Rank_{docking} - Rank_{MCS}$. A negative Δ_{Rank} value means, in the MCS similarity the compound is ranked lower, a positive Δ_{Rank} that it is ranked higher than by the docking procedure.

| Rank | CID | Score | Rank _{MCS_{ext}} | $\Delta_{Rank_{MCS_{ext}}}$ |
|------|--------------|------------|-----------------------------------|-----------------------------|
| 1 | ZINC00588723 | -10.382184 | 16 | -15 |
| 2 | 24848419 | -7.980885 | 3 | -1 |
| 3 | ZINC01253780 | -7.385909 | 9 | -6 |
| 4 | ZINC00625939 | -7.157018 | 11 | -7 |
| 5 | ZINC01032240 | -7.104563 | 5 | 0 |
| 6 | ZINC00864379 | -7.052449 | 10 | -4 |
| 7 | ZINC00026851 | -6.910078 | 19 | -12 |
| 8 | ZINC00714466 | -6.702119 | 6 | 2 |
| 9 | ZINC01112466 | -6.667553 | 7 | 2 |
| 10 | 64715 | -6.654007 | 12 | -2 |
| 11 | ZINC02336737 | -6.537559 | 8 | 3 |
| 12 | ZINC00590911 | -6.45151 | 21 | -9 |
| 13 | 60823 | -6.29428 | 2 | 11 |
| 14 | ZINC03431465 | -6.289821 | 26 | -12 |
| 15 | ZINC00599752 | -6.09275 | 4 | 11 |
| 16 | ZINC00588053 | -5.887202 | 22 | -6 |
| 17 | ZINC04259960 | -5.79234 | 20 | -3 |
| 18 | ZINC02563245 | -5.748378 | 24 | -6 |
| 19 | 53232 | -5.683409 | 13 | 6 |
| 20 | ZINC03202042 | -5.606497 | 15 | 5 |
| 21 | ZINC04597014 | -5.130039 | 23 | -2 |
| 22 | ZINC03639638 | -5.095658 | 17 | 5 |
| 23 | ZINC03671410 | -4.205335 | 18 | 5 |
| 24 | 54454 | -3.865563 | 14 | 10 |
| 25 | ZINC02129514 | -3.838221 | 25 | 0 |

Table B.2: Results of the second docking run. $\Delta_{Rank} = Rank_{docking} - Rank_{MCS_{orECFP}}$. A negative Δ_{Rank} value means, in the extended similarity the compound is ranked lower, a positive Δ_{Rank} that it is ranked higher than by the docking procedure.

| DuD set | MCS | | | MCS _{ext} | | |
|---------------|-----------|----------|---------|--------------------|------------------|------------------|
| | 1% | 5% | 10% | 1% | 5% | 10% |
| HMGR | 8.5± 4.5 | 7.0±6.8 | 2.8±3.6 | 9.1± 2.5 | 6.5±6.1 | 2.0± 2.8 |
| ER | 13.6± 7.6 | 12.6±4.1 | 5.3±1.7 | 10.2± 6.5 | 10.6±2.5 | 5.0± 1.0 |
| PPAR γ | 4.6±10.6 | 1.2±5.4 | 1.7±2.8 | 4.5±11.0 | 3.8±5.5 | 1.6± 2.9 |
| P38 MAP | 9.6± 7.9 | 8.6±3.7 | 3.3±1.8 | 7.1± 6.8 | 7.3±3.7 | 2.7± 2.0 |
| TK | 20.1± 4.4 | 12.6±2.1 | 5.1±1.6 | 19.7± 5.3 | 14.0±2.1 | 5.5± 1.5 |
| FXa | 4.6±11.2 | 7.6±3.8 | 3.3±1.8 | 3.5±11.2 | 6.2±4.7 | 2.5± 2.6 |
| ADA | 10.1± 6.4 | 8.2±3.0 | 4.3±3.6 | 12.8± 6.4 | 8.8±3.3 | 6.1± 4.6 |
| DHFR | 10.9±10.6 | 11.7±2.9 | 4.7±1.1 | 3.4± 7.0 | 2.4±2.1 | 0.1± 0.1 |
| AChE | 10.3±12.5 | 11.3±4.7 | 4.8±2.5 | 10.1±11.9 | 10.4±5.1 | 4.4± 3.0 |
| COX-2 | 12.3± 9.2 | 11.7±2.2 | 5.3±1.1 | 11.4±10.3 | 10.5±3.7 | 2.5± 2.5 |
| w/d/l | | | | 8 / 0 / 2 | 7 / 0 / 3 | 8 / 0 / 2 |

Table B.3: Mean Δ_{EF} and standard deviations for the MCS and MCS_{ext} similarity methods at 1%, 5% and 10% of the database (receptor specific decoy set DuD_{set}). The extension fingerprint is calculated from 10% (20% for HMGR, TK and ADA) of the ligands (approach B2). Improvements of MCS_{ext} compared to MCS are marked with bold print. w/d/l = wins/draws/losses.

| DuD set | ECFP | | | ECFP _{ext} | | |
|---------------|-----------|-----------|----------|---------------------|------------------|------------------|
| | 1% | 5% | 10% | 1% | 5% | 10% |
| HMGR | 8.7± 9.4 | 6.8± 8.5 | 4.2±5.6 | 4.3± 9.5 | 6.5±4.8 | 2.9± 2.6 |
| ER | 8.0± 4.0 | 7.4± 3.9 | 6.7±4.6 | 6.8± 7.8 | 9.6±2.5 | 4.4± 1.4 |
| PPAR γ | 1.3± 0.7 | 7.1±11.2 | 1.0±0.7 | 4.2±11.0 | 3.6±5.6 | 1.8± 2.8 |
| P38 MAP | 7.0± 5.9 | 5.9± 3.0 | 3.4±2.0 | 4.0± 6.0 | 7.0±3.6 | 3.0± 1.9 |
| TK | 9.8± 6.0 | 12.1± 8.9 | 10.9±6.4 | 18.8± 7.3 | 11.8±3.8 | 4.9± 2.0 |
| FXa | 7.4±11.3 | 2.4± 2.0 | 1.7±1.5 | 3.5±11.2 | 4.5±5.3 | 2.0± 2.7 |
| ADA | 6.3± 3.3 | 6.4± 4.5 | 8.9±6.0 | 8.3± 7.1 | 9.3±2.0 | 4.4± 1.1 |
| DHFR | 2.5± 2.0 | 1.8± 1.5 | 1.8±1.5 | 5.7± 5.5 | 0.5±0.8 | 0.0± 0.0 |
| AChE | 15.0±11.2 | 5.2± 2.3 | 6.8±3.8 | 12.2±12.7 | 10.0±5.4 | 4.6± 2.9 |
| COX-2 | 8.7±10.6 | 3.4± 1.9 | 3.4±2.5 | 6.8±10.2 | 5.4±4.9 | 2.0± 2.7 |
| w/d/l | | | | 6 / 0 / 4 | 4 / 0 / 6 | 8 / 0 / 2 |

Table B.4: Mean Δ_{EF} and standard deviations for the ECFP and ECFP_{ext} similarity methods at 1%, 5% and 10% of the database (receptor specific decoy set DuD_{set}). The extension fingerprint is calculated from 10% (20% for HMGR, TK and ADA) of the ligands (approach B2). Improvements of ECFP_{ext} compared to ECFP are marked with bold print. w/d/l = wins/draws/losses.

| DuD set | bind_fp | | |
|---------------|-----------|-----------|-----------------------|
| | 1% | 5% | 10% |
| HMGR | 36.5± 2.0 | 20.2± 3.2 | 10.0± 2.2 |
| ER | 36.8±11.2 | 20.2± 6.0 | 10.0± 3.9 |
| PPAR γ | 34.3± 5.4 | 19.6± 4.1 | 6.8± 2.3 |
| P38 MAP | 20.0± 0.0 | 17.3± 6.8 | 8.6± 1.4 |
| TK | 36.6±13.7 | 20.2±12.0 | 10.1±4.2 [•] |
| FXa | 9.9± 7.6 | 8.5± 0.0 | 4.2± 0.8 |
| ADA | 36.7± 8.5 | 20.1± 0.0 | 10.0± 0.9 |
| DHFR | 36.5±16.8 | 20.0± 7.9 | 10.0± 1.0 |
| AChE | 36.4± 6.9 | 20.1± 9.2 | 10.0± 3.2 |
| COX-2 | 35.7± 7.5 | 19.8± 6.2 | 9.9± 1.0 |

Table B.5: Mean Δ_{EF} and standard deviation for the bind_fp similarity method at 1%, 5% and 10% of the database (receptor specific decoy set DuD_{set}). The extension fingerprint is calculated from 10% (20% for HMGR, TK and ADA) of the ligands (approach B2). Cases where bind_fp is better than ECFP or MCS are marked with a [•] or [°], respectively.

| Data | ligands | decoys | ligands + decoys |
|---------|---------|--------|------------------|
| HMGR | 0.644 | 0.304 | 0.210 |
| ER | 0.415 | 0.311 | 0.290 |
| PPAR | 0.571 | 0.351 | 0.350 |
| P38 MAP | 0.417 | 0.242 | 0.262 |
| TK | 0.494 | 0.217 | 0.248 |
| FXa | 0.492 | 0.299 | 0.293 |
| ADA | 0.389 | 0.206 | 0.188 |
| DHFR | 0.437 | 0.255 | 0.249 |
| AChE | 0.372 | 0.265 | 0.245 |
| COX-2 | 0.328 | 0.115 | 0.246 |

Table B.6: Intra set Tanimoto fingerprint similarities for the DuD ligand and decoy sets.

