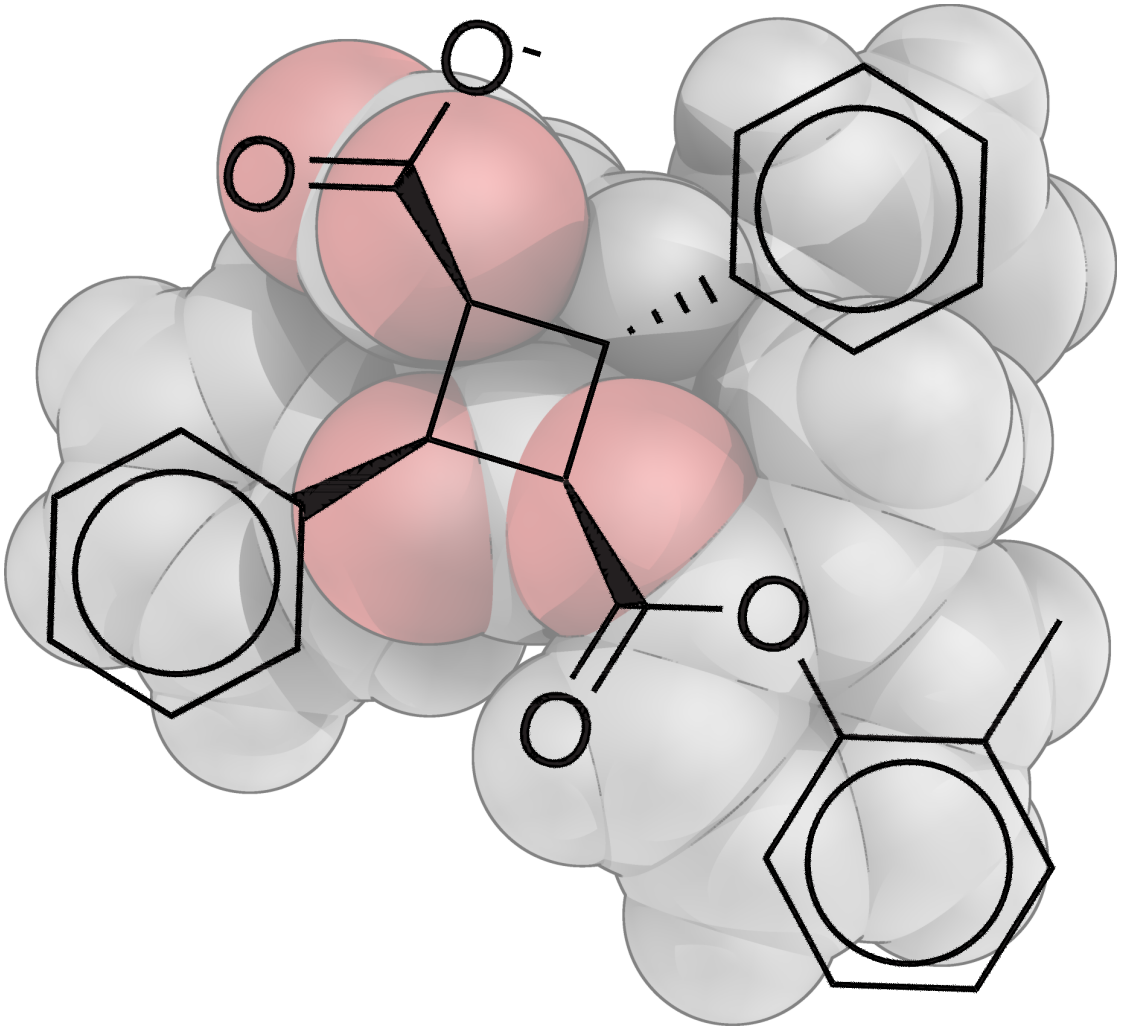
Kernel Methods for Virtual Screening

Matthias Rupp

Kernel Methods for Virtual Screening

Matthias Rupp



A dissertation submitted to the Johann Wolfgang Goethe-University, Frankfurt am Main, Germany, in partial fulfillment of the requirements for the degree of doctor of natural sciences (doctor philosophiae naturalis, dr. phil. nat.), subject bioinformatics.

Supervisors: Prof. Dr. Gisbert Schneider
(Chem- and bioinformatics)
Institute for organic chemistry and chemical biology
Johann Wolfgang Goethe-University
60323 Frankfurt am Main, Germany

Prof. Dr. Klaus-Robert Müller
(Machine learning)
Institute for software engineering and theoretical informatics
Technical University of Berlin
10587 Berlin, Germany

Prof. Dr. Manfred Schubert-Zsilavecz
(Analysis of natural drugs and drug synthesis)
Institute for pharmaceutical chemistry
Johann Wolfgang Goethe-University
60323 Frankfurt am Main, Germany

Version of February 15th, 2013.
Typeset by the author using LaTeX 2ε.

Abstract

We investigate the utility of modern kernel-based machine learning methods for ligand-based virtual screening. In particular, we introduce a new graph kernel based on iterative graph similarity and optimal assignments, apply kernel principal component analysis to projection error-based novelty detection, and discover a new selective agonist of the peroxisome proliferator-activated receptor γ using Gaussian process regression.

Virtual screening, the computational ranking of compounds with respect to a predicted property, is a cheminformatics problem relevant to the hit generation phase of drug development. Its ligand-based variant relies on the similarity principle, which states that (structurally) similar compounds tend to have similar properties. We describe the kernel-based machine learning approach to ligand-based virtual screening; in this, we stress the role of molecular representations, including the (dis)similarity measures defined on them, investigate **effects in high-dimensional chemical descriptor spaces** and their consequences for similarity-based approaches, review **literature recommendations on retrospective virtual screening**, and present an example workflow.

Graph kernels are formal similarity measures that are defined directly on graphs, such as the annotated molecular structure graph, and correspond to inner products. We review **graph kernels**, in particular those based on random walks, subgraphs, and optimal vertex assignments. Combining the latter with an iterative graph similarity scheme, we develop the **iterative similarity optimal assignment graph kernel**, give an iterative algorithm for its computation, prove convergence of the algorithm and the uniqueness of the solution, and provide an upper bound on the number of iterations necessary to achieve a desired precision. In a retrospective virtual screening study, our kernel consistently improved performance over chemical descriptors as well as other optimal assignment graph kernels.

Chemical data sets often lie on manifolds of lower dimensionality than the embedding chemical descriptor space. Dimensionality reduction methods try to identify these manifolds, effectively providing descriptive models of the data. For spectral methods based on **kernel principal component analysis**, the projection error is a quantitative measure of how well new samples are described by such models. This can be used for the identification of compounds structurally dissimilar to the training samples, leading to **projection error-based novelty detection for virtual screening using only positive samples**. We provide proof of principle by using principal component analysis to learn the concept of fatty acids.

The peroxisome proliferator-activated receptor (PPAR) is a nuclear transcription factor that regulates lipid and glucose metabolism, playing a crucial role in the development of type 2 diabetes and dyslipidemia. We establish a **Gaussian process regression** model for PPARγ agonists using a combination of chemical descriptors and the iterative similarity optimal assignment kernel via multiple kernel learning. Screening of a vendor library and subsequent testing of 15 selected compounds in a cell-based transactivation assay resulted in 4 active compounds (27 % hit rate). One compound, a natural product with cyclobutane scaffold, is a full selective PPARγ-agonist ($EC_{50} = 10 \pm 0.2\,\mu$M, inactive on PPAR$\alpha$ and PPARβ/δ at $10\,\mu$M). The study delivered a **novel PPARγ agonist**, de-orphanized a natural bioactive product, and, hints at the natural product origins of pharmacophore patterns in synthetic ligands.

Contents

Detailed contents

Notation, symbols, abbreviations

Symbol	Description
Algebra	
$\mathbb{N}$, $\mathbb{N}_0$	Set of natural numbers $\{1, 2, \ldots\}$. $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$
$\mathbb{R}$, $\mathbb{R}_{\geq 0}$, $\mathbb{R}_+$	Set of real numbers. $\mathbb{R}_{\geq 0} = \{x \in \mathbb{R} \mid x \geq 0\}$, $\mathbb{R}_+ = \{x \in \mathbb{R} \mid x > 0\}$
$[a, b]$	Closed interval on the real line, $[a, b] = \{x \mid a \leq x \leq b\}$
(a, b)	Open interval on the real line, $(a, b) = \{x \mid a < x < b\}$
$\boldsymbol{x}^T, \boldsymbol{M}^T$	Transpose of vector $\boldsymbol{x}$, transpose of matrix $\boldsymbol{M}$
$\mathrm{Tr}(\boldsymbol{M})$	Trace $\sum_{i=1}^{n} \boldsymbol{M}_{i,i}$ of a quadratic matrix $\boldsymbol{M} \in \mathbb{R}^{n \times n}$
$\mathrm{vec}(\boldsymbol{M})$	Concatenation of the columns of a matrix $\boldsymbol{M}$ into a vector
$\mathrm{diag}(\boldsymbol{M})$	Diagonal of a quadratic matrix $\boldsymbol{M}$
$\mathbf{1}_p$, $\mathbf{1}_{p \times q}$	Vector (matrix) of length p (dimension $p \times q$) with all entries 1
$1_{\{\mathrm{cond}\}}$	Indicator function; 1 iff cond is true, 0 otherwise
$\langle \boldsymbol{x}, \boldsymbol{z} \rangle$	Inner product (also dot product) between (vectors) $\boldsymbol{x}$ and $\boldsymbol{z}$
iff	If and only if
Stochastics	
$\mathrm{E}(A)$	Expectation of a random variable A
$\mathrm{E}(A \mid B)$	Conditional expectation of a random variable A given an event B
$\mathbb{P}(A)$	Probability of an event A
$\mathbb{P}(A \mid B)$	Conditional probability of an event A given an event B
$\mathrm{var}(A)$	Variance of a random variable A
$\mathrm{covar}(A, B)$	Covariance of two random variables A and B
$\boldsymbol{C}$	(Empirical) covariance matrix
$\mathcal{N}(\mu, \sigma^2)$	The normal distribution with mean μ and variance σ^2
i. i. d.	Independent and identically distributed
Q_x	$x\,\%$ quantile
Graph theory	
$G = (V, E)$	Graph G with set of vertices V and set of edges E
$G_\times$	The direct product graph of two other graphs
v_i, e_j	The i-th vertex and the j-th edge of a graph
$n(v)$	The set of neighbors of a vertex v
k_v, k_e	Kernels on vertices and edges of a graph, $0 \leq k_v, k_e \leq 1$
$\mathcal{L}$	The set of vertex and edge labels
$\boldsymbol{A}$	The $\lvert V \rvert \times \lvert V \rvert$ adjacency matrix of a graph
$\boldsymbol{L}$	The $\lvert \mathcal{L} \rvert \times \lvert V \rvert$ label-vertex matrix of a graph

Continued on next page...