| CAC | MCS | | | MCS _{ext} | | |
|-------|-----------|----------|---------|--------------------|-------------------|-------------------|
| | 1% | 5% | 10% | 1% | 5% | 10% |
| 4 | 50.0±23.1 | 5.8±5.5 | 2.2±2.6 | 40.0±11.1 | 4.0± 3.2 | 0.0± 1.5 |
| 9 | 66.7±14.3 | 13.3±2.9 | 6.5±1.4 | 59.3±17.2 | 11.1± 5.0 | 5.6± 2.8 |
| 10 | 66.0±16.5 | 13.2±3.3 | 6.2±1.6 | 53.3± 9.8 | 6.7± 1.5 | 2.7± 0.9 |
| 21 | 80.0± 8.2 | 15.4±2.3 | 7.4±1.4 | 80.0±15.8 | 16.0± 2.6 | 8.0± 0.0 |
| 35 | 75.0± 9.6 | 14.2±1.7 | 7.1±0.8 | 72.2±11.7 | 12.2± 4.8 | 1.1± 3.2 |
| 44 | 66.4±25.8 | 9.3±6.3 | 3.3±3.2 | 30.1±21.1 | 6.0± 3.5 | 3.0± 1.7 |
| 52 | 71.4± 9.8 | 13.0±2.1 | 5.6±0.9 | 70.0± 9.7 | 4.0± 0.8 | 0.0± 1.0 |
| 54 | 81.4±10.7 | 15.4±2.5 | 7.4±1.5 | 70.0±16.6 | 12.1± 0.0 | 0.0± 0.0 |
| 57 | 54.0±23.2 | 9.2±5.7 | 4.4±2.8 | 60.0±13.5 | 3.9± 6.9 | 4.9± 1.1 |
| 81 | 82.0± 7.9 | 15.4±1.6 | 6.7±0.9 | 80.0± 6.7 | 9.8± 1.5 | 2.0± 2.6 |
| 86 | 67.0±18.3 | 10.2±5.4 | 4.0±3.0 | 50.0±17.9 | 4.0± 0.7 | 5.1± 1.4 |
| 98 | 80.0±10.0 | 14.7±2.4 | 6.7±1.5 | 70.0±14.6 | 6.1± 4.9 | 3.0± 1.8 |
| 105 | 72.5±14.9 | 13.3±4.3 | 6.6±2.1 | 88.7±10.1 | 10.0± 2.5 | 2.9± 2.6 |
| 113 | 71.1± 7.8 | 12.9±1.1 | 5.9±0.6 | 70.1± 6.9 | 8.4± 3.1 | 1.0± 1.4 |
| 121 | 74.0± 5.2 | 14.8±1.0 | 7.2±0.6 | 69.9± 4.9 | 13.9± 1.2 | 6.9± 1.4 |
| 129 | 65.0±10.0 | 10.5±1.0 | 4.8±1.3 | 50.0± 9.9 | 2.0± 0.4 | 0.9± 1.6 |
| 152 | 80.0±12.2 | 16.0±2.4 | 8.0±1.2 | 76.5±17.6 | 16.0± 0.6 | 8.0± 2.6 |
| 181 | 66.0± 5.5 | 9.6±2.6 | 2.8±0.8 | 60.0± 7.6 | 4.0± 4.7 | 0.8± 1.6 |
| 186 | 80.0± 7.1 | 14.0±2.8 | 6.0±1.6 | 20.0±15.6 | 2.0± 3.9 | 0.0± 1.8 |
| 195 | 77.8±11.1 | 14.7±2.5 | 5.8±0.9 | 62.9± 4.5 | 9.6± 3.6 | 4.8± 2.6 |
| 211 | 50.0± 0.0 | 10.0±0.0 | 5.0±0.0 | 50.0±18.0 | 0.0± 3.7 | 0.0± 1.6 |
| 213 | 77.8±13.6 | 15.1±2.4 | 7.1±1.5 | 74.1± 6.3 | 13.3± 0.8 | 6.7± 2.6 |
| 230 | 64.0±19.5 | 8.4±6.1 | 3.4±2.9 | 90.0±20.1 | 14.0± 1.6 | 0.9± 1.4 |
| 234 | 52.0±16.4 | 10.4±3.3 | 5.2±1.6 | 40.0±15.5 | 6.2± 2.7 | 1.0± 1.6 |
| 238 | 66.0±15.2 | 10.4±4.3 | 4.8±2.0 | 70.0±12.0 | 14.0± 4.9 | 3.1± 1.8 |
| w/d/l | | | | 19 / 2 / 4 | 21 / 1 / 3 | 21 / 1 / 3 |

Table B.7: Mean Δ_{EF} and standard deviation for the MCS and MCS_{ext} similarity methods at 1%, 5% and 10% of the database (ZINC subset). The extension fingerprint is calculated from 10% of the ligands (approach B2). Improvements of MCS_{ext} compared to MCS are marked in bold print. CAC = ChEMBL activity class. w/d/l = wins/draws/losses.

| CAC | ECFP | | | ECFP _{ext} | | |
|-------|-----------|----------|---------|---------------------|-------------------|-------------------|
| | 1% | 5% | 10% | 1% | 5% | 10% |
| 4 | 46.0±17.1 | 6.2±3.7 | 1.6±1.3 | 30.0± 9.9 | 4.0± 4.9 | 0.0± 0.0 |
| 9 | 65.4±13.0 | 12.1±2.5 | 5.3±1.6 | 59.3± 9.7 | 11.1± 4.7 | 5.2± 3.2 |
| 10 | 66.0±11.7 | 11.0±3.3 | 4.9±1.5 | 46.7±10.3 | 6.1± 3.9 | 1.9± 1.3 |
| 21 | 66.0±17.8 | 12.2±3.7 | 5.8±1.8 | 49.9±17.8 | 8.0± 2.5 | 3.0± 0.0 |
| 35 | 44.4±25.1 | 6.7±6.3 | 2.8±2.7 | 66.7±11.6 | 5.6± 2.3 | 0.0± 1.3 |
| 44 | 70.0±12.5 | 12.8±3.3 | 5.7±1.7 | 39.4±14.3 | 8.0± 2.5 | 4.1± 1.8 |
| 52 | 71.0±11.0 | 11.0±2.4 | 4.4±1.1 | 39.6±16.5 | 0.0± 3.3 | 0.0± 0.8 |
| 54 | 74.0±11.7 | 12.4±4.1 | 5.3±1.9 | 60.0±10.5 | 7.6± 1.6 | 1.9± 1.5 |
| 57 | 59.0±17.3 | 10.0±3.9 | 3.9±2.1 | 50.0±12.5 | 2.0± 0.9 | 0.8± 2.2 |
| 81 | 77.0± 6.7 | 14.2±1.5 | 6.5±1.0 | 80.0±11.0 | 12.0± 2.6 | 3.6± 1.5 |
| 86 | 55.0±17.2 | 7.4±4.4 | 2.6±2.0 | 70.0±11.7 | 6.0± 4.8 | 1.0± 1.8 |
| 98 | 60.0±22.1 | 11.2±4.3 | 5.5±2.3 | 40.0±25.8 | 6.0± 3.3 | 3.0± 1.0 |
| 105 | 58.0±24.9 | 10.8±4.6 | 5.2±2.2 | 60.0± 9.8 | 10.0± 2.4 | 4.0± 0.0 |
| 113 | 64.0±10.8 | 10.0±3.4 | 4.2±1.8 | 50.0± 7.9 | 2.1± 2.6 | 0.9± 2.6 |
| 121 | 74.0± 5.2 | 14.6±1.0 | 6.5±0.8 | 70.0±10.7 | 14.0± 2.4 | 6.8± 1.6 |
| 129 | 69.0± 9.9 | 12.4±1.6 | 5.5±1.5 | 49.8±23.2 | 6.0± 4.3 | 0.8± 1.3 |
| 152 | 74.0±12.6 | 14.6±2.5 | 6.9±1.5 | 90.0±13.3 | 14.2± 0.7 | 4.2± 1.7 |
| 181 | 61.0±12.9 | 10.6±3.0 | 4.7±1.6 | 39.8±24.9 | 7.8± 4.9 | 3.0± 1.9 |
| 186 | 60.0±14.9 | 7.4±4.6 | 2.6±1.7 | 20.1±13.6 | 2.0± 2.5 | 0.0± 2.1 |
| 195 | 69.1±14.5 | 12.3±2.7 | 5.7±1.4 | 59.3±19.5 | 11.1± 3.3 | 4.8± 1.8 |
| 211 | 42.0± 9.2 | 7.8±2.9 | 3.1±1.4 | 19.8±12.6 | 0.0± 1.2 | 0.0± 2.6 |
| 213 | 61.7±13.7 | 9.4±2.2 | 3.7±1.1 | 55.6±20.2 | 10.4± 0.4 | 5.2± 1.4 |
| 230 | 60.0±18.3 | 10.2±3.6 | 3.3±1.3 | 90.0±17.8 | 11.9± 3.6 | 2.0± 1.4 |
| 234 | 57.0±17.0 | 8.4±3.9 | 3.2±1.7 | 41.2±14.9 | 2.0± 4.7 | 1.1± 1.6 |
| 238 | 64.0±17.8 | 11.8±3.9 | 5.0±1.9 | 80.0±14.5 | 16.3± 1.6 | 7.0± 1.8 |
| w/d/l | | | | 18 / 0 / 7 | 22 / 0 / 3 | 22 / 0 / 3 |

Table B.8: Mean Δ_{EF} and standard deviation for the ECFP and ECFP_{ext} similarity methods at 1%, 5% and 10% of the database (ZINC subset). The extension fingerprint is calculated from 10% of the ligands (approach B2). Improvements of MCS_{ext} compared to MCS are marked in bold print. CAC = ChEMBL activity class. w/d/l = wins/draws/losses.

| | ECFP _{ext} | | | | | | MCS _{ext} | | | | | |
|-------------------|---------------------|------|-----|------|-----|------|--------------------|------|-----|------|-----|------|
| | 1% | | 5% | | 10% | | 1% | | 5% | | 10% | |
| | win | loss | win | loss | win | loss | win | loss | win | loss | win | loss |
| HMGR | 10 | 0 | 10 | 0 | 10 | 0 | 5 | 0 | 5 | 0 | 5 | 0 |
| ER | 8 | 2 | 10 | 0 | 9 | 1 | 10 | 0 | 10 | 0 | 10 | 0 |
| PPAR _γ | 9 | 1 | 9 | 1 | 9 | 1 | 9 | 1 | 10 | 0 | 9 | 1 |
| P38 MAP | 9 | 1 | 10 | 0 | 10 | 0 | 10 | 0 | 10 | 0 | 9 | 1 |
| TK | 10 | 0 | 9 | 1 | 8 | 2 | 9 | 1 | 9 | 1 | 9 | 1 |
| FXa | 10 | 0 | 9 | 1 | 9 | 1 | 10 | 0 | 9 | 1 | 9 | 1 |
| ADA | 10 | 0 | 6 | 4 | 10 | 0 | 8 | 1 | 8 | 1 | 8 | 1 |
| DHFR | 8 | 2 | 10 | 0 | 10 | 0 | 10 | 0 | 10 | 0 | 10 | 0 |
| ACHE | 9 | 1 | 9 | 1 | 9 | 1 | 10 | 0 | 10 | 0 | 9 | 1 |
| COX-2 | 6 | 4 | 7 | 3 | 8 | 2 | 7 | 3 | 9 | 1 | 9 | 1 |
| sum | 89 | 11 | 89 | 11 | 92 | 8 | 88 | 6 | 90 | 4 | 87 | 7 |

Table B.9: Win/Loss counts for all ten random folds for extended similarities MCS_{ext} and ECFP_{ext} versus their base similarities MCS and ECFP for the receptor specific decoy sets DuD_{set} at 1%, 5% and 10% of the database. The extension fingerprint is calculated from all ligands (approach B1).

| | ECFP _{ext} | | | | | | MCS _{ext} | | | | | |
|-------------------|---------------------|------|-----|------|-----|------|--------------------|------|-----|------|-----|------|
| | 1% | | 5% | | 10% | | 1% | | 5% | | 10% | |
| | win | loss | win | loss | win | loss | win | loss | win | loss | win | loss |
| HMGR | 102 | 8 | 110 | 0 | 110 | 0 | 48 | 7 | 51 | 4 | 55 | 0 |
| ER | 70 | 40 | 98 | 12 | 99 | 11 | 97 | 13 | 109 | 1 | 103 | 7 |
| PPAR _γ | 93 | 17 | 78 | 32 | 86 | 24 | 94 | 16 | 90 | 20 | 77 | 33 |
| P38 MAP | 102 | 8 | 104 | 6 | 104 | 6 | 103 | 7 | 105 | 5 | 100 | 10 |
| TK | 96 | 14 | 99 | 11 | 92 | 18 | 96 | 14 | 97 | 13 | 99 | 11 |
| FXa | 101 | 9 | 100 | 10 | 94 | 16 | 104 | 6 | 100 | 10 | 100 | 10 |
| ADA | 91 | 19 | 89 | 21 | 109 | 1 | 81 | 18 | 96 | 3 | 84 | 15 |
| DHFR | 74 | 36 | 110 | 0 | 110 | 0 | 104 | 6 | 110 | 0 | 110 | 0 |
| ACHE | 93 | 17 | 100 | 10 | 101 | 9 | 96 | 14 | 105 | 5 | 101 | 9 |
| COX-2 | 57 | 53 | 60 | 50 | 90 | 20 | 78 | 32 | 86 | 24 | 100 | 10 |
| sum | 879 | 221 | 948 | 152 | 995 | 105 | 901 | 133 | 949 | 85 | 929 | 105 |

Table B.10: Win/Loss counts for all 110 random folds (10 repetitions * 11 os) for extended similarities MCS_{ext} and ECFP_{ext} versus their base similarities MCS and ECFP for the receptor specific decoy sets DuD_{set} at 1%, 5% and 10% of the database. The extension fingerprint is calculated from all ligands (approach B1).

| | ECFP _{ext} | | | | | | MCS _{ext} | | | | | |
|---------------|---------------------|------|-----|------|-----|------|--------------------|------|-----|------|-----|------|
| | 1% | | 5% | | 10% | | 1% | | 5% | | 10% | |
| | win | loss | win | loss | win | loss | win | loss | win | loss | win | loss |
| HMGR | 102 | 8 | 88 | 22 | 98 | 12 | 48 | 7 | 49 | 6 | 55 | 0 |
| ER | 70 | 40 | 74 | 36 | 74 | 36 | 87 | 23 | 89 | 21 | 67 | 43 |
| PPAR γ | 93 | 17 | 86 | 24 | 88 | 22 | 69 | 41 | 95 | 15 | 72 | 38 |
| P38 MAP | 93 | 17 | 96 | 14 | 103 | 7 | 93 | 17 | 100 | 10 | 110 | 0 |
| TK | 75 | 35 | 95 | 15 | 94 | 16 | 96 | 14 | 85 | 25 | 82 | 28 |
| FXa | 101 | 9 | 67 | 43 | 80 | 30 | 99 | 11 | 84 | 26 | 83 | 27 |
| ADA | 86 | 24 | 79 | 31 | 98 | 12 | 60 | 39 | 88 | 11 | 76 | 23 |
| DHFR | 75 | 35 | 107 | 3 | 110 | 0 | 100 | 10 | 110 | 0 | 110 | 0 |
| ACHE | 86 | 24 | 95 | 15 | 96 | 14 | 87 | 23 | 98 | 12 | 94 | 16 |
| COX-2 | 69 | 41 | 76 | 34 | 78 | 32 | 65 | 45 | 86 | 24 | 90 | 20 |
| sum | 850 | 250 | 863 | 237 | 919 | 181 | 804 | 230 | 884 | 150 | 839 | 195 |

Table B.11: Win/Loss counts for all 110 random folds (10 repetitions * 11 α s) for extended similarities MCS_{ext} and ECFP_{ext} versus their base similarities MCS and ECFP for the receptor specific decoy sets DuD_{set} at 1%, 5% and 10% of the database. The extension fingerprint is calculated from 10% (20% for HMGR, TK and ADA) of the ligands (approach B2).

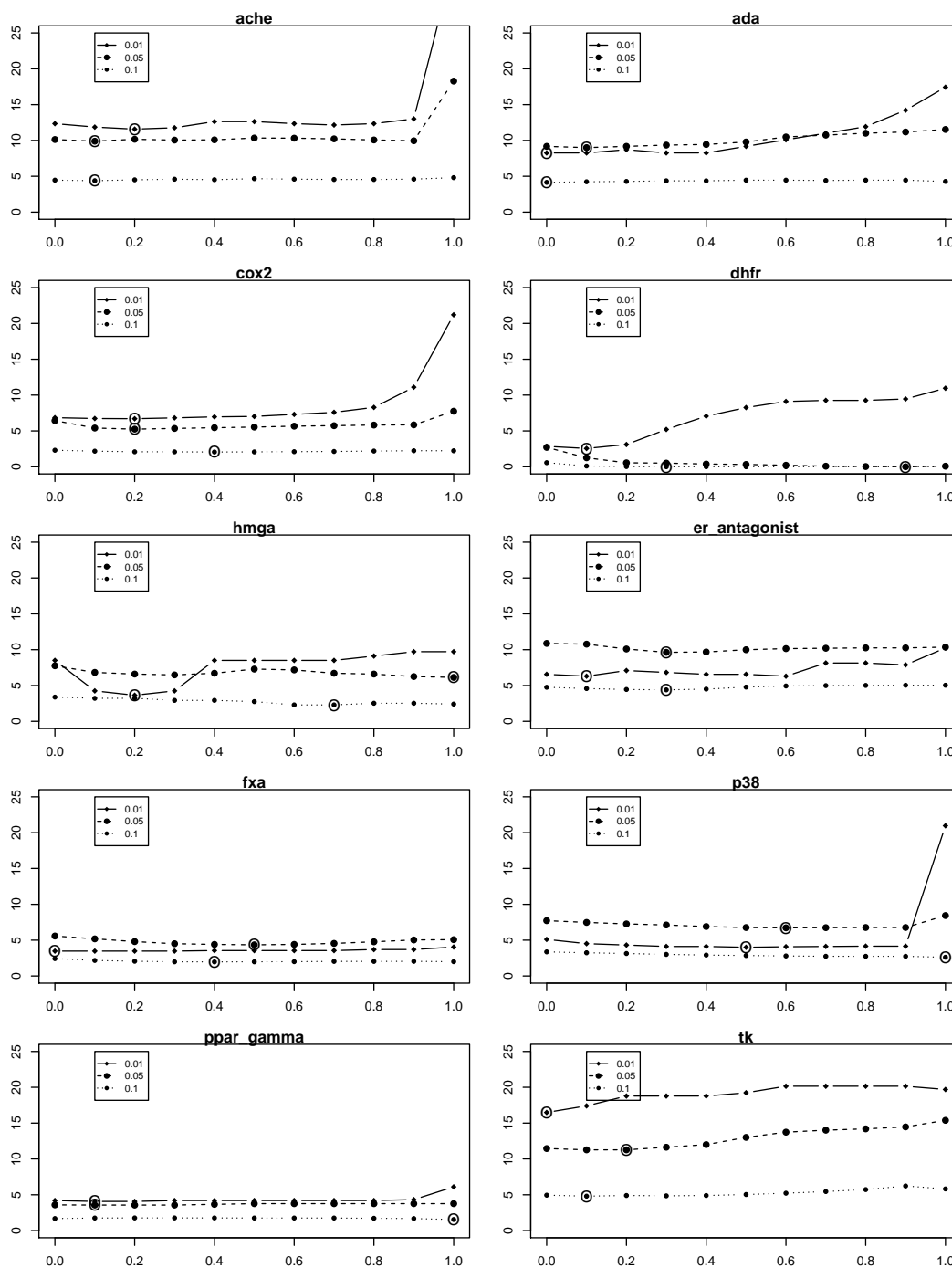


Figure B.1: Plot of α vs. Mean Δ_{EF} for $ECFP_{ext}$. On the x-axis the values of the combining factor α is plotted versus the mean Δ_{EF} for $ECFP_{ext}$ on the y-axis. (approach B2)