...continued from previous page

Symbol	Description
Machine learning	
$\mathcal{X}$	Input domain; a non-empty set
$\mathcal{Y}$	Target label domain; a non-empty set
$\mathcal{H}$	Feature space with inner product
ϕ	Feature map $\phi : \mathcal{X} \to \mathcal{H}$
k	Kernel; (conditionally) positive semidefinite function $k : \mathcal{X} \times \mathcal{X} \to \mathcal{H}$
$\boldsymbol{K}$	Kernel matrix (also Gram matrix) of training samples, $\boldsymbol{K}_{i,j} = k(x_i, x_j)$
$\boldsymbol{L}$	Kernel matrix between training and test samples, $\boldsymbol{L}_{i,j} = k(x_i, x'_j)$
n, m	Number of training samples, number of test samples
$x_1, \ldots, x_n$	Training samples; if vectorial, $\boldsymbol{x}_{i,j}$ denotes the j-th component of $\boldsymbol{x}_i$
$x'_1, \ldots, x'_m$	Test samples; if vectorial, $\boldsymbol{x}'_{i,j}$ denotes the j-th component of $\boldsymbol{x}'_i$
$\boldsymbol{X}$	Matrix of vectorial training samples (rows), $\boldsymbol{X}_{i,j} = \boldsymbol{x}_{i,j}$
$\boldsymbol{X}'$	Matrix of vectorial test samples (rows), $\boldsymbol{X}'_{i,j} = \boldsymbol{x}'_{i,j}$
GP	Gaussian process
RBF	Radial basis function kernel
FI_{20}	Fraction of inactives among the 20 top-ranked compounds
RMSE	Root mean squared error
ROC (AUC)	Receiver operating characteristic (area under curve)
SVM	Support vector machine
PC, PCA	Principal component, principal component analysis
λ_i	The i-th PCA eigenvalue, sorted in descending order by absolute size
$\boldsymbol{v}_i$	The i-th PCA eigenvector, using the same order as the eigenvalues
$\boldsymbol{V}$	The $q \times d$ matrix with $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_q$ as rows
Natural sciences	
EC_{50}, IC_{50}	Half maximal effective (inhibitory) concentration
K_i	Dissociation constant
pEC_{50}	Negative decadic logarithm of the EC_{50}, $\mathrm{pEC}_{50} = -\log_{10} \mathrm{EC}_{50}$
pIC_{50}	Negative decadic logarithm of the IC_{50}, $\mathrm{pIC}_{50} = -\log_{10} \mathrm{IC}_{50}$
$\mathrm{p}K_i$	Negative decadic logarithm of the K_i, $\mathrm{p}K_i = -\log_{10} K_i$
ppm	Parts per million
AF-1, AF-2	Activation function 1, activation function 2
DNA, RNA	(Deoxy)ribonucleic acid
DBD, LBD	DNA-binding domain, ligand-binding domain
QSAR	Quantitative structure-activity relationships
QSPR	Quantitative structure-property relationships
HTS	High-throughput screening
PDB	Protein data bank, `www.rcsb.org`
PPAR	Peroxisome proliferator-activated receptor, with subtypes α, β/δ, and γ
hPPAR	Human peroxisome proliferator-activated receptor
PPRE	Peroxisome proliferator-activated receptor response element
RXR	Retinoid X receptor
Other	
USD	United states of America dollars

List of Figures

List of Tables

List of Schemes

List of Algorithms

I would rather discover one scientific fact
than become king of Persia.

Democritus (430 B. C.)

Preface

In this doctoral thesis, I present results from three years of my research into kernel-based learning methods for ligand-based virtual screening. Most of all, it was a great time, and I hope to convey to you, the reader, part of the excitement, curiosity, and satisfaction that I experienced during this time.

Scope and contribution

The central theme of this thesis is the investigation of the utility of modern kernel-based machine learning methods for ligand-based virtual screening. This includes the modification and further development of these methods with respect to the specific requirements of this application. The underlying hypothesis is that *ligand-based virtual screening can benefit from modern kernel learning methods*. Four thematically self-contained chapters address different aspects of this theme. The contributions of this thesis are

- *Chapter 1 (ligand-based virtual screening):* An introduction to ligand-based virtual screening, an investigation of distance phenomena in high-dimensional chemical descriptor spaces, a survey of literature recommendations on retrospective validation.
- *Chapter 2 (iterative similarity optimal assignment graph kernel):* A survey of graph kernels, development and retrospective validation of a new graph kernel.
- *Chapter 3 (dimensionality reduction and novelty detection):* An introduction to kernel principal component analysis, proof of principle for projection error-based novelty detection for ligand-based virtual screening using only positive samples.
- *Chapter 4 (peroxisome proliferator-activated receptor):* A survey of this receptor, a prospective virtual screening study using Gaussian process regression yielding a selective agonist of the peroxisome proliferator-activated receptor γ with new scaffold.

In recognition of others' contributions, and to maintain the habitual style of the scientific literature, the thesis is written in the first person plural. Specific contributions by others:

- Dr. Ewgenij Proschak performed the docking experiments (Figure 4.12).
- Dr. Oliver Rau, Heiko Zettl, Stephan Bihler, Ramona Steri, and Michaela Dittrich determined the stereochemistry of Compound MR16 (p. 169; Figures A.1–A.4).
- Timon Schroeter and Katja Hansen, with help of Fabian Rathke and Peter Vascovic, ran the Gaussian process computations (Subsection 4.2.6).
- Ramona Steri carried out the transactivation assay measurements (Subsection 4.3.2).

Parts of this thesis have been published:

- Ramona Steri, Matthias Rupp, Ewgenij Proschak, Timon Schroeter, Heiko Zettl, Katja Hansen, Oliver Schwarz, Lutz Müller-Kuhrt, Klaus-Robert Müller, Gisbert Schneider, Manfred Schubert-Zsilavecz: *Truxillic acid derivatives act as peroxisome proliferator-activated receptor γ activators*, submitted, 2010.
- Matthias Rupp, Gisbert Schneider: *Graph kernels for molecular similarity*, submitted, 2010.
- Matthias Rupp, Timon Schroeter, Ramona Steri, Heiko Zettl, Ewgenij Proschak, Katja Hansen, Oliver Rau, Oliver Schwarz, Lutz Müller-Kuhrt, Manfred Schubert-Zsilavecz, Klaus-Robert Müller, Gisbert Schneider: *From machine learning to bioactive natural products selectively activating transcription factor PPARγ*, ChemMedChem 5(2): 191–194, Wiley, 2010.
- Matthias Rupp, Timon Schroeter, Ramona Steri, Ewgenij Proschak, Katja Hansen, Heiko Zettl, Oliver Rau, Manfred Schubert-Zsilavecz, Klaus-Robert Müller, Gisbert Schneider: *Kernel learning for virtual screening: discovery of a new PPARγ agonist*, poster, 5th German Conference on Chemoinformatics, Goslar, Germany, 2009.
- Matthias Rupp, Petra Schneider, Gisbert Schneider: *Distance phenomena in high-dimensional chemical descriptor spaces: consequences for similarity-based approaches*, Journal of Computational Chemistry 30(14): 2285–2296, Wiley, 2009.
- Matthias Rupp, Petra Schneider, Gisbert Schneider: *Distance phenomena in chemical spaces: consequences for similarity approaches*, poster, 4th German Conference on Chemoinformatics, Goslar, Germany, 2008.
- Matthias Rupp, Ewgenij Proschak, Gisbert Schneider: *Kernel approach to molecular similarity based on iterative graph similarity*, Journal of Chemical Information and Modeling 47(6): 2280–2286, American Chemical Society, 2007.
- Matthias Rupp, Ewgenij Proschak, Gisbert Schneider: *Molecular similarity for machine learning in drug development*, poster, 3rd German Conference on Chemoinformatics, Goslar, Germany, 2007. Best poster award.

Other publications during this time:

- Ramona Steri, Petra Schneider, Alexander Klenner, Matthias Rupp, Manfred Schubert-Zsilavecz, Gisbert Schneider: *Target profile prediction: cross-activation of peroxisome proliferator-activated receptor (PPAR) and farnesoid X Receptor (FXR)*, Molecular Informatics, accepted, Wiley, 2009.
- Ewgenij Proschak, Matthias Rupp, Swetlana Derksen, Gisbert Schneider: *Shapelets: Possibilities and limitations of shape-based virtual screening*, Journal of Computational Chemistry 29(1): 108-114, Wiley, 2008.

Acknowledgments

This work was made possible by the helpful input from, productive cooperation with, constructive criticism by, and continuous support from other people. I thank

- My supervisor, Prof. Dr. Gisbert Schneider, with profound gratitude, for making this possible, for excellent support, and for believing in me.
- My father, Reinhard Rupp, and my late mother, Ursula Rupp, for unconditional support and love. My brother, Dr. Thomas Rupp, and my sisters, Marina Rupp and Marlene Rupp, for being family.

- My significant other, Simone Galluba, for loving me, and for bearing with me.
- My cooperation partners in Berlin, Prof. Dr. Klaus-Robert Müller, Timon Schroeter, and Katja Hansen, for their machine learning expertise, constructive discussions, and enjoyable visits.
- My cooperation partners in Frankfurt, Prof. Dr. Manfred Schubert-Zsilavecz, Ramona Steri, Heiko Zettl, Oliver Rau, Stephan Bihler, and Michaela Dittrich, for taking chances with our predictions, the experimental work, and constructive discussions.
- The master students that I helped to supervise, for their great work: Alexander Klenner, Quan Wang, Bärbel Lasitschka.
- The modlab team (`www.modlab.de`) for a great working atmosphere and constructive discussions: Janosch Aschenbach, Dr. Swetlana Derksen, Norbert Dichter, Dr. Uli Fechner, Dr. Lutz Franke, Tim Geppert, Kristina Grabowski, Volker Hähnke, Markus Hartenfeller, Dr. Jan Hiss, Dr. Bettina Hofmann, Natalie Jäger, Mahmut Kara, Sarah Keppner, Robert Kleinkauf, Alexander Klenner, Christian Koch, Dr. Björn Krüger, Bärbel Lasitschka, Dr. Martin Löwer, Michael Meissner, Dr. Manuel Nietert, Nils Oberhauser, Dr. Ewgenij Proschak, Felix Reisen, Carmen Rödel, Florian Roersch, Brigitte Scheidemantel-Geiß, Dr. Andreas Schüller, Dr. Yusuf Tanrikulu, Quan Wang, Dr. Martin Weisel, Tim Werner, Matthias Wirth, Joanna Wisniewska.
- All members of the Frankfurt international research graduate school for translational biomedicine, for the opportunity to learn, and for companionship along the way.
- The volunteers who proofread the manuscript or parts of it: Tim Geppert, Volker Hähnke, Dr. Jan Hiss, Mahmut Kara, Alexander Klenner, Robert Körner, Dr. Martin Löwer, Antonin Nemec, Felix Reisen, Dr. Thomas Rupp.
- For constructive discussions: Dr. Holger Fröhlich, Antonin Nemec, Dr. Thomas Rupp.
- My new colleagues at the Helmholtz center Munich, Dr. Igor Tetko, Stefan Brandmaier, Robert Körner, Sergii Novotarskyi, Anil Pandey, and Iurii Sushko, for welcoming me into their group.
- All my friends whom I neglected during writing, for their understanding.
- I almost surely forgot one person or another; these, I dearly thank for their contribution, and apologize for not listing them.