APPENDIX C

Additional Material for Chapter 8

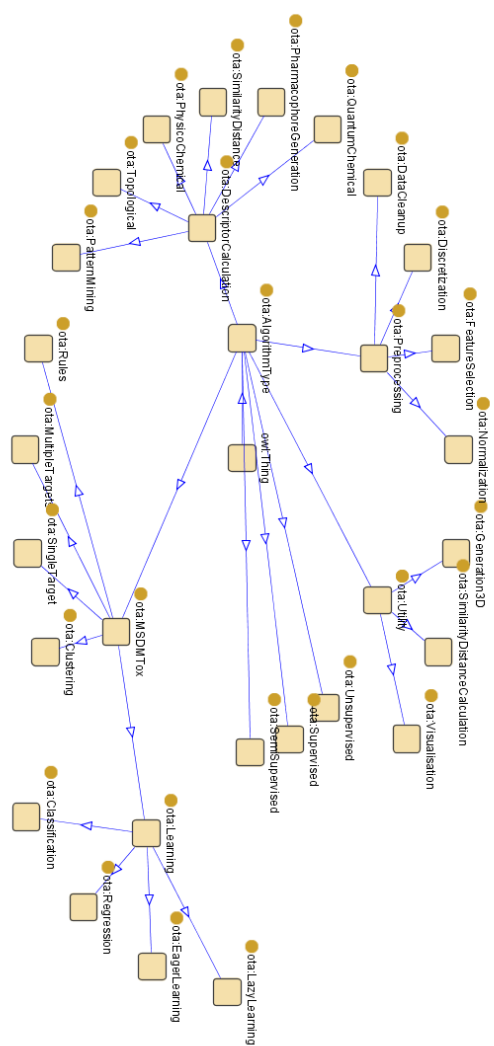
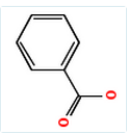
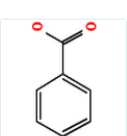


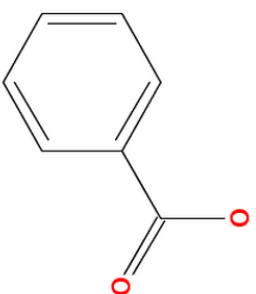
Figure C.1: Algorithm Type Ontology developed in the OpenTox project. Existing ontologies, e.g. for physico-chemical molecular descriptors, are also used in the project.

| Description | Method | URI | Parameters | Result | Status codes |
|---|--------|-----------------|---|---|--------------|
| Get URIs of all available algorithms | GET | /algorithm | [subjectid] [?sameas=URI-of-the-owl:sameAs-entry] | List of all algorithm URIs or RDF representation, or algorithms of specific types, if query parameter exists Returns all algorithms, for which owl:sameAs is given by the query | 200,404,503 |
| Get the ontology representation of an algorithm | GET | /algorithm/{id} | [subjectid] | Algorithm representation in one of the supported MIME types | 200,404,503 |
| Apply the algorithm | POST | /algorithm/{id} | dataset_uri prediction_feature parameter (specified by the algorithm provider), dataset_service=dataset_service_uri , result_dataset , [subjectid] | <i>model URI (Prefer to create algorithm services that return model URIs instead of datasets or features)</i> <i>dataset URI</i> <i>featureURI</i> Redirect to task URI for time consuming computations | 200,404,503 |

Figure C.2: OpenTox Algorithm API: Provides access to OpenTox algorithms; see: <http://www.opentox.org/dev/apis/api-1.2/Algorithm>

[Help](#)



[Download](#)
Browse all
[Start Read Across](#)

ToxPredict

WELCOME, GUEST
My account
Log out

PREDICT
Search structure
Upload structure
View results

BROWSE
Datasets
Models

MY WORKSPACE
My uploads

| Predictions | Datasets |
|-------------------------|--|
| CASRN | 65-85-0 |
| EINECS | 200-618-2 |
| IUPAC name | benzoic acid |
| Chemical Name | benzoic acid |
| SMILES | OC(=O)c1ccccc1 |
| Standard InChI | InChI=1S/C7H6O2/c8-7(9)6-4-2-1-3-5-6/h1-5H,(H,8,9) |
| Standard InChI key | WPYMKLBDIGXBTP-UHFFFAOYSA-N |
| REACH registration date | 30.11.2010 |

[Run All](#)

MolecularWeight Calculate

OSAR SRC KOWWIN fingerprints AD Calculate

Caco-2 Cell Permeability <http://www.ncbi.nlm.nih.gov/pubmed/16959190> Calculate

ToxTree: Structure Alerts for the in vivo micronucleus assay in rodents Calculate

OpenTox model created with TUM's KNN/regression model learning web service. Calculate

MLR model for Exp LogKow Calculate

<http://pentox.ntua.gr:8080/model/2b22fab-a9bf-4295-8172-1d18d4476da0> Calculate

<http://pentox.ntua.gr:8080/model/536ce048-b78b-4fb-3-8bf-1c85447bf41> Calculate

<http://pentox.ntua.gr:8080/model/6a3df28-b3f3-426d-9428-b7c31d9eb599> Calculate

OpenTox lazar model for ISSCAN Carcinogenicity (SAL) Calculate

OpenTox lazar model for Salmonella Mutagenicity(SAL) Calculate

OpenTox lazar model for Fish (aquatic) toxicity (LC50_mmol) Calculate

Physicochemical effects >> Dissociation constant (pKa)

pKa Calculate

pKa-SMARTS 3.52

Figure C.3: ToxPredict View results page for benzoic acid

List of Figures

| | | |
|-----|--|-----|
| 1.1 | Virtual screening schema | 5 |
| 2.1 | Compounds from the family of statins | 9 |
| 2.2 | Triangle illustrating the triangular inequality of a metric. | 10 |
| 2.3 | Molecular fingerprint example | 11 |
| 2.4 | Maximum common substructure example | 13 |
| 2.5 | LoMoGraph schematic overview. | 17 |
| 4.1 | Schematic depiction of similarity descriptor vector composition. | 37 |
| 4.2 | Scatter plot of running times. | 42 |
| 4.3 | Bar charts of RF and SVM results for single similarity descriptors. | 43 |
| 4.4 | Predictive accuracies for BBRC and ECFP vs. $SD(\mathcal{S}_{ALL}, \mathcal{R}^{act})$ | 44 |
| 4.5 | Predictive accuracies for $SD(\mathcal{S}_{ALL}, \mathcal{R}^{act})$ and the combination with BBRC and ECFP. | 45 |
| 5.1 | Overview of the experimental setup. | 53 |
| 5.2 | 2D depiction of six statin structures. | 57 |
| 5.3 | PPAR γ approved active drugs. | 57 |
| 5.4 | Docking illustrations. | 60 |
| 5.5 | Structures from MCS ranking | 61 |
| 5.6 | PPAR γ binding relevant substructures. | 63 |
| 5.7 | Plot of α vs. Mean Δ_{EF} | 69 |
| 6.1 | Graphical overview of the four strategies. | 79 |
| 6.2 | Learning curves for nearest neighbor | 81 |
| 6.3 | Learning curves for nearest neighbor. | 85 |
| 6.4 | Learning curves for nearest neighbor. | 86 |
| 6.5 | Graphical representation of the α_i and α_i^p | 86 |
| 6.6 | Learning curves for the data-driven selection. | 90 |
| 6.7 | Learning curves for the data-driven selection. | 92 |
| 7.1 | Approach synergies. | 95 |
| 8.1 | SOAP web service structure | 108 |

| | | |
|-----|---|-----|
| 8.2 | OpenTox API building blocks | 109 |
| 8.3 | OpenTox API learning blocks | 109 |
| 8.4 | OpenTox API prediction blocks | 109 |
| 8.5 | OpenTox API reporting blocks | 110 |
| 8.6 | OpenTox resource relationships | 111 |
| 8.7 | ToxPredict entry page | 114 |
| 9.1 | Interactions for PDB instance 1HWI | 117 |
| A.1 | Reference compounds for the hERG data. | 119 |
| A.2 | Reference compounds for the AhR data. | 121 |
| A.3 | Reference compounds for the ER data. | 121 |
| A.4 | Reference compounds for the PPAR γ data. | 121 |
| A.5 | Reference compounds for the THR data. | 122 |
| A.6 | Reference compounds for the KCNQ2 data. | 122 |
| A.7 | Reference compounds for the M1 data. | 123 |
| B.1 | Plot of α vs. Mean Δ_{EF} | 136 |
| C.1 | OpenTox Algorithm Type Ontology | 138 |
| C.2 | OpenTox Algorithm API | 139 |
| C.3 | ToxPredict <i>View results</i> page | 140 |

List of Tables

| | | |
|------|---|-----|
| 4.1 | Parameters of the structural clustering algorithm. | 39 |
| 4.2 | Summary of the used PubChem assay datasets. | 39 |
| 4.3 | Random Forest (RF) and SVM prediction accuracies. | 44 |
| 4.4 | $SD(\mathcal{S}_{ALL}, \mathcal{R}^{act})$ combined with structural descriptors. | 45 |
| 4.5 | Analysis of classifier diversity for the AhR and SRC-1 datasets. | 46 |
| 4.6 | Prediction accuracy results of the two stacking variants. | 47 |
| | | |
| 5.1 | Overview of the similarity extension steps. | 54 |
| 5.2 | PPAR γ market approved drugs | 58 |
| 5.3 | Overview of the used DuD datasets. | 58 |
| 5.4 | Results of the docking run (MCS top 25). | 63 |
| 5.5 | Results of the docking run (MCS $_{ext}$ top 25). | 63 |
| 5.6 | Δ_{EF} values for HMGR | 64 |
| 5.7 | Δ_{EF} values for PPAR γ | 64 |
| 5.8 | Mean Δ_{EF} and standard deviation for MCS and MCS $_{ext}$ (approach B1). | 65 |
| 5.9 | Mean Δ_{EF} and standard deviation for ECFP and ECFP $_{ext}$ (approach B1). | 66 |
| 5.10 | Mean Δ_{EF} and standard deviation for bind $_{fp}$ (approach B1). | 67 |
| 5.11 | Best α coefficients for MCS $_{ext}$ and ECFP $_{ext}$ (approach B2). | 68 |
| 5.12 | Mean Δ_{EF} and standard deviation using the best α coefficients (approach B1). | 68 |
| 5.13 | Mean Δ_{EF} and standard deviation using the best α coefficients (approach B2). | 70 |
| 5.14 | Win/Loss counts for ten random folds for extended similarites on DuD set. | 70 |
| | | |
| 6.1 | Overview of the datasets and the minimum support threshold <i>minsup</i> | 78 |
| 6.2 | Distance learning vs. Best single distance | 83 |
| 6.3 | Penalized adapted transfer vs. Distance learning | 84 |
| 6.4 | Datasets used for data-driven selection of source datasets. | 87 |
| 6.5 | Distance learning vs. Best single distance | 91 |
| 6.6 | Result comparison with TrAdaBoost.R2 for three datasets. | 91 |
| | | |
| 7.1 | Top 10 inhibitors in assay 631 and 639 | 98 |
| 7.2 | Δ_{EF} values for SimBoostedQSAR similarity rankings. | 98 |
| 7.3 | Summary of datasets. | 100 |

| | | |
|------|--|-----|
| 8.1 | OpenTox project partners | 105 |
| 8.2 | OECD principles | 107 |
| A.1 | SMILES representation of hERG reference compounds. | 119 |
| A.2 | SMILES representation of AhR reference compounds. | 120 |
| A.3 | SMILES representation of ER reference compounds. | 120 |
| A.4 | SMILES representation of PPAR γ reference compounds. | 120 |
| A.5 | SMILES representation of THR reference compounds. | 120 |
| A.6 | SMILES representation of KCNQ2 reference compounds. | 120 |
| A.7 | SMILES representation of M1 reference compounds. | 122 |
| A.8 | Example of CPU runtimes in seconds. | 124 |
| A.9 | RF ten-fold cross-validation results. | 124 |
| A.10 | SVM ten-fold cross-validation results. | 124 |
| A.11 | Significance analysis of single similarities and their combination. | 125 |
| A.12 | Significance analysis of BBRC and the variants for finding reference molecules. | 125 |
| A.13 | Significance analysis of ECFP and the variants for finding reference molecules. | 125 |
| A.14 | Mean sensitivity and specificity values for BBRC and ECFP. | 126 |
| A.15 | Mean sensitivity and specificity values for \mathcal{R}_{LIT} , \mathcal{R}_{ACT} | 126 |
| A.16 | Mean sensitivity and specificity values for \mathcal{R}_{DB} | 126 |
| A.17 | Cross-classification table | 127 |
| B.1 | Results of the first docking run. | 129 |
| B.2 | Results of the second docking run. | 130 |
| B.3 | Mean Δ_{EF} values and standard deviations for the MCS and MCS _{ext} similarity methods (approach B2). | 130 |
| B.4 | Mean Δ_{EF} values and standard deviations for the ECFP and ECFP _{ext} similarity methods (approach B2). | 131 |
| B.5 | Mean Δ_{EF} and standard deviation for the bind _{fp} similarity method (approach B2). | 131 |
| B.6 | DuD intra set Tanimoto fingerprint similarities | 131 |
| B.7 | Mean Δ_{EF} and standard deviation for the experiments on the ChEMBL activity classes (approach B2). | 132 |
| B.8 | Mean Δ_{EF} and standard deviation for the experiments on the ChEMBL activity classes (approach B2). | 133 |
| B.9 | Win/Loss counts for ten random folds for extended similarites on DuD set ($\alpha = 0.3$; approach B1). | 134 |
| B.10 | Win/Loss counts for all random folds for extended similarites on DuD set ($\alpha \in (0.0, 0.1)$; approach B1). | 134 |
| B.11 | Win/Loss counts for all random folds for extended similarites on DuD set ($\alpha \in (0.0, 0.1)$; approach B2). | 135 |

Bibliography

- [1] Special Issue: Computational Methods for Drug Repurposing. *Briefings in Bioinformatics* 12 (2011).
- [2] AJMANI, S., JADHAV, K., AND KULKARNI, S. Three-dimensional QSAR using the k-nearest neighbor method and its interpretation. *Journal of Chemical Information and Modeling* 46, 1 (2006), 24–31.
- [3] BALCAN, M.-F., AND BLUM, A. On a theory of learning with similarity functions. In *Proceedings of the International Conference on Machine Learning (ICML'06)* (New York, NY, USA, 2006), ICML '06, ACM, pp. 73–80.
- [4] BAR-HILLEL, A., HERTZ, T., SHENTAL, N., AND WEINSHALL, D. Learning distance functions using equivalence relations. In *Proceedings of the International Conference on Machine Learning (ICML'03)* (2003), T. Fawcett and N. Mishra, Eds., AAAI Press, pp. 11–18.
- [5] BAR-HILLEL, A., HERTZ, T., SHENTAL, N., AND WEINSHALL, D. Learning a Mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research* 6 (2005), 937–965.
- [6] BARNARD, J., AND DOWNS, G. Clustering of chemical structures on the basis of two-dimensional similarity measures. *Journal of Chemical Information and Computer Sciences* 32, 6 (1992), 644–649.
- [7] BENDER, A., AND GLEN, R. Molecular similarity: a key technique in molecular informatics. *Organic & Biomolecular Chemistry* 2, 22 (2004), 3204–3218.
- [8] BENIGNI, R., AND BOSSA, C. Predictivity of QSAR. *Journal of Chemical Information and Modeling* 48, 5 (2008), 971–980.
- [9] BENIGNI, R., BOSSA, C., AND VARI, M. Chemical carcinogens: Structures and experimental data, 2008. <http://www.iss.it/binary/ampp/cont/ISSCANv2aEn.1134647480.pdf>.
- [10] BERMAN, H., WESTBROOK, J., FENG, Z., GILLILAND, G., BHAT, T., WEISSIG, H., SHINDYALOV, I., AND BOURNE, P. The Protein Data Bank. *Nucleic Acids Research* 28 (2000), 235–242.

- [11] BOLTON, E., WANG, Y., THIESSEN, P., AND BRYANT, S. Pubchem: integrated platform of small molecules and biological activities. *Annual Reports in Computational Chemistry* 4 (2008), 217–241.
- [12] BREIMAN, L. Stacked regressions. *Machine Learning* 24, 1 (1996), 49–64.
- [13] BREIMAN, L. Random forests. *Machine Learning* 45, 1 (2001), 5–32.
- [14] BUCHWALD, F., GIRSCHICK, T., EIBE, F., AND KRAMER, S. Fast conditional density estimation for quantitative structure-activity relationships. In *Proceedings of the Conference on Artificial Intelligence (AAAI'10)* (2010), pp. 1268–1273.
- [15] BUCHWALD, F., GIRSCHICK, T., SEELAND, M., AND KRAMER, S. Using Local Models to Improve (Q) SAR Predictivity. *Molecular Informatics* 30, 2-3 (2011), 205–218.
- [16] CERONI, A., COSTA, F., AND FRASCONI, P. Classification of small molecules by two-and three-dimensional decomposition kernels. *Bioinformatics* 23, 16 (2007), 2038–2035.
- [17] CHEN, J., LINSTEAD, E., SWAMIDASS, S., WANG, D., AND BALDI, P. ChemDB update - full-text search and virtual chemical space. *Bioinformatics* 23, 17 (2007), 2348–2351.
- [18] CHEN, S., MA, B., AND ZHANG, K. On the similarity metric and the distance metric. *Theoretical Computer Science* 410, 24 (2009), 2365–2376.
- [19] CHEPELEV, L., AND DUMONTIER, M. Chemical entity semantic specification: Knowledge representation for efficient semantic cheminformatics and facile data integration. *Journal of Cheminformatics* 3, 1 (2011), 20.
- [20] CHEPELEV, L., AND DUMONTIER, M. Semantic Web integration of Cheminformatics resources with the SADI framework. *Journal of Cheminformatics* 3, 1 (2011), 16.
- [21] CLARK, D., AND PICKETT, S. Computational methods for the prediction of 'drug-likeness'. *Drug Discovery Today* 5, 2 (2000), 49–58.
- [22] CRIPPEN, G., AND HAVEL, T. *Distance geometry and molecular conformation*. Research Studies Press, Taunton, Somerset, England, 1988.
- [23] CUADRADO, M., RUIZ, I., AND GÓMEZ-NIETO, M. A steroids qsar approach based on approximate similarity measurements. *Journal of Chemical Information and Modeling* 46, 4 (2006), 1678–1686.
- [24] CUMMING, G., FIDLER, F., AND VAUX, D. Error bars in experimental biology. *Journal of Cell Biology* 177, 1 (2007), 7–11.

-
- [25] DAI, W., YANG, Q., XUE, G.-R., AND YU, Y. Boosting for transfer learning. In *Proceedings of the International Conference on Machine Learning (ICML'07)* (2007), ACM, pp. 193–200.
- [26] DE-BIE, T., MOMMA, M., AND CRISTIANINI, N. Efficiently learn the metric with side information. *Lecture Notes in Artificial Intelligence 2842* (2003), 175 – 189.
- [27] DELANO, W. The case for open-source software in drug discovery. *Drug Discovery Today 10* (2005), 213–217.
- [28] DEMSAR, J. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research 7* (Jan 2006), 1–30.
- [29] DIETTERICH, T., AND MICHALSKI, R. A comparative review of selected methods for learning from examples. In *Machine Learning: An Artificial Intelligence Approach*, R. Michalski, J. Carbonell, and T. Mitchell, Eds. 1983, pp. 41–81.
- [30] DUDLEY, J. T., DESHPANDE, T., AND BUTTE, A. Exploiting drug-disease relationships for computational drug repositioning. *Briefings in Bioinformatics 12*, 4 (2011), 303–311.
- [31] DURANT, J., LELAND, B., HENRY, D., AND NOURSE, J. Reoptimization of MDL keys for use in drug discovery. *Journal of Chemical Information and Computer Sciences 42*, 6 (2002), 1273–1280.
- [32] EATON, E., DESJARDINS, M., AND LANE, T. Modeling transfer relationships between learning tasks for improved inductive transfer. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD'08)* (2008), pp. 317–332.
- [33] EISENBERG, D. Cholesterol lowering in the management of coronary artery disease: The clinical implications of recent trials. *The American Journal of Medicine 104*, 2, Supplement 1 (1998), 2S–5S.
- [34] ENDO, A., KURODA, M., AND TANZAWA, K. Competitive inhibition of 3-hydroxy-3-methylglutaryl coenzyme A reductase by ML-236A and ML-236B fungal metabolites, having hypocholesterolemic activity. *FEBS letters 72*, 2 (1976), 323–326.
- [35] EVERS, A., AND KLABUNDE, T. Structure-based drug discovery using GPCR homology modeling: successful virtual screening for antagonists of the alpha1a adrenergic receptor. *Journal of Medicinal Chemistry 48*, 4 (2005), 1088–1097.
- [36] EVGENIOU, T., MICHELLI, C., AND PONTIL, M. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research 6* (2005), 615–637.
- [37] FIELDING, R. Architectural styles and the design of network-based software architectures, university of california, irvine, 2000.