I acknowledge financial support by the following organizations:

- The Frankfurt international research graduate school for translational biomedicine (FIRST), for a PhD scholarship.
- The collaborative research center 579 RNA-ligand interactions for project support.
- The Beilstein institute for the advancement of chemical sciences, for sponsorship of the host chair.
- The German chemical society for travel grants.
- The Helmholtz center Munich for a follow-up position.

Frankfurt am Main,
Matthias Rupp June 28th, 2009

The most fruitful basis
for the discovery of a new drug
is to start with an old drug.

James Black

Chapter 1

Ligand-based virtual screening

Virtual screening is a general term describing the computer-based evaluation of chemical compounds with regard to various properties, often related to drug development. In this chapter, we describe virtual screening in this context and present the (kernel-based) machine learning approach to its ligand-based variant. In doing so, we stress the importance of molecular representations, of the measures of (dis)similarity defined on them, and of sound retrospective validation. We summarize literature recommendations on retrospective virtual screening and present an example workflow.

1.1 Introduction

The term virtual screening denotes several related but distinct tasks, each with differing assumptions (p. 24). These tasks are in general well suited for machine learning approaches, with various learning problems and algorithms being appropriate, depending on circumstances.

1.1.1 Background

Cheminformatics

Virtual screening comprises aspects of computer science as well as of pharmacology, biochemistry, biology, and medicine. It belongs to the field of *cheminformatics* (also *chemoinformatics*; Bajorath, 2004), which is "the application of informatics methods to solve chemical problems" (Gasteiger, 2006), with the distinction to the neighboring disciplines of computational chemistry and bioinformatics not always clear-cut.

Important aspects of cheminformatics in general and virtual screening in particular include compound database processing, motivated by the huge amount of chemical data involved, and inductive learning approaches, motivated by the computational infeasibility of first principles approaches (p. 32). Cheminformatics applications that relate to virtual screening include:

- *Quantitative structure-activity relationships* (QSAR; Kubinyi, 2003) and *quantitative structure-property relationships* (QSPR; Jurs, 2003)*:* The establishment of statistical models that relate molecular representations to either activity on a given target or to physico-chemical properties. This often involves feature selection on descriptors. Examples of physico-chemical properties are solubility measured by the octanol/water partition coefficient, as well as absorption, distribution, metabolism, excretion (ADME), oral bioavailability, and toxicity. Since these are selection criteria related to drug development, QSAR/QSPR models can be used for virtual screening.
- *Diversity analysis for focused library design* (Ghose and Viswanadhan, 2001)*:* On the one hand, virtual screening can be used to create small diverse compound libraries geared towards a specific target, for use with bioassay tests. On the other hand, diversity is an important aspect in data set design for virtual screening purposes.
- *De novo design* (Schneider and Fechner, 2005)*:* Molecules are virtually created from scratch, e. g., by the application of synthesis rules to fragment libraries. The resulting virtual compounds can be used as input for virtual screening. Alternatively, compound creation can be guided by the predicted property.

For further information on cheminformatics, see the review by Gasteiger (2006).

Drug development

Virtual screening is used mainly for the development of new drugs, during the hit generation phase (Figure 1.1).[1] There, small compounds are sought that interact in the desired way with an identified and confirmed molecular target, e. g., a receptor, channel, gene, or other biopolymer.[2] This is done by *screening*, the systematic investigation of a large number of compounds with respect to the desired target interaction.

Compounds are called *hits* if their activity is experimentally confirmed. Hits with improvement potential become leads, and are further optimized, some eventually becoming drug candidates. Properties relevant to lead selection include activity on the target, selectivity, drug-likeness, solubility, cytotoxicity, freedom to operate, synthetic accessibility, availability, and metabolization, as well as more specific aspects like interference with cytochrome P450 and binding to human serum albumin (Thomas, 2003).

For large compound numbers, the manual performance of bioassay tests is not feasible. There are two approaches to this problem: prioritization of compounds, leading to virtual screening, and, automated assay tests, leading to *high throughput screening* (HTS; Janzen, 2004). There, large numbers of compounds (on the order of 10^6; Schneider and Baringhaus, 2008) are tested automatically using robots, computers, handling devices, and sensors. HTS is an effective, but not an efficient process, in the sense that it provides hits, but at high costs. For this reason, it has traditionally been the domain of larger pharmaceutical companies.[3] Hit rates for HTS have been reported at 0.01–0.1 % (Sills et al., 2002) and 0.5–1 % (Assay Drug Dev. Technol., 2008).

[1] In drug development, it is also used to a lesser extent for lead optimization. Another motivation is the development of molecular tools for pharmacology, (molecular) biology, and medicine. The related disciplines of QSAR/QSPR are relevant to chemical manufacturers, pharmaceutical companies and government agencies, particularly with respect to the European Unions registration, evaluation, authorization and restriction of chemicals legislative (REACH; regulation EC 1907/2006).

[2] This is the dominant approach employed in drug development today. Other approaches, e. g., to start with a traditional medicine, to identify and to isolate the active pharmaceutical ingredient, and, to synthesize it, have their own merits (Sams-Dodd, 2005).

[3] Some universities established HTS facilities, e. g., Harvard Medical School (Boston, Massachusetts), University of California (Los Angeles, California), University of Texas Southwestern Medical Center (Dallas, Texas), McMaster University (Ontario, Canada), Rockefeller University (New York, New York).

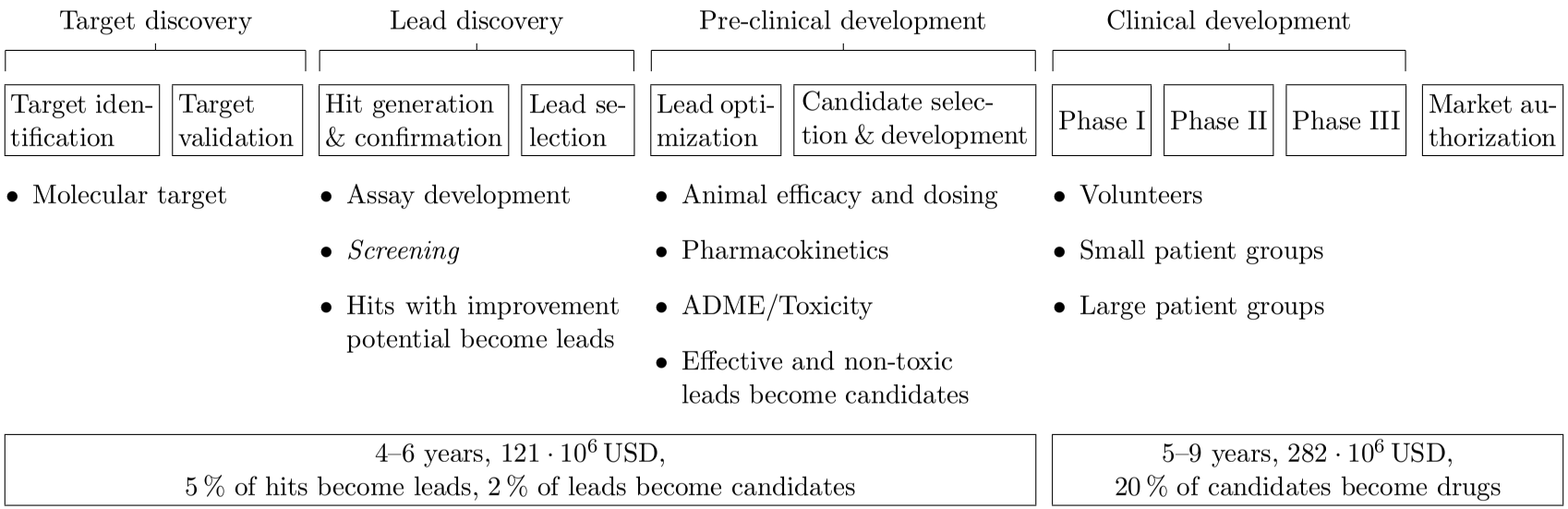


Figure 1.1 The drug development pipeline. The total average out-of-pocket cost of developing a new drug (including failures) has been estimated at $403 \cdot 10^6$ USD (year 2000 dollars; capitalization at 11 % to the time of market approval yields $802 \cdot 10^6$ USD; DiMasi et al., 2003). Another study indicates high variance in these cost estimates (Adams and Brantner, 2006). The non-governmental organization Public Citizen (2001) reports lower costs. To avoid high costs in later phases, ADME/Toxicity (p. 48) concerns are increasingly addressed in earlier phases.

1.1.2 Definition

Virtual screening has been defined as

- "automatically evaluating very large libraries of compounds [using a computer program] ... [to] decide what to synthesize." (Walters et al., 1998)
- "computational methods to prioritize the selection and testing of large chemical datasets so as to ensure that those molecules that have the largest a priori probabilities of activity are assayed first." (Willett, 2006b)
- "the computational equivalent of (experimental) high-throughput screening, wherein a large number of samples are quickly assayed to discriminate active samples from inactive samples." (Truchon and Bayly, 2007)
- "the search for the molecules within a database of compounds that match a given query." (Seifert and Lang, 2008)
- "any method that ranks a set of compounds by some score, [...] usually defined as active against a target protein." (Hawkins et al., 2008)

Some definitions stress compound numbers, some emphasize speed, others differ in the task (classification versus ranking). Hristozov et al. (2007) investigated different virtual screening scenarios: compound prioritization for HTS, compound selection for lead optimization, deciding whether a compound is active, and, identification of the most active compound. These scenarios have slightly different requirements, e. g., few false negatives, early recognition, confidence estimates, and, correlation of rank with activity.

For our purposes, we define *virtual screening* as the computational ranking of a compound data set. We do not refer to data set size or processing speed because the meaning of "large" and "fast" changes over time and quantification would be arbitrary.[4] Note that ranking can be done by scoring, and includes classification via ties.