- [38] FRANK, E., HALL, M., AND PFAHRINGER, B. Locally weighted naive Bayes. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI '03)* (Acapulco, Mexico, 2003), Morgan Kaufmann, pp. 249–256.
- [39] FRANK, M., CHEHREGHANI, M., AND BUHMANN, J. The minimum transfer cost principle for model-order selection. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD'11)*, vol. 6911 of *Lecture Notes in Computer Science*. 2011, pp. 423–438.
- [40] GAREY, M., AND JOHNSON, D. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.
- [41] GÄRTNER, T. *Kernels for structured data*, vol. 72. World Scientific Pub Co Inc, 2009.
- [42] GASTEIGER, J. *Chemoinformatics: a textbook*. Vch Verlagsgesellschaft MbH, 2003.
- [43] GELDENHUYS, W., GAASCH, K., WATSON, M., ALLEN, D., AND VAN DER SCHYF, C. Optimizing the use of open-source software applications in drug discovery. *Drug Discovery Today* 11, 3-4 (2006), 127–132.
- [44] GEPPERT, H., VOGT, M., AND BAJORATH, J. Current trends in ligand-based virtual screening: molecular representations, data mining methods, new application areas, and performance evaluation. *Journal of Chemical Information and Modeling* 50, 2 (2010), 205–216.
- [45] GIRSCHICK, T., BUCHWALD, F., HARDY, B., AND KRAMER, S. OpenTox: A Distributed REST Approach to Predictive Toxicology. In *Proceedings of the Workshop on Third Generation Data Mining: Towards Service-oriented Knowledge Discovery (SoKD'10) at ECML/PKDD'10* (2010), N. Lavrac, V. Podpecan, J. Kok, and M. Hilario, Eds., pp. 61–62.
- [46] GIUGNO, R., AND SHASHA, D. Graphgrep: A fast and universal method for querying graphs. In *Proceedings of the International Conference on Pattern Recognition (ICPR'02)* (2002), pp. 112–115.
- [47] GLOBERSON, A., AND ROWEIS, S. Metric learning by collapsing classes. In *Advances in Neural Information Processing Systems 18*, Y. Weiss, B. Schölkopf, and J. Platt, Eds. MIT Press, 2006, pp. 451–458.
- [48] GOLDBERGER, J., ROWEIS, S., HINTON, G., AND SALAKHUTDINOV, R. Neighborhood Component Analysis. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS'04)* (2005), pp. 513–520.

- [49] HALL, M., FRANK, E., HOLMES, G., PFAHRINGER, B., REUTEMANN, P., AND WITTEN, I. H. The WEKA data mining software: an update. *SIGKDD Explorations* 11, 1 (2009), 10–18.
- [50] HAMMETT, L. The effect of structure upon the reactions of organic compounds. benzene derivatives. *Journal of the American Chemical Society* 59, 1 (1937), 96–103.
- [51] HANSCH, C., AND LEO, A. Exploring QSAR: Fundamentals and Applications in Chemistry and Biology. *Applied Categorical Structures* (1995).
- [52] HARDY, B., DOUGLAS, N., HELMA, C., RAUTENBERG, M., JELIAZKOVA, N., JELIAZKOV, V., NIKOLOVA, I., BENIGNI, R., TCHEREMENSKAIA, O., KRAMER, S., GIRSCHICK, T., BUCHWALD, F., WICKER, J., KARWATH, A., GÜTLEIN, M., MAUNZ, A., SARIMVEIS, H., MELAGRAKI, G., AFANTITIS, A., SOPASAKIS, P., GALLAGHER, D., POROIKOV, V., FILIMONOV, D., ZAKHAROV, A., LANGUNIN, A., GLORIOZOVA, T., NOVIKOV, S., SKVORTSOVA, N., DRUZHILOVSKY, D., CHAWLA, S., GOSH, I., RAY, S., PATEL, H., AND ESCHER, S. Collaborative Development of Predictive Toxicology Applications. *Journal of Cheminformatics* 2, 7 (2010).
- [53] HAWIZY, L., JESSOP, D., ADAMS, N., AND MURRAY-RUST, P. Chemicaltagger: A tool for semantic text-mining in chemistry. *Journal of Cheminformatics* 3, 1 (2011), 17.
- [54] HE, L., AND JURIS, P. Assessing the reliability of a QSAR model's predictions. *Journal of Molecular Graphics and Modelling* 23, 6 (2005), 503–23.
- [55] HEIKAMP, K., AND BAJORATH, J. Large-scale similarity search profiling of chembl compound data sets. *Journal of Chemical Information and Modeling* 51, 8 (2011), 1831–1839.
- [56] HEIKAMP, K., AND BAJORATH, J. Prediction of compounds with closely related activity profiles using weighted support vector machine linear combinations. *Journal of Chemical Information and Modeling* 53, 4 (2013), 791–801.
- [57] HELMA, C. Lazy structure-activity relationships (lazar) for the prediction of rodent carcinogenicity and salmonella mutagenicity. *Molecular Diversity* 10, 2 (2006), 147–158.
- [58] HERT, J., WILLET, P., WILTON, D. J., ACKLIN, P., AZZAOU, K., JACOBY, E., AND SCHUFFENHAUER, A. Comparison of topological descriptors for similarity-based virtual screening using multiple bioactive reference structures. *Organic & Biomolecular Chemistry* 2 (2004), 3256–3266.

- [59] HILLEL, A., AND WEINSHALL, D. Learning distance function by coding similarity. In *Proceedings of the International Conference on Machine Learning (ICML'07)* (2007), pp. 65–72.
- [60] HORVÁTH, T., GÄRTNER, T., AND WROBEL, S. Cyclic pattern kernels for predictive graph mining. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'04)* (2004), pp. 158–167.
- [61] HSIEH, J., WANG, X., TEOTICO, D., GOLBRAIKH, A., AND TROPSHA, A. Differentiation of AmpC beta-lactamase binders vs. decoys using classification kNN QSAR modeling and application of the QSAR classifier to virtual screening. *Journal of Computer-Aided Molecular Design* 22, 9 (2008), 593–609.
- [62] HUANG, N., SHOICHET, B., AND IRWIN, J. Benchmarking sets for molecular docking. *Journal of Medicinal Chemistry* 49, 23 (2006), 6789–6801.
- [63] HURST, T. Flexible 3D searching: the directed tweak technique. *Journal of Chemical Information and Computer Sciences* 34, 1 (1994), 190–196.
- [64] IRWIN, J. J., STERLING, T., MYSINGER, M. M., BOLSTAD, E. S., AND COLEMAN, R. G. Zinc: A free tool to discover chemistry for biology. *Journal of Chemical Information and Modeling* 52, 7 (2012), 1757–1768.
- [65] ISTVAN, E., AND DEISENHOFER, J. Structural Mechanism for Statin Inhibition of HMG-CoA Reductase. *Science* 292, 5519 (2001), 1160–1164.
- [66] JAAKKOLA, T. S., AND HAUSSLER, D. Exploiting generative models in discriminative classifiers. In *Proceedings of the 1998 conference on Advances in neural information processing systems II* (Cambridge, MA, USA, 1999), MIT Press, pp. 487–493.
- [67] JAHN, K., AND KRAMER, S. Optimizing gSpan for Molecular Datasets. In *Proceedings of the International Workshop on Mining Graphs, Trees and Sequences (MGTS'05) at the ECML/PKDD'05* (2005).
- [68] JANAMANCHI, B., KATSAMAKAS, E., RAGHUPATHI, W., AND GAO, W. The state and profile of open source software projects in health and medical informatics. *International Journal of Medical Informatics* 78, 7 (2009), 457–72.
- [69] JAWORSKA, J., NIKOLOVA-JELIAZKOVA, N., AND ALDENBERG, T. QSAR applicability domain estimation by projection of the training set in descriptor space: A review. *ATLA. Alternatives to Laboratory Animals* 33, 5 (2005), 445–459.
- [70] JELIAZKOVA, N., AND JELIAZKOV, V. Ambit restful web services: an implementation of the opentox application programming interface. *Journal of Cheminformatics* 3, 1 (2011), 18.

- [71] JOHNSON, M., AND MAGGIORA, G., Eds. *Concepts and Applications of Molecular Similarity*. 1990.
- [72] KABSCH, W. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography* 32, 5 (1976), 922–923.
- [73] KIM, S., BOLTON, E., AND BRYANT, S. PubChem3D: Biologically relevant 3-D similarity. *Journal of Cheminformatics* 3, 1 (2011), 26.
- [74] KIM, S., BOLTON, E., AND BRYANT, S. PubChem3D: Shape compatibility filtering using molecular shape quadrupoles. *Journal of Cheminformatics* 3, 1 (2011), 25.
- [75] KNOX, C., LAW, V., JEWISON, T., LIU, P., LY, S., FROLKIS, A., PON, A., BANCO, K., MAK, C., NEVEU, V., DJOUMBOU, Y., EISNER, R., GUO, A. C., AND WISHART, D. S. DrugBank 3.0: a comprehensive resource for 'Omics' research on drugs. *Nucleic Acids Research* 39, suppl 1 (2011), D1035–D1041.
- [76] KUNCHEVA, L., AND WHITAKER, C. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning* 51, 2 (2003), 181–207.
- [77] LAMDAN, Y., AND WOLFSON, H. Geometric hashing: A general and efficient model-based recognition scheme. In *Proceedings of the International Conference on Computer Vision (ICCV'88)* (1988), pp. 238–249.
- [78] LEACH, A., AND GILLET, V. *An introduction to chemoinformatics*. Springer Verlag, 2007.
- [79] LEMMEN, C., AND LENGAUER, T. Computational methods for the structural alignment of molecules. *Journal of Computer-Aided Molecular Design* 14, 3 (2000), 215–232.
- [80] LEWINGTON, S., WHITLOCK, G., CLARKE, R., SHERLIKER, P., EMBERSON, J., HALSEY, J., QIZILBASH, N., PETO, R., AND COLLINS, R. Blood cholesterol and vascular mortality by age, sex, and blood pressure: a meta-analysis of individual data from 61 prospective studies with 55000 vascular deaths. *The Lancet* 370, 9602 (2007), 1829 – 1839.
- [81] LIBERATO, M. V., NASCIMENTO, A. S., AYERS, S. D., LIN, J. Z., CVORO, A., SILVEIRA, R. L., MARTÁNEZ, L., SOUZA, P. C. T., SAIDEMBERG, D., DENG, T., AMATO, A. A., TOGASHI, M., HSUEH, W. A., PHILLIPS, K., PALMA, M. S., NEVES, F. A. R., SKAF, M. S., WEBB, P., AND POLIKARPOV, I. Medium Chain Fatty Acids Are Selective Peroxisome Proliferator Activated Receptor (PPAR) gamma Activators and Pan-PPAR Partial Agonists. *PLoS ONE* 7, 5 (05 2012), e36297.