Targets and ligands

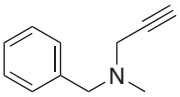
Virtual screening needs a rational starting point, either a model of the target structure, or, known ligands, leading to structure-based and ligand-based virtual screening.

Structure-based virtual screening (also *receptor-based virtual screening*; Klebe, 2006) uses structural information about the target, e. g., crystal structures (models fitted to electron densities derived from diffraction data; see Figure 4.1 for an example), structures derived by nuclear magnetic resonance spectroscopy (Wüthrich, 2003), or homology models (model interpolations based on related targets with high sequence similarity). The most prominent technique is *docking* (Kitchen et al., 2004), where the compound is placed within a binding site of the target (*pose prediction*) and the binding affinity of the resulting pose is estimated (*scoring*).[5]

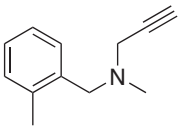
In practice, problems have been associated with structure-based virtual screening. Several frequent assumptions like correctness and relevance of the protein-ligand crystal structure are not always met (Davis et al., 2008), and scoring functions are often biased towards positive samples (Pham and Jain, 2006). In a comparison of 10 docking pro-

[4] Asymptotic worst-case runtime is no solution either: Because ligands are small, runtime is dominated by constant factors, allowing algorithms that are asymptotically slower to outperform others for the input sizes of interest.

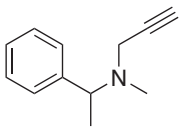
[5] Technically, docking is only the placement of the ligand in a binding pocket, and scoring is the prediction of binding affinity; most newer docking programs do both (Kroemer, 2003).



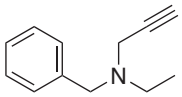
1 Pargyline



2

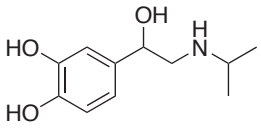


3

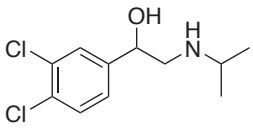


4

Pargyline (*N*-methyl-*N*-2-propynylbenzylamine), a potent irreversible inhibitor of monoamine oxidase, and three derivatives differing only by an additional methyl group ("magic methyls"). Compounds 1 and 2 are active, Compounds 3 and 4 are inactive.



5 Isoproterenol



6 Dichloro-isoproterenol

Isoproterenol is an agonist of the α-adrenergic receptor. Substitution of two hydroxy groups by chlorine yields dichloro-isoproterenol, a β-adrenergic antagonist.

Scheme 1.1 Examples of similarity principle violations (similar compounds, disparate activities; Schneider and Baringhaus, 2008). For more examples, see Kubinyi (1998).

grams and 37 scoring functions, Warren et al. (2006) found no statistically significant relationship between scores and ligand affinity.

Ligand-based virtual screening (Douguet, 2008) is based on the *similarity principle* (Johnson and Maggiora, 1990), which states that (structurally) similar compounds exhibit similar biological activities. Besides exceptions (Scheme 1.1), the validity of this assumption depends on the choice of compound similarity, i. e., the choice of molecular representation and similarity measure (Section 1.3). Quantitative investigations (Martin et al., 2002) have confirmed the similarity principle, albeit with large variance.[6]

In practice, ligand-based methods have been reported to outperform docking in virtual screening (Hawkins et al., 2006; McGaughey et al., 2007), especially with regard to false positives. However, ligand-based approaches do not use information about the target structure, even if it is available, and do not provide insight into binding pose, or mechanism of action. Table 1.1 contrasts advantages and disadvantages of structure- and ligand-based virtual screening.

Several strategies have been proposed to combine both approaches. In *hybrid virtual screening protocols* (Sperandio et al., 2008), ligand-based virtual screening hits are further investigated by docking methods. Combining the results of multiple methods, known as *data fusion* (Willett, 2006a), or *consensus scoring* in docking (Feher, 2006),[7] can lead to performance superior to that of the individual methods alone. *Pseudo-receptor* models (Tanrikulu and Schneider, 2008) are receptor surrogates for virtual screening derived from the alignment of ligand conformations. Note that combinations of structure- and ligand-based virtual screening also inherit combinations of their (dis)advantages.

[6]The frequency with which a compound similar to an active was itself active has been estimated at 0.012–0.5, 0.3, 0.4–0.6, 0.8 (Unity fingerprints, Tanimoto similarity $\geq$ 0.85), 0.43–0.7 (topological torsions, atom pair fingerprints, clustering), and 0.67 (Daylight fingerprints, Tanimoto similarity $\geq$ 0.85); all numbers as given by Martin et al. (2002). Part of the variance has been traced back to diversity issues in the used data sets.

[7]Other methods for combining multiple models exist outside of cheminformatics, e. g., forecast combination by encompassing tests in econometrics (Newbold and Harvey, 2002), where a linear combination of predictive models is sought such that no model is contained (encompassed) by another.

Table 1.1 Comparison of structure-based and ligand-based virtual screening.

Method	Advantages	Disadvantages
Structure-based	• No restriction to known ligands • Use of target information • Low false negative rate	• Requires model of target structure • High computational complexity • Scoring functions often do not consider entropy, desolvation energy, metal ions, polarization effects, ... • Protein flexibility is often not sufficiently treated to model induced fit phenomena • High false positive rate
Ligand-based	• Low computational complexity • No target information required	• Chemical diversity is limited to the known ligands • Target information is not used • 3D methods depend on conformer generation

For further information on virtual screening, see Böhm and Schneider (2000); Alvarez and Shoichet (2005). An extensive list of successful virtual screening applications is given by Kubinyi (2006); Seifert and Lang (2008).

1.2 The machine learning approach

Ligand-based virtual screening lends itself naturally to machine learning approaches, with different variants requiring different paradigms, models, and methods of learning.

Inductive machine learning (also *pattern recognition*) is the algorithmic search for patterns in data, a field closely connected to statistics, information theory, and compression.[8,9] The investigated data, called *training data*, are given in the form of *examples*, or *samples*, corresponding to the known ligands. Explicit prior information can be exploited, both in terms of the used molecular representation[10] and the sought pattern[11]. A learned pattern can be applied to new data, called the *test data*, corresponding to the screened compound library.

[8] A pattern (or regularity) in some data enables, via exploitation of the pattern, a shorter representation of this data. Vice versa, a shorter representation implies regularity. For further information on the connections between information theory, compression, and inductive learning, see MacKay (2003).

[9] Induction is different from *transduction* (also *instance-based learning*), where reasoning is directly from samples (training data) to other samples (test data), without an intermediate search for a pattern (Vapnik, 1998). An example is k-nearest neighbor classification.

[10] Consider a substructure-based model for activity on a given receptor. If there are known steric constraints, incorporation of spatial information into the description may improve predictions.

[11] See Section 3.3 for an example where a linear pattern is known a priori to be sufficient. If the type of pattern is not known in advance, universal kernels like the Gaussian kernel (p. 30) may be used.

We focus on inductive kernel-based machine learning approaches to ligand-based virtual screening. Other approaches such as deductive, i. e., logic-based, machine learning (Russell and Norvig, 2002), or, inductive methods not based on kernels such as artificial neural networks (Bishop, 1996) and decision trees (Rokach and Maimon, 2008), have been applied to ligand-based virtual screening, but are outside the scope of this work.

For further information on inductive machine learning, see the text books by Duda et al. (2001); Hastie et al. (2003); Bishop (2006).

1.2.1 Learning paradigms

Machine learning problems can be classified with regard to various aspects. Depending on data quality and availability, virtual screening scenario, and experimental setup, ligand-based virtual screening can be formulated within many of these problem settings.

Supervised, semi-supervised, and unsupervised learning

Supervised learning (Kotsiantis, 2007) is the classic inductive learning scenario, where each training sample is associated with a *label*, and the task is to infer the unknown labels of test samples. In *semi-supervised learning* (Chapelle et al., 2006), only some of the training samples have labels, and the unlabeled samples are used to improve the prediction of test sample labels. In *unsupervised learning* (Ghahramani, 2004), samples are not associated with labels. Tasks include density estimation and the discovery of structure, e. g., via clustering. Spectral dimensionality reduction methods (Section 3.2) and Gaussian processes (Subsection 4.2.4) are examples of unsupervised and supervised learning techniques, respectively.

Regression, classification, and novelty detection

In supervised learning, one differentiates by the type of label associated with samples.

In *regression*, each sample is associated with a real number. Examples are QSAR / QSPR, and, analysis of HTS data. For regression, the accuracy of the data (p. 41) is important. The latter is often surprisingly low for biochemical experimental data (Footnote 17, p. 41). Chapter 4 describes a virtual screening experiment based on Gaussian process regression.

In *classification*, only categorical information is available, i. e., the labels belong to a finite set of two or more classes. *Binary classification*, e. g., active versus inactive samples, is a special case. Classification is often used as a first approximation based on an activity cut-off, or, when combining data from different sources due to accuracy issues, e. g., if compound activity was measured in different assays, as is typical when data is collected from the scientific and patent literature. Another example is the classification of drugs versus non-drugs (Hutter, 2009).

In *novelty detection*, only samples from one class are available. The task is to decide whether new samples were drawn from the same distribution as the training data, i. e., whether they are novel or not. This scenario can be appropriate for new targets with few known ligands, especially if resources are limited or an HTS assay is not available. Another reason is publication bias: Since negative results are not published, it is often difficult to obtain data about experimentally verified inactive compounds.

Negative samples problem

The unavailability of verified negative samples is often addressed by substituting randomly sampled compounds, usually from the test samples to be screened. On the one hand, this introduces additional information in the form of (supposedly) negative samples, and, turns the novelty detection problem into a binary classification problem, for which more algorithms exist. On the other hand, this also introduces property bias (p. 39), as well as errors in the form of negative training samples which are actually active (the test set is assumed to contain actives, otherwise it would be pointless to screen it). It can also be argued that, when studying a ligand class, it is more interesting to characterize it than to separate it from an (arbitrary) fraction of chemical space (the screening library).

Together, these reasons commend the use of novelty detection approaches to ligand-based virtual screening. We introduce dimensionality reduction based on kernel principal component analysis as such an approach in Chapter 3.

Other criteria

In a *batch* setting, all training samples are available from the beginning, whereas in an *on-line* setting, samples are successively made available. In *passive learning*, training samples are given; in *active* learning, the learning algorithm may request labeling of unlabeled samples. Most machine learning applications in virtual screening take place in passive batch settings.

1.2.2 Kernel-based learning

Since their introduction in the 1990s,[12] kernel-based machine learning methods have been widely applied in both science and industry, and have become an active area of research. We introduce only the concepts required for this and the following chapters. For a recent review of kernel-based learning, see Hofmann et al. (2008); for an introductory text, see the books by Schölkopf and Smola (2002); Shawe-Taylor and Cristianini (2004).