- [82] LIPINSKI, C. Drug-like properties and the causes of poor solubility and poor permeability. *Journal of Pharmacological and Toxicological Methods* 44, 1 (2000), 235–249.
- [83] MANLY, C., LOUISE-MAY, S., AND HAMMER, J. The impact of informatics and computational chemistry on synthesis and screening. *Drug Discovery Today* 6, 21 (2001), 1101–1110.
- [84] MARTIN, T. M., GRULKE, C. M., YOUNG, D. M., RUSSOM, C. L., WANG, N. Y., JACKSON, C. R., AND BARRON, M. G. Prediction of aquatic toxicity mode of action using linear discriminant and random forest models. *Journal of Chemical Information and Modeling* 53, 9 (2013), 2229–2239.
- [85] MATHIEU, M., ET AL. *Parexel's Bio/Pharmaceutical Ramp; D Statistical Sourcebook 2008/2009*. Barnett Educational Services/Chi, 2008.
- [86] MAUNZ, A., HELMA, C., CRAMER, T., AND KRAMER, S. Latent structure pattern mining. In *Balcazar, Jose; Bonchi, Francesco; Gionis, Aristides; Sebag, Michele: ECML/PKDD 2010: Machine Learning and Knowledge Discovery in Databases* (Berlin / Heidelberg, 2010), vol. 6322, Springer, pp. 353–368.
- [87] MAUNZ, A., HELMA, C., AND KRAMER, S. Efficient mining for structurally diverse subgraph patterns in large molecular databases. *Machine Learning* 83 (2011), 193–218.
- [88] MUNOS, B. Can open-source R&D reinvigorate drug research? *Nature Reviews Drug Discovery* 5, 723–729.
- [89] NADEAU, C., AND BENGIO, Y. Inference for the generalization error. *Machine Learning* 52, 3 (2003), 239–281.
- [90] NAPA, J. *Open Source Drug Discovery – A feasible business model?*, 2011. http://www.pharmafocusasia.com/strategy/open_source_drug_discovery.htm.
- [91] NEAMATI, N., AND BARCHI JR, J. New paradigms in drug design and discovery. *Current Topics in Medicinal Chemistry* 2, 3 (2002), 211–227.
- [92] NETZEVA, T., WORTH, A., ALDENBERG, T., BENIGNI, R., CRONIN, M., GRAMATICA, P., JAWORSKA, J., KAHN, S., KLOPMAN, G., MARCHANT, C., MYATT, G., NIKOLOVA-JELIAZKOVA, N., PATLEWICZ, G., PERKINS, R., ROBERTS, D., SCHULTZ, T., STANTON, D., VAN DE SANDT, J., TONG, W., VEITH, G., AND YANG, C. Current status of methods for defining the applicability domain of (quantitative) structure-activity relationships. *ATLA. Alternatives to Laboratory Animals* 33, 2 (2005), 1–19.
- [93] NIKOLOVA, N., AND JAWORSKA, J. Approaches to measure chemical similarity—a review. *QSAR & Combinatorial Science* 22, 9-10 (2003), 1006–1026.

- [94] O'BOYLE, N., BANCK, M., JAMES, C., MORLEY, C., VANDERMEERSCH, T., AND HUTCHISON, G. Open babel: An open chemical toolbox. *Journal of Cheminformatics* 3, 33 (2011).
- [95] OECD ENVIRONMENT DIRECTORATE. *Guidance Document on the Validation of (Quantitative) Structure-Activity Relationships [(Q)SAR] Models* (2007), Series on Testing and Assessment.
- [96] OPREA, T., AND GOTTFRIES, J. Chemography: the art of navigating in chemical space. *Journal of Combinatorial Chemistry* 3, 2 (2001), 157–166.
- [97] ORTÍ, L., CARBAJO, R., PIEPER, U., ESWAR, N., MAURER, S., RAI, A., TAYLOR, G., TODD, M., PINEDA-LUCENA, A., SALI, A., AND MARTI-RENOM, M. A kernel for open source drug discovery in tropical diseases. *PLoS Neglected Tropical Diseases* 3, 4 (2009), e418.
- [98] PARDOE, D., AND STONE, P. Boosting for regression transfer. In *Proceedings of the International Conference on Machine Learning (ICML'10)* (2010), ACM, pp. 863–870.
- [99] PATLEWICZ, G., JELIAZKOVA, N., SAFFORD, R., WORTH, A., AND ALEKSIEV, B. An evaluation of the implementation of the Cramer classification scheme in the toxtree software. *SAR & QSAR in Environmental Research* 19 (2008), 495–524.
- [100] PAUL, S., MYTELKA, D., DUNWIDDIE, C., PERSINGER, C., MUNOS, B., LINDBORG, S., AND SCHACHT, A. How to improve r&d productivity: the pharmaceutical industry's grand challenge. *Nature Reviews Drug Discovery* 9, 3 (2010), 203–214.
- [101] PEPPERRELL, C., AND WILLETT, P. Techniques for the calculation of three-dimensional structural similarity using inter-atomic distances. *Journal of Computer-Aided Molecular Design* 5, 5 (1991), 455–474.
- [102] QU, X., LATINO, D., AND AIRES-DE SOUSA, J. A big data approach to the ultra-fast prediction of dft-calculated bond energies. *Journal of Cheminformatics* 5, 1 (2013), 34.
- [103] RAJU, T. The nobel chronicles. 1988: James whyte black, (b 1924), gertrude elion (1918-99), and george h hitchings (1905-98). *The Lancet* 355, 9208 (2000), 1022.
- [104] RAYMOND, J., GARDINER, E., AND WILLETT, P. RASCAL: Calculation of Graph Similarity using Maximum Common Edge Subgraphs. *The Computer Journal* 45, 6 (2002), 631–644.
- [105] RAYMOND, J., AND WILLETT, P. Effectiveness of graph-based and fingerprint-based similarity measures for virtual screening of 2D chemical structure databases. *Journal of Computer-Aided Molecular Design* 16, 1 (2002), 59–71.

- [106] RESTER, U. From virtuality to reality-virtual screening in lead discovery and lead optimization: a medicinal chemistry perspective. *Current Opinion in Drug Discovery & Development* 11, 4 (2008), 559.
- [107] RICHARDSON, L., AND RUBY, S. *RESTful Web Services*, 1 ed. O'Reilly Media, 2007.
- [108] RICHTER, L., HECHTL, S., AND KRAMER, S. Leveraging chemical background knowledge for the prediction of growth inhibition. In *Proceedings of the Conference of BioInformatics and BioEngineering (BIBE'06)* (2006), pp. 319–324.
- [109] RICHTER, L., RÜCKERT, U., AND KRAMER, S. Learning a predictive model for growth inhibition from the NCI DTP human tumor cell line screening data: does gene expression make a difference? In *Proceedings of the Pacific Symposium on Biocomputing (PSB'06)* (2006), vol. 11.
- [110] RODGERS, S., DAVIS, A., TOMKINSON, N., AND VAN DE WATERBEEMD, H. Predictivity of simulated adme autoqsar models over time. *Molecular Informatics* 30, 2-3 (2011), 256–266.
- [111] ROGERS, D., AND HAHN, M. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling* 50, 5 (2010), 742–754.
- [112] ROLLINGER, J., STUPPNER, H., AND LANGER, T. Virtual screening for the discovery of bioactive natural products. *Natural Compounds as Drugs* 1 (2008), 211–249.
- [113] RÜCKERT, U., AND KRAMER, S. Frequent free tree discovery in graph data. In *Proceedings of the ACM SIG Symposium on Applied Computing (SAC'04)* (2004), ACM Press, pp. 564–570.
- [114] RÜCKERT, U., AND KRAMER, S. Optimizing feature sets for structured data. In *Proc. of ECML'07* (2007), J. Kok, J. Koronacki, R. Lopez de Mantaras, S. Matwin, and D. Mladenic, Eds., vol. 4701 of *LNCS/LNAI*, Springer, pp. 716–723.
- [115] RÜCKERT, U., AND KRAMER, S. Kernel-based inductive transfer. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD'08)* (2008), ECML PKDD '08, Springer, pp. 220–233.
- [116] RUIZ, I., CUADRADO, M., AND GÓMEZ-NIETO, M. Combining Similarity and Dissimilarity Measurements for the Development of QSAR Models Applied to the Prediction of Antiobesity Activity of Drugs. *International Journal of Biological and Medical Sciences* 2, 3 (2007).
- [117] SAMWALD, M., JENTZSCH, A., BOUTON, C., KALLESOE, C., WILLIGHAGEN, E., HAJAGOS, J., MARSHALL, M., PRUD'HOMMEAUX, E., HASSENZADEH, O.,