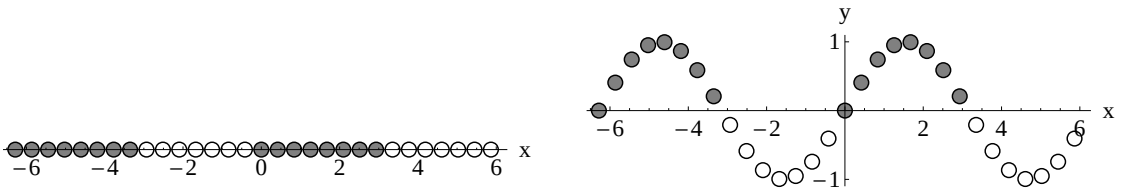
Idea

The basic idea of *kernel-based machine learning* methods is to turn linear algorithms, e. g., least squares regression (p. 157) or principal component analysis (Subsection 3.2.1), into non-linear algorithms in a systematic way. This is done by (implicitly) mapping the input samples into a higher-dimensional space, and applying the linear algorithm there (Figure 1.2). This approach has two immediate problems: Computational complexity, and how to find the right mapping.

The kernel trick

Consider the mapping $\phi : \mathbb{R}^p \to \mathbb{R}^{p^d}$ which maps $\boldsymbol{x}$ to the space of all ordered monomials of degree d, e. g., for $p = d = 2$, $\phi\big((\boldsymbol{x}_1, \boldsymbol{x}_2)\big) = (\boldsymbol{x}_1^2, \boldsymbol{x}_1\boldsymbol{x}_2, \boldsymbol{x}_2\boldsymbol{x}_1, \boldsymbol{x}_2^2)$. The size p^d of the space mapped into (the *feature space*) depends polynomially on the size p of the input space. For a typical input space dimensionality of $p = 120$ (Table 1.3a) and $d = 3$, the size of the target space is already $120^3 = 1\,728\,000$. Other mappings are into feature

[12] Support vector machines (Subsection 2.4.3), the first widely successful kernel algorithm, were introduced by Boser et al. (1992). The involved concepts had been investigated since the 1960s, e. g., the first use of kernels in machine learning by Aizerman et al. (1964). See Cristianini and Shawe-Taylor (2000).



(a) Linear inseparability in input space. **(b)** Linear separability in transformed space.

Figure 1.2 Linear separability via a non-linear mapping into a higher-dimensional space. In the input space $\mathbb{R}$, samples from the two classes (blank and grey disks) are not linearly separable. The non-linear function $x \mapsto (x, \sin x)$ maps the samples into the higher-dimensional space $\mathbb{R}^2$, where samples are linearly separable (by the x-axis).

spaces of infinite dimension. For these mappings, explicit computations in feature space are computationally either infeasible or impossible.

The *kernel trick* is to replace computations in feature space by computations in input space that give the same results. This is achieved by the use of *inner products*, which generalize geometric concepts like length, angle, and orthogonality. For a real vector space $\mathcal{X}$, a function $\langle \cdot, \cdot \rangle : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is an inner product iff for all $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathcal{X}$, $\alpha \in \mathbb{R}$ holds

- $\langle \boldsymbol{x}, \boldsymbol{x} \rangle \geq 0$ (non-negativity) and $\langle \boldsymbol{x}, \boldsymbol{x} \rangle = 0 \Leftrightarrow \boldsymbol{x} = 0$,
- $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \langle \boldsymbol{y}, \boldsymbol{x} \rangle$ (symmetry),
- $\langle \boldsymbol{x} + \boldsymbol{y}, \boldsymbol{z} \rangle = \langle \boldsymbol{x}, \boldsymbol{z} \rangle + \langle \boldsymbol{y}, \boldsymbol{z} \rangle$ and $\langle \alpha \boldsymbol{x}, \boldsymbol{y} \rangle = \alpha \langle \boldsymbol{x}, \boldsymbol{y} \rangle$ (linearity).

The pair $(\mathcal{X}, \langle \cdot, \cdot \rangle)$ is called an *inner product space*. Two vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$ are *orthogonal* iff their inner product is zero, $\boldsymbol{x} \perp \boldsymbol{y} \Leftrightarrow \langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0$. In a real inner product space $\mathcal{X}$, the *angle* (measured in radians) between two non-zero $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$ is defined as

$$\theta \in [0, \pi] \quad \text{such that} \quad \cos\theta = \frac{\langle \boldsymbol{x}, \boldsymbol{y} \rangle}{\|\boldsymbol{x}\| \, \|\boldsymbol{y}\|}. \tag{1.1}$$

An inner product $\langle \cdot, \cdot \rangle$ can be used to construct a norm (and corresponding metric) via $\|\boldsymbol{x}\| = \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}$. Conversely, given a norm $\|\cdot\|$ on $\mathcal{X}$, $\|\boldsymbol{x}\|^2 = \langle \boldsymbol{x}, \boldsymbol{x} \rangle$ iff the *parallelogram identity* $\|\boldsymbol{x} + \boldsymbol{y}\|^2 + \|\boldsymbol{x} - \boldsymbol{y}\|^2 = 2(\|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2)$ holds; the Euclidean norm is the only L^p-norm satisfying this identity (Meyer, 2001).

Many machine learning algorithms can be expressed in terms of the lengths, angles, and distances between input samples; in other words, they can be rewritten to use only inner products of input samples. For a worked-through example of how to turn principal component analysis, a linear algorithm, into its kernel variant, see Chapter 3.

Kernels

A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a *kernel* iff there exists a map $\phi : \mathcal{X} \to \mathcal{H}$ such that

$$\forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{X} : k(\boldsymbol{x}, \boldsymbol{y}) = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{y}) \rangle , \tag{1.2}$$

i. e., if it corresponds to an inner product in some feature space $\mathcal{H}$. It is not necessary to know ϕ or $\mathcal{H}$ explicitly, their existence is sufficient. Using a kernel, one can implicitly compute inner products in high-dimensional feature spaces by computing kernel values in input space, thereby alleviating the computational complexity issue due to feature space dimensionality.

As an example, let $\phi : \mathbb{R}^p \to \mathbb{R}^{p^d}$ denote the map to the space of ordered monomials as before, and consider the homogeneous polynomial kernel $k = \langle \boldsymbol{x}, \boldsymbol{y} \rangle^d$ of degree d:

$$\begin{aligned} k(\boldsymbol{x}, \boldsymbol{y}) = \langle \boldsymbol{x}, \boldsymbol{y} \rangle^d = \left(\sum_{i=1}^{p} \boldsymbol{x}_i \boldsymbol{y}_i \right)^d = \sum_{i_1=1}^{p} \boldsymbol{x}_{i_1} \boldsymbol{y}_{i_1} \sum_{i_2=1}^{p} \boldsymbol{x}_{i_2} \boldsymbol{y}_{i_2} \cdots \sum_{i_d=1}^{p} \boldsymbol{x}_{i_d} \boldsymbol{y}_{i_d} \\ = \sum_{i_1=1}^{p} \sum_{i_2=1}^{p} \cdots \sum_{i_d=1}^{p} \boldsymbol{x}_{i_1} \boldsymbol{x}_{i_2} \cdots \boldsymbol{x}_{i_d} \, \boldsymbol{y}_{i_1} \boldsymbol{y}_{i_2} \cdots \boldsymbol{y}_{i_d} = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{y}) \rangle \, . \end{aligned} \tag{1.3}$$

A kernel algorithm is only allowed to use inner products of input samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$. The matrix $\boldsymbol{K} \in \mathbb{R}^{n \times n}$, $\boldsymbol{K}_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is called the *kernel matrix* (also *Gram matrix*). It contains all the information about the input samples accessible to the algorithm.

Positive definiteness

A symmetric matrix $\boldsymbol{K} \in \mathbb{R}^{n \times n}$ is *positive definite* iff

$$\forall \boldsymbol{c} \in \mathbb{R}^n : \boldsymbol{c}^T \boldsymbol{K} \boldsymbol{c} = \sum_{i,j=1}^{n} \boldsymbol{c}_i \boldsymbol{c}_j \boldsymbol{K}_{i,j} \geq 0. \tag{1.4}$$

$\boldsymbol{K}$ is *strictly positive definite* iff equality occurs only for $\boldsymbol{c} = 0$.[13] A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that has (strictly) positive definite Gram matrix for all $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$, $n \in \mathbb{N}$ is called (strictly) positive definite. Inner products are positive definite due to

$$\sum_{i,j=1}^{n} \boldsymbol{c}_i \boldsymbol{c}_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left\langle \sum_{i=1}^{n} \boldsymbol{c}_i \phi(\boldsymbol{x}_i), \sum_{j=1}^{n} \boldsymbol{c}_j \phi(\boldsymbol{x}_j) \right\rangle \geq 0. \tag{1.5}$$

Vice versa, it can be shown that every positive definite function corresponds to an inner product in some inner product space (via reproducing kernel Hilbert spaces and the Moore-Aronszajn theorem, Aronszajn, 1950). Proper kernels are therefore characterized by the positive definiteness property.[14]

Criteria for the positive definiteness of matrices other than Equation 1.4 include Sylvester's criterion,[15] and, the eigenspectrum of a matrix: $\boldsymbol{K}$ is (strictly) positive definite iff all of its eigenvalues are non-negative (positive).

The Gaussian kernel

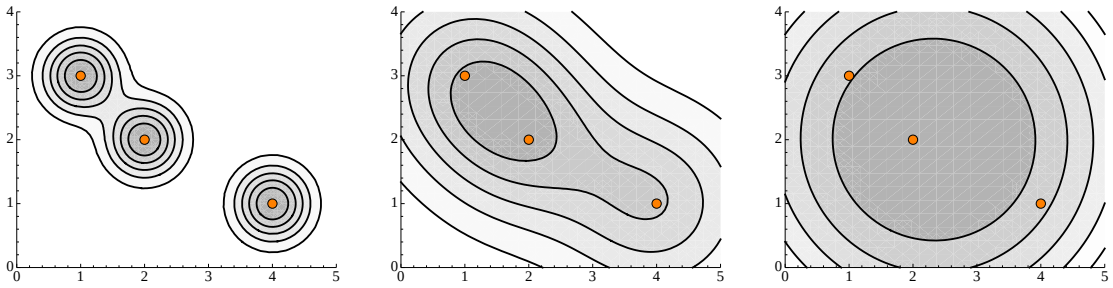
The kernel has to transform the input data in a way that allows successful application of a linear algorithm in feature space. There are two basic approaches to this: Problem domain-specific kernels and generic kernels. Chapter 2 describes a kernel on molecular structure graphs developed for ligand-based virtual screening. A good default choice on vectorial input data such as molecular descriptors is the *Gaussian kernel* (also *radial basis function kernel*, RBF)

$$k(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{||\boldsymbol{x} - \boldsymbol{y}||^2}{2\sigma^2} \right), \tag{1.6}$$

[13] Positive definite and strictly positive definite matrices are also called *positive semidefinite* and *positive definite* matrices, respectively; corresponding care has to be taken when consulting the literature.

[14] Historically, kernels satisfying the theorem of Mercer (1909) were used. Such functions correspond to inner products, but not all functions corresponding to inner products satisfy the theorem's conditions.

[15] $\boldsymbol{K}$ is strictly positive definite iff its leading principal minors are positive; it is positive semidefinite iff all of its principal minors are non-negative (Kerr, 1990).



(a) $\sigma = 0.4$. **(b)** $\sigma = 1$. **(c)** $\sigma = 10$.

Figure 1.3 The behavior of the Gaussian kernel. Shown are 5 isolines (having different values in the three plots; black lines) of a Gaussian mixture model $g(\boldsymbol{x}) = \frac{1}{3}\sum_{i=1}^{3} \mathrm{rbf}(\boldsymbol{x}_i, \boldsymbol{x})$ for three data points $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, $\boldsymbol{x}_3$ (orange disks), where rbf indicates the kernel function from Equation 1.6. Graylevels indicate the density of g, with darker values corresponding to higher densities.

where $\sigma > 0$ is the kernel width. This kernel maps into an infinite-dimensional feature space (Steinwart et al., 2006).

To understand the behavior of the Gaussian kernel, consider the limiting cases of the kernel width. For $\sigma \to \infty$, the kernel matrix becomes the all-ones matrix $\mathbf{1}_{n\times n}$, i. e., all samples are mapped into a single point, leading to underfitting. For $\sigma \to 0$, the kernel matrix becomes the identity matrix $\boldsymbol{I}_{n\times n}$, i. e., all samples are mapped into different dimensions orthogonal to each other, leading to overfitting. For intermediate values of σ, the kernel value depends on $\|\boldsymbol{x} - \boldsymbol{y}\|$, approaching 1 for $\|\boldsymbol{x} - \boldsymbol{y}\| \to 0$, and 0 for $\|\boldsymbol{x} - \boldsymbol{y}\| \to \infty$. Samples that are close in input space are therefore correlated in feature space, whereas faraway samples are mapped to orthogonal subspaces. In this way, the Gaussian kernel can be seen as a local approximator, with scale dependent on σ (Figure 1.3).

Remarks

The input space $\mathcal{X}$ can be any non-empty set with sufficient structure to allow the definition of a kernel; in particular, $\mathcal{X}$ does not have to be a vector space (for kernels on structured data, see p. 59). In contrast to other methods like neural networks, kernel methods are often deterministic and globally optimal with respect to the optimization problem they solve, e. g., kernel principal component analysis (Subsection 3.2.2) and support vector machines (Subsection 2.4.3). The property of positive definiteness, which guarantees the existence of an inner product feature space, is often also the property required to ensure convexity of these optimization problems.

Examples

In Chapter 3, an unsupervised kernel-based machine learning algorithm for dimensionality reduction, kernel principal component analysis, is formally derived and used for visualization and novelty detection. In Chapter 4, a kernel-based regression algorithm, Gaussian process regression, is applied in a prospective virtual screening study.

1.3 Representation and similarity of molecules

Success in virtual screening depends primarily on how well the used molecular representation and (dis)similarity measure capture characteristics relevant to the target. If crucial information is not encoded in the input representation, all methods fail, whereas simple methods such as similarity search (Willett, 1998) can perform well with the right representation.

1.3.1 Molecular representations

The abundance of available molecular representations — the handbook of molecular descriptors (Todeschini and Consonni, 2000) lists more than 1 600 of them — is caused by the necessity to selectively model molecular properties relevant to the specific target interaction under investigation. This necessity originates from the computational demands of first principles methods, which make them infeasible in large scale applications. The selection of a molecular representation for a specific task is a problem in itself.

Limits of quantum theoretical representations

Quantum mechanics (Cohen-Tannoudji et al., 2006) is concerned with matter and energy on an atomic scale, and constitutes the most fundamental and accurate theory available to date. Its application to chemistry, *quantum chemistry* (McQuarrie, 2007), provides a theoretical foundation for the description of molecular interactions, e. g., between ligand and receptor.

In quantum mechanics, a system is completely described by its wave function (and the evolution of it over time). Theoretically, no other molecular representation is needed. In practice, the underlying Schrödinger equation, essentially a many-body problem, can be solved analytically only for the hydrogen atom. Numerical (ab-initio) solutions are presently limited to a few dozen electrons for computational reasons, i. e., a computational complexity exponential in the number of electrons (Friesner, 2005).

Necessity of different representations

Virtual screening involves large numbers of compounds, and therefore requires methods of low computational complexity. This severely limits the applicability of exact quantum mechanical representations and necessitates more abstract representations, i. e., representations that contain only the information relevant to the target. Such representations are computationally less complex to obtain, since only specific molecular characteristics have to be computed; for the same reasons, they are also specific to targets, or target classes. Along with the number of targets — Overington et al. (2006) identified 324 targets of approved therapeutic drugs alone — this is one cause for the abundance of available molecular representations.

Descriptors

Different classification systems for molecular representations are in use, e. g., by dimensionality (sometimes of the representation, sometimes of the property; e. g., line notations and molecular weight have both been called one-dimensional), by the used data structure (e. g., string, vector, graph), or, by the property described (e. g., shape, charge, connectivity).

A popular category of molecular representations are chemical *descriptors* (Todeschini and Consonni, 2000), which describe molecules with numerical attributes, either experimentally determined or computed ones. The corresponding mathematical abstraction are vector spaces (Meyer, 2001), resulting in chemical descriptor spaces (Subsection 1.3.2). Vectorial representations have several advantages: They are widely used across disciplines, vector spaces are mathematically well developed, and many algorithms are available. For these reasons, vector representations were used in the first applications of machine learning methods in ligand-based virtual screening. Later, structured data representations such as graphs were used.

Spatial information

Many descriptors depend only on a molecule's constitution (e. g., molecular weight, number of rotatable bonds) or its topology (e. g., graph diameter, connectivity indices). The incorporation of spatial aspects should lead to more realistic models, improving performance. Empirical investigations, however, showed little or no advantage of 3D descriptors over topological ones (Hristozov et al., 2007).

In part, this has been ascribed to inductive bias, i. e., a bias towards 2D similarity in existing data sets, caused by a corresponding bias in human experts who created them and extensive use of 2D similarity methods in their creation (Cleves and Jain, 2008).

Conformational isomers (also *conformers*) are stereoisomers (molecules with identical constitution, but different arrangement in space) due to rotation around σ-bonds (single bonds; Moss, 1996). In principle, the performance of ligand-based 3D virtual screening methods depends on the used conformations of the training compounds. In empirical investigations, however, source and number of conformations had little effect on performance (Hristozov et al., 2007; McGaughey et al., 2007).

Measures of (dis)similarity

Besides the molecular representation itself, the applicability of the similarity principle depends on the measure of (dis)similarity used to compare two such representations. Different measures are available, and their choice influences virtual screening results (Fechner and Schneider, 2004; Rupp et al., 2009). Many measures of (dis)similarity exist; the most common categories are norms, which measure length, metrics, which measure distance, inner products, which encode information about length, angle, and orthogonality, and, similarity coefficients, which measure similarity, but lack some formal properties of the others. See Table 1.2 for examples and Meyer (2001) for details.

1.3.2 Distance phenomena in high-dimensional descriptor spaces

Real-valued chemical descriptor spaces are widely used in ligand-based virtual screening and QSAR modeling. Although a successful concept, some problems have been associated with high-dimensional descriptor spaces, e. g., chance correlations in QSAR (Topliss and Edwards, 1979; Mager, 1982). Other problems, sometimes summarized under the umbrella term *curse of dimensionality* (coined by Bellman, 1957), have been recognized both within (Willett et al., 1998) and outside (François, 2007) of chemistry, e. g., in the database community, where they are relevant to indexing and retrieval. The root cause of these phenomena is that distance is measured across volume, which increases exponentially with dimension.

Table 1.2 Common norms, metrics, inner products, and similarity coefficients. For Minkowski distances with $0 \leq p < 1$, the triangle inequality is reversed. The Frobenius matrix norm is the L^2 norm applied to the concatenated rows or columns of a matrix $\boldsymbol{A}$. Vectors are over the domain $\mathbb{R}^m$; for matrices, $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{k \times m}$, and $\boldsymbol{M} \in \mathbb{R}^{m \times m}$. Similarity coefficients have range $[-1, 1]$, except for the Tanimoto coefficient, which has range $[-\frac{1}{3}, 1]$. $\boldsymbol{A}^T$ = transpose of matrix $\boldsymbol{A}$, $\operatorname{tr}(\boldsymbol{A})$ = trace of matrix $\boldsymbol{A}$, $\operatorname{covar}(\boldsymbol{x}, \boldsymbol{y})$ = (empirical) covariance of $\boldsymbol{x}$ and $\boldsymbol{y}$, $\operatorname{var}(\boldsymbol{x})$ = (empirical) variance of $\boldsymbol{x}$, s. p. d. = symmetric positive definite.