- PICHLER, E., AND STEPHENS, S. Linked open drug data for pharmaceutical research and development. *Journal of Cheminformatics* 3, 1 (2011), 19.
- [118] SCARSI, M., PODVINEC, M., ROTH, A., HUG, H., KERSTEN, S., ALBRECHT, H., SCHWEDE, T., MEYER, U. A., AND RUECKER, C. Sulfonylureas and glinides exhibit peroxisome proliferator-activated receptor gamma activity: A combined virtual screening and biological assay approach. *Molecular Pharmacology* 71, 2 (2007), 398–406.
- [119] SCHÖLKOPF, B., TSUDA, K., AND VERT, J. *Kernel methods in computational biology*. The MIT press, 2004.
- [120] SEELAND, M., BERGER, S., STAMATAKIS, A., AND KRAMER, S. Parallel structural graph clustering. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD'11)* (2011), vol. 3, pp. 256–272.
- [121] SEELAND, M., GIRSCHICK, T., BUCHWALD, F., AND KRAMER, S. Online structural graph clustering using frequent subgraph mining. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD'10)* (2010), J. Balcázar, F. Bonchi, A. Gionis, and M. Sebag, Eds., vol. 3 of *Lecture Notes in Computer Science*, Springer, pp. 213–228.
- [122] SHANG, H., LIN, X., ZHANG, Y., YU, J., AND WANG, W. Connected substructure similarity search. In *Proceedings of the ACM SIGMOD/PODS Conference (SIGMOD'10)* (2010), pp. 903–914.
- [123] SHERVASHIDZE, N., VISHWANATHAN, S., PETRI, T., MEHLHORN, K., AND BORGWARDT, K. Efficient Graphlet Kernels for Large Graph Comparison. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTats'09)* (2009).
- [124] SIPPL, M. On distance and similarity in fold space. *Bioinformatics* 24, 6 (2008), 872–873.
- [125] SOMMER, S., AND KRAMER, S. Three data mining techniques to improve lazy structure-activity relationships for noncongeneric compounds. *Journal of Chemical Information and Modeling* 47, 6 (2007), 2035–2043.
- [126] STAHL, M. Open-source software: not quite endsville. *Drug Discovery Today* 10, 3 (2005), 219–22.
- [127] STALRING, J., CARLSSON, L., ALMEIDA, P., AND BOYER, S. AZOrange-High performance open source machine learning for QSAR modeling in a graphical programming environment. *Journal of Cheminformatics* 3, 1 (2011), 28.

- [128] STEINBECK, C., HOPPE, C., KUHN, S., FLORIS, M., GUHA, R., AND WILLIGHAGEN, E. Recent developments of the chemistry development kit (CDK) – an open-source java library for chemo- and bioinformatics. *Current Pharmaceutical Design* 12 (2006), 2111–2120.
- [129] STIERAND, K., MAASS, P., AND RAREY, M. Molecular complexes at a glance: automated generation of two-dimensional complex diagrams. *Bioinformatics* 22, 14 (2006), 1710–1716.
- [130] SUSHKO, I., NOVOTARSKYI, S., KÖRNER, R., PANDEY, A., RUPP, M., TEETZ, W., BRANDMAIER, S., ABDELAZIZ, A., PROKOPENKO, V., TANCHUK, V., TODESCHINI, R., VARNEK, A., MARCOU, G., ERTL, P., POTEMKIN, V., GRISHINA, M., GASTEIGER, J., SCHWAB, C., BASKIN, I., PALYULIN, V., RADCHENKO, E., WELSH, W., KHOLODOVYCH, V., CHERKASOV, A., DE SOUSA, J. A., ZHANG, Q., BENDER, A., NIGSCH, F., PATINY, L., WILLIAMS, A., TKACHENKO, V., TETKO, I., AND CHEKMAREV, D. Online chemical modeling environment (OCHEM): web platform for data storage, model development and publishing of chemical information. *Journal of Computer-Aided Molecular Design* 25, 6 (2011), 533–554.
- [131] SUTHERLAND, J. J., O'BRIEN, L., AND WEAVER, D. Spline-fitting with a genetic algorithm: A method for developing classification structure-activity relationships. *Journal of Chemical Information and Modeling* 43, 6 (2003), 1906–1915.
- [132] SUTHERLAND, J. J., O'BRIEN, L., AND WEAVER, D. A comparison of methods for modeling quantitative structure-activity relationships. *Journal of Medicinal Chemistry* 47, 22 (2004), 5541–5554.
- [133] SWAMIDASS, S., CHEN, J., BRUAND, J., PHUNG, P., RALAIVOLA, L., AND BALDI, P. Kernels for small molecules and the prediction of mutagenicity, toxicity and anti-cancer activity. *Bioinformatics* 21, suppl 1 (2005), i359–i368.
- [134] TABEL, Y., AND TSUDA, K. Kernel-based similarity search in massive graph databases with wavelet trees. In *Proceedings of the SIAM Conference on Data Mining (SDM'11)* (2011), pp. 154–163.
- [135] TANIMOTO, T. IBM Internal Report. Tech. rep., IBM, 1957.
- [136] TERSTAPPEN, G., AND REGGIANI, A. In silico research in drug discovery. *Trends in Pharmacological Sciences* 22, 1 (2001), 23–26.
- [137] TETKO, I., BRUNEAU, P., MEWES, H., ROHRER, D., AND PODA, G. Can we estimate the accuracy of ADME-Tox predictions? *Drug Discovery Today* 11, 15-16 (2006), 700–707.
- [138] TODESCHINI, R., AND CONSONNI, V. *Molecular descriptors for chemoinformatics*. Vch Pub, 2009.

-
- [139] TSUDA, K. Support vector classifier with asymmetric kernel functions. In *Proceedings of the European Symposium on Artificial Neural Networks (ESANN'99)* (1999), pp. 183–188.
- [140] TVERSKY, A. Features of similarity. *Psychological Reviews* 84, 4 (1977), 327–352.
- [141] VAN DE WATERBEEMED, H., AND GIFFORD, E. ADMET in silico modelling: towards prediction paradise? *Nature Reviews Drug Discovery* 2 (2003), 192–204.
- [142] VAPNIK, V. *The nature of statistical learning theory*. Springer, 1995.
- [143] VREEKEN, J., VAN LEEUWEN, M., AND SIEBES, A. Krimp: mining itemsets that compress. *Data Mining and Knowledge Discovery* 23, 1 (2011), 169–214.
- [144] WALLIS, W., SHOUBRIDGE, P., KRAETZ, M., AND RAY, D. Graph distances using graph union. *Pattern Recognition Letters* 22 (2001), 701–704.
- [145] WANG, X., HUAN, J., SMALTER, A., AND LUSHINGTON, G. Application of kernel functions for accurate similarity search in large chemical databases. *BMC Bioinformatics* 11, Suppl 3 (2010), S8.
- [146] WANG, X., SMALTER, A., HUAN, J., AND LUSHINGTON, G. G-hash: towards fast kernel-based similarity search in large graph databases. In *Proceedings of the International Conference on Extending Database Technology (EDBT'09)* (2009), ACM, pp. 472–480.
- [147] WANG, Y., BOLTON, E., DRACHEVA, S., KARAPETYAN, K., SHOEMAKER, B., SUZEK, T., WANG, J., XIAO, J., ZHANG, J., AND BRYANT, S. An overview of the PubChem BioAssay resource. *Nucleic Acids Research* 38(Database issue), suppl 1 (2010), D255–D266.
- [148] WEINBERGER, K., AND TESAURO, G. Metric learning for kernel regression. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTats'07)* (2007).
- [149] WEINBERGER, K. Q., AND SAUL, L. K. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research* 10 (2009), 207–244.
- [150] WEININGER, D., WEININGER, A., AND WEININGER, J. SMILES. 2. algorithm for generation of unique SMILES notation. *Journal of Chemical Information and Computer Sciences* 29, 2 (1989), 97–101.
- [151] WIENER, H. Structural determination of paraffin boiling points. *Journal of the American Chemical Society* 69, 1 (1947), 17–20.
- [152] WILLETT, P., BARNARD, J., AND DOWNS, G. Chemical Similarity Searching. *Journal of Chemical Information and Computer Sciences* 38, 6 (1998), 983–996.

- [153] WILLIGHAGEN, E., AND BRANDLE, M. Resource description framework technologies in chemistry. *Journal of Cheminformatics* 3, 1 (2011), 15.
- [154] WILLIGHAGEN, E., JELIAZKOVA, N., HARDY, B., GRAFSTROM, R., AND SPJUTH, O. Computational toxicology using the OpenTox application programming interface and Bioclipse. *BMC Research Notes* 4, 1 (2011), 487.
- [155] WILTON, D., WILLETT, P., LAWSON, K., AND MULLIER, G. Comparison of ranking methods for virtual screening in lead-discovery programs. *Journal of Chemical Information and Computer Sciences* 43, 2 (2003), 469–474.
- [156] WOLPERT, D. Stacked generalization. *Neural networks* 5, 2 (1992), 241–259.
- [157] WOOD, D., BUTTAR, D., CUMMING, J., DAVIS, A., NORINDER, U., AND RODGERS, S. Automated QSAR with a hierarchy of global and local models. *Molecular Informatics* 30, 11-12 (2011), 960–972.
- [158] WOZNICA, A., KALOUSIS, A., AND HILARIO, M. Learning to combine distances for complex representations. In *Proceedings of the International Conference on Machine Learning (ICML'07)* (2007), pp. 1031–1038.
- [159] XING, E., NG, A., JORDAN, M., AND RUSSELL, S. Distance metric learning with application to clustering with side-information. In *Advances in Neural Information Processing Systems 15*. MIT Press, 2003, pp. 505–512.
- [160] XU, K., AND COTÉ, T. Database identifies FDA-approved drugs with potential to be repurposed for treatment of orphan diseases. *Briefings in Bioinformatics* 12, 4 (2011), 341–345.
- [161] YAN, X., AND HAN, J. gSpan: Graph-based substructure pattern mining. In *Proceedings of the International Conference on Data Mining (ICDM'02)* (2002), pp. 721–724.
- [162] YAN, X., YU, P., AND HAN, J. Substructure similarity search in graph databases. In *Proceedings of the ACM SIGMOD/PODS Conference (SIGMOD'05)* (2005), pp. 766–777.
- [163] ZHA, Z.-J., MEI, T., WANG, M., WANG, Z., AND HUA, X.-S. Robust distance metric learning with auxiliary knowledge. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'09)* (2009), pp. 1327–1332.