Formula	Name
Norms	
$\left(\sum_{i=1}^m \lvert\boldsymbol{x}_i\rvert^p\right)^{1/p}$, $p \geq 1$	L^p norm $\lVert\cdot\rVert_p$
$\sum_{i=1}^m \lvert\boldsymbol{x}_i\rvert$	L^1 norm $\lVert\cdot\rVert_1$, grid norm, sum norm
$\sqrt{\sum_{i=1}^m \lvert\boldsymbol{x}_i\rvert^2}$	L^2 norm $\lVert\cdot\rVert_2$, Euclidean norm
$\max_{1 \leq i \leq m} \lvert\boldsymbol{x}_i\rvert$	L^∞ norm $\lVert\cdot\rVert_\infty$, max norm
$\sqrt{\sum_{i=1}^k \sum_{j=1}^m \lvert a_{i,j}\rvert^2} = \sqrt{\operatorname{tr}(\boldsymbol{A}\boldsymbol{A}^T)}$	Frobenius norm
$\max_{\lVert\boldsymbol{x}\rVert=1} \lVert A\boldsymbol{x}\rVert$	Matrix norm induced by $\lVert\cdot\rVert$
Metrics	
$\left(\sum_{i=1}^m \lvert\boldsymbol{x}_i - \boldsymbol{y}_i\rvert^p\right)^{1/p}$, $p \geq 1$	L^p norm induced metric, Minkowski distance
$\sum_{i=1}^m \lvert\boldsymbol{x}_i - \boldsymbol{y}_i\rvert$	Manhattan metric (L^1 norm-based)
$\sqrt{\sum_{i=1}^m \lvert\boldsymbol{x}_i - \boldsymbol{y}_i\rvert^2}$	Euclidean metric (L^2 norm-based)
$\max_{1 \leq i \leq m} \lvert\boldsymbol{x}_i - \boldsymbol{y}_i\rvert$	Maximum (Chebyshev) metric (L^∞ norm-based)
$\sqrt{(\boldsymbol{x}-\boldsymbol{y})^T \boldsymbol{M} (\boldsymbol{x}-\boldsymbol{y})}$, $\boldsymbol{M}$ s. p. d.	Mahalanobis metric
Inner products	
$\boldsymbol{x}^T\boldsymbol{y} = \sum_{i=1}^m \boldsymbol{x}_i\boldsymbol{y}_i$	Standard inner product, dot product
$\operatorname{tr}(\boldsymbol{A}^T\boldsymbol{B})$	Matrix standard inner product
$\boldsymbol{x}^T\boldsymbol{M}\boldsymbol{y}$, $\boldsymbol{M}$ s. p. d.	Weighted inner product
$\int_a^b f(t)g(t)\,\mathrm{d}t$	Inner product of continuous functions on $[a, b]$
$\int_{\mathcal{X}} f(t)g(t)\,\mathrm{d}t$	Inner product of square integrable functions
Similarity coefficients	
$\frac{\operatorname{covar}(\boldsymbol{x},\boldsymbol{y})}{\sqrt{\operatorname{var}(\boldsymbol{x})\operatorname{var}(\boldsymbol{y})}}$	Product-moment (Pearsons) correlation coefficient
$\frac{2\langle\boldsymbol{x},\boldsymbol{y}\rangle}{\langle\boldsymbol{x},\boldsymbol{x}\rangle+\langle\boldsymbol{y},\boldsymbol{y}\rangle}$	Hodgkin index, Dice coefficient
$\frac{\langle\boldsymbol{x},\boldsymbol{y}\rangle}{\langle\boldsymbol{x},\boldsymbol{x}\rangle+\langle\boldsymbol{y},\boldsymbol{y}\rangle-\langle\boldsymbol{x},\boldsymbol{y}\rangle}$	Tanimoto coefficient
$\frac{\langle\boldsymbol{x},\boldsymbol{y}\rangle}{\sqrt{\langle\boldsymbol{x},\boldsymbol{x}\rangle\langle\boldsymbol{y},\boldsymbol{y}\rangle}} = \frac{\langle\boldsymbol{x},\boldsymbol{y}\rangle}{\lVert\boldsymbol{x}\rVert_2 \lVert\boldsymbol{y}\rVert_2}$	Carbó index, cosine similarity

Table 1.3 Descriptor dimensionalities and data set sizes.

(a) Dimensionality d of common descriptor spaces.

d	Descriptor
~ 50	Mini-fingerprints (Xue et al., 1999, 2000)
72	VolSurf descriptor (Cruciani et al., 2000)
120	Ghose-Crippen fragment descriptors (Viswanadhan et al., 1989)
150	CATS2D pharmacophore descriptor (p. 154)
184	MOE 2D descriptors (p. 156)

(b) Typical data set sizes n with maximum covered dimension $\text{maxd} = \lfloor \log_2(n) \rfloor$.

n	maxd	Description
10^2	6	Virtual screening training set
10^4	13	`COBRA` drug database (p. 88)
10^5	16	Known drugs
10^6	19	High-throughput screening data set
10^7	23	CAS REGISTRY database (Weisgerber, 1997)

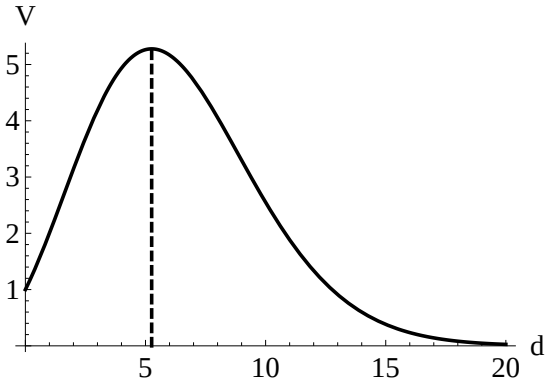
Empty space phenomenon

Consider a finite sample $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$. A partitioning of each dimension into two parts, such that each part contains at least one sample, results in a partitioning of $\mathbb{R}^d$ into 2^d compartments. Since the number of compartments grows exponentially with the dimension d, an exponential number of samples is needed to (binary) *cover* $\mathbb{R}^d$ in the sense that each compartment contains at least one sample. This is a reason why density estimation in high dimensions is difficult (Scott and Thompson, 1983).

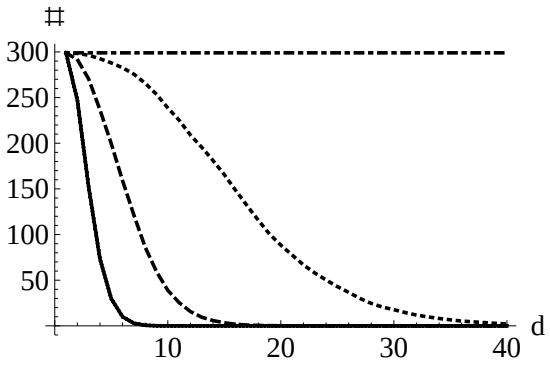
For practical scenarios, almost all of the compartments will be empty. As an example, consider a compound library with 10^8 compounds described by Ghose-Crippen fragment descriptors, for which $d = 120$, a common dimensionality of chemical descriptor spaces (Table 1.3a). Although the data set is large (Table 1.3b), the fraction of compartments covered is at most $10^8/2^{120} \approx 10^{-28} \approx 0$. The maximum dimension that could be covered by this data set is $\lfloor \log_2(10^8) \rfloor = 26$. From Table 1.3, it is clear that in typical scenarios the chemical space spanned by a descriptor will be empty in terms of data set coverage.

The distribution of the samples is another matter. Compound collections usually exhibit structure due to selection bias, which suggests that they lie on lower-dimensional manifolds in descriptor space.

Consider $n \leq 2^d$ samples drawn independently and uniformly distributed from $[0,1]^d \subset \mathbb{R}^d$, where each dimension is partitioned into intervals $[0, \frac{1}{2}]$ and $(\frac{1}{2}, 1]$. The probability that at least one compartment is shared by two or more samples is $1 - \binom{m}{n} \frac{n!}{m^n}$, where $m = 2^d$. For the `COBRA` data set (p. 88) and the CATS2D descriptor (p. 154; $n = 9\,705$, $d = 141$ after removal of constant components), this is $\approx 1.689 \cdot 10^{-35} \approx 0$. Rescaling this data set to the range $[0,1]^{141}$, however, leads to 1 016 compartments with two samples or more. Values for MOE 2D descriptors (p. 156; $n = 9\,950$, $d = 165$ after removal of constant components) are comparable ($1.058 \cdot 10^{-42}$, 1 072). This statistical test shows that the `COBRA` compounds were not sampled uniformly and identically distributed from either the CATS2D or MOE 2D descriptor spaces.



(a) Volume $V_{S_{d,1}}$ of the unit sphere as a function of the dimension d. The maximum is at ≈ 5.257, and, $V_{S_{5,1}} = \frac{8}{15}\pi$. V = volume, d = dimension.



(b) Average number of samples included in a unit sphere centered on each of 300 random samples in $[0,1]^d$. Shown are L^p norm-based metrics for $p=1$ (solid line), $p=2$ (dashed line), $p=4$ (dotted line); in the limit $p \to \infty$ (dash-dotted line), the unit sphere becomes a cube. # = number of samples, d = dimension.

Figure 1.4 The dependence of sphere volume on dimension. Because sphere volume vanishes with increasing dimension (a), spherical neighborhoods of fixed radius contain less and less neighbors (b).

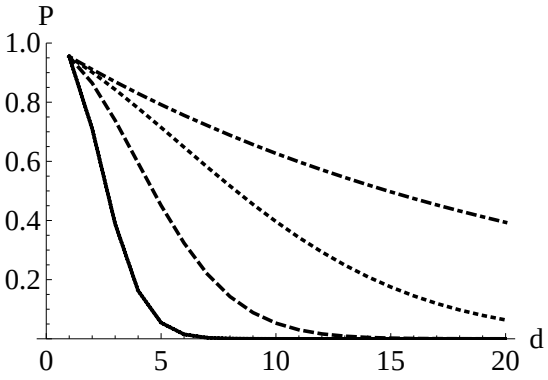
Another way to look at this is that n samples can span (if the embedding space allows it) a subspace of dimension at most n, but they can only cover a subspace of dimension $\lfloor \log_2(n) \rfloor$. These findings suggest the usefulness of feature selection (Guyon and Elisseeff, 2003) and dimensionality reduction (Fodor, 2002) for chemical data sets. Indeed, feature selection is common practice in quantitative structure-activity relationship modeling.

Sphere volumes

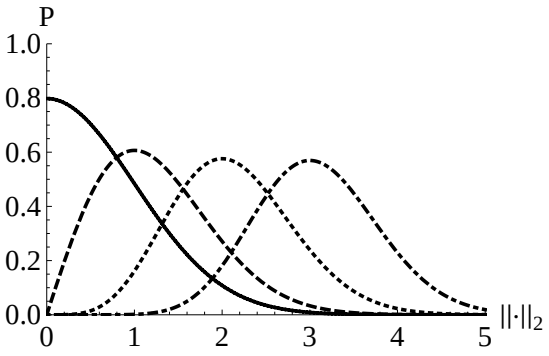
The d-dimensional *Euclidean sphere* $S_{d,r} = \left\{ \boldsymbol{x} \;\middle|\; \|\boldsymbol{x}\|_2 = r \right\}$ of radius $r \geq 0$ has volume $V_{S_{d,r}} = (\pi^{d/2} r^d)/\Gamma(1 + \frac{d}{2})$, where Γ denotes the gamma function (Hamming, 1980). $V_{S_{d,r}}$ goes to zero for $d \to \infty$ (Figure 1.4a). As a consequence, for a finite sample $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$ and fixed radius r, there is a dimension d after which a sphere of radius r centered on $\boldsymbol{x}_i$ contains only $\boldsymbol{x}_i$ and no other sample (Figure 1.4b).

The unit cube $\{\boldsymbol{x} \mid -1 \leq \boldsymbol{x}_i \leq 1\} = \{\boldsymbol{x} \mid \|\boldsymbol{x}\|_\infty \leq 1\}$ (the unit sphere as measured by the max norm) circumscribes the Euclidean unit sphere; its volume 2^d goes to infinity. Therefore, for $d \to \infty$ a sample drawn uniformly from this cube is almost surely located at its corners, i. e., in the cube, but outside of the sphere. For two spheres with radii $r < r'$, the ratio of their volumes $V_{S_{d,r}}/V_{S_{d,r'}} = \left(\frac{r}{r'}\right)^d$ decreases exponentially with d. Samples drawn uniformly from the larger sphere will therefore lie outside of the smaller sphere with probability $1 - (\frac{r}{r'})^d \xrightarrow{d\to\infty} 1$.

From the previous considerations, it is clear that both norm and dimension should be considered when choosing radii in spherical neighborhood computations, e. g., k-nearest neighbor classification. In this work, we mainly use the Euclidean distance due to its close connection with inner products (p. 29), while the dimension d is given by the used descriptor space. The radius r can be determined by solving an optimization problem (Balasubramanian, 2002).



(a) Probability mass of the d-dimensional standard normal distribution contained in a sphere of radius 2, measured by the grid norm (solid line), the Euclidean norm (dashed line), the L^4 norm (dotted line), and, the max norm (dash-dotted line). Numerical estimation with 10^6 samples.



(b) Distribution of the Euclidean norm of samples drawn from a standard normal distribution in d dimensions. The L^2 norm follows a χ-distribution with d degrees of freedom. Shown are probability densities for $d = 1$ (solid line), $d = 2$ (dashed line), $d = 5$ (dotted line), and, $d = 10$ (dash-dotted line).

Figure 1.5 The multivariate standard normal distribution and dimensionality. With increasing dimension, less probability mass is found within a fixed radius around the mean (a), and samples center on a sphere surface (b). P = probability, d = dimension.

The normal distribution

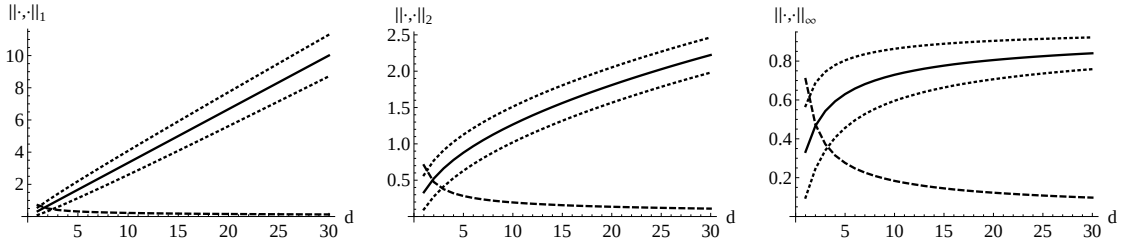
Phenomena related to spherical or ellipsoidal volumes also affect statistics. Consider a d-dimensional standard normal distribution, i. e., a distribution that generates samples $\boldsymbol{x}_i \in \mathbb{R}^d$ with independent components $(\boldsymbol{x}_i)_j \sim \mathcal{N}(0, 1)$. With increasing dimension, the probability mass contained in a sphere of fixed radius around the origin decreases rapidly (Figure 1.5a). In one dimension most points lie close to the origin, while in higher dimensions almost no point does; in this sense, for high dimensions most of the probability mass lies in the tails and not in the center of a normal distribution.

This behavior is due to the L^p norms definition in terms of absolute component values $|(\boldsymbol{x}_i)_j|$, with $\mathrm{E}(|(\boldsymbol{x}_i)_j|) = \sqrt{2/\pi}$ causing $\|\boldsymbol{x}_i\|_p$ to grow with each added dimension. Since the distribution of $\|\boldsymbol{x}_i\|_p$ is unimodal, the samples tend to lie on a hypersphere with radius $r = \mathrm{E}(\|\boldsymbol{x}_i\|_p)$ (Figure 1.5b). The value of r depends on d and p: For the grid norm, $\mathrm{E}(\|\boldsymbol{x}_i\|_1) = d\sqrt{2/\pi}$; for the Euclidean norm, $\mathrm{E}(\|\boldsymbol{x}_i\|_2) = \sqrt{2}\,\Gamma(\frac{d+1}{2})/\Gamma(\frac{d}{2})$.

Distance concentration

The concentration of norms is not limited to normally distributed samples; it also affects the distances between samples. In high dimensional spaces, under mild assumptions, sample norms tend to concentrate. As a consequence, all distances are similar, samples lie on a hypersphere, and each sample is nearest neighbor of all other samples.

For an intuitive explanation, consider independent samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ drawn uniformly from $[0, 1] \subset \mathbb{R}$. For $n \to \infty$, $\mathrm{E}(\boldsymbol{x}_i) = \frac{1}{2}$ because the values average out over the samples. Now consider a single sample $\boldsymbol{x} \in [0, 1]^d$ for $d \to \infty$. Again, the values average out, but this time over the components of x, so $\lim_{d\to\infty} \frac{\|\boldsymbol{x}\|_1}{d} = \frac{1}{2}$. Note that L^p norms increase with d. Figure 1.6 illustrates this for different norm-induced distances.



(a) Manhattan distance. **(b)** Euclidean distance. **(c)** Chebyshev distance.

Figure 1.6 Behavior of L^p norm-based distances. For $n = 10^5$ distances between points sampled uniformly and independently from $[0, 1]^d$, the mean (solid line) $\pm$ its standard deviation (dotted lines), and, the coefficient of variation (dashed line) are shown.

The concentration of L^p norms and associated Minkowski metrics has been formally studied (Beyer et al., 1999; Hinneburg et al., 2000; Aggarwal et al., 2001; François et al., 2007), often using the (*absolute*) *contrast* $\max_i \|\boldsymbol{x}_i\| - \min_i \|\boldsymbol{x}_i\|$ and the *relative contrast*

$$\frac{\max_i \|\boldsymbol{x}_i\| - \min_i \|\boldsymbol{x}_i\|}{\min_i \|\boldsymbol{x}_i\|} \tag{1.7}$$

as measures of concentration. However, these depend on extremal values, and therefore on sample size, and are highly volatile. Instead, we use another measure of spread versus location, the *variation coefficient* (also *relative variance*; François et al., 2007)

$$\frac{\sigma}{\mu} = \frac{\sqrt{\mathrm{var}(\|\cdot,\cdot\|)}}{\mathrm{E}(\|\cdot,\cdot\|)}, \tag{1.8}$$

where small values indicate concentration. Note that $\frac{\sigma}{\mu}$ can equivalently be defined in terms of norms by a change of domain (François et al., 2007). Table 1.4 lists empirical variation coefficients computed on the `COBRA` data set.

Let $\boldsymbol{X} \in \mathbb{R}^d$ be a random variable with i. i. d. components. Then

$$\lim_{d\to\infty} \frac{\mathrm{E}\big(\|\boldsymbol{X}\|_p\big)}{d^{\frac{1}{p}}} = c, \qquad \lim_{d\to\infty} \frac{\mathrm{var}\big(\|\boldsymbol{X}\|_p\big)}{d^{\frac{2}{p}-1}} = c', \quad \text{and,} \quad \lim_{d\to\infty} \frac{\sqrt{\mathrm{var}\big(\|\boldsymbol{X}\|_p\big)}}{\mathrm{E}\big(\|\boldsymbol{X}\|_p\big)} = 0, \tag{1.9}$$

where c and c' are constants not depending on d (François et al., 2007). This shows that, under the strong assumption of independence and identical distribution of the components, all L^p norms and Minkowski distances concentrate, and also gives the rates of growth (Figure 1.6) as

$$\mathrm{E}\Big(\|\boldsymbol{X}\|_p\Big) \sim c\, d^{\frac{1}{p}} \qquad \text{and} \qquad \mathrm{var}\Big(\|\boldsymbol{X}\|_p\Big) \sim c'\, d^{\frac{2}{p}-1}, \tag{1.10}$$

where c and c' depend on p.

Equation 1.9 stays valid for differently distributed components and dependences between them (François et al., 2007). In the first case, the equation still holds if the data are standardized, i. e., if they have zero mean and unit variance (subtracting the mean and dividing by the standard deviation achieves this). Standardization ensures that the norm is not dominated by a few components. In the second case, concentration takes place, but depends on the intrinsic dimensionality of the data, as opposed to the dimensionality of the vector space itself.

Table 1.4 Variation coefficients for different descriptors and (dis)similarity measures on the COBRA data set. r = Pearsons correlation, d = Dice coefficient, t = Tanimoto coefficient, c = Carbó index.

	Minkowski metrics (p)					(Dis)similarity coefficients			
Descriptor	1	2	3	5	∞	$1-r$	$1-d$	$1-t$	$1-c$
CATS2D	0.30	0.21	0.19	0.18	0.21	0.34	0.35	0.25	0.36
MOE 2D	0.42	0.38	0.37	0.40	0.49	0.29	0.30	0.17	0.33

Chemical descriptor spaces are, as a rule, normalized in some form or other, and from Table 1.3a, as well as Figures 1.4, 1.5, and 1.6, it is evident that their dimensionality is high enough for distance concentration to occur. However, due to dependencies between descriptors, various forms of bias (p. 39), and reasons given before (p. 36), the intrinsic dimensionality of chemical data sets will be lower than the dimensionality of the embedding descriptor space. For a more detailed treatment of these phenomena, and their consequences for virtual screening, see Rupp et al. (2009).

1.4 Retrospective evaluation

The evaluation of virtual screening methods on known data, called *retrospective evaluation*, serves two purposes: selection of the best model for a given target, and, estimation of a given models hit rate on a target, with the focus usually on the former.

Although virtual screening has become an essential part of the drug discovery process, still "there is no agreed upon theory as to how to conduct a retrospective evaluation" (Nicholls, 2008). In the following, we discuss model validation issues with regard to ligand-based virtual screening, based on literature recommendations (Gramatica, 2007; Hawkins et al., 2008; Jain and Nicholls, 2008, and the references therein). Table 1.5 summarizes recommendations; Cornell (2006) compares existing retrospective studies.

1.4.1 Data selection

The data used in a retrospective virtual screening study are critical, and mistakes can easily lead to overestimated performance. We discuss some important aspects.

Property bias

Differences in simple properties between actives and inactives can lead to good performance of any method (artificial enrichment). The vast size of chemical space makes a purely random selection of decoys both unlikely to be representative of inactivity space and likely to be trivially separable from the actives. Good results due to property bias can be detected by comparison with any simple baseline method. The maximum unbiased validation data set (MUV; Rohrer and Baumann, 2008) was designed to eliminate property (and analogue) bias by matching property distributions of actives and inactives.

Analogue bias and correlation

Virtual screening data sets often contain series of structurally related actives. This violates the assumption of independence which underlies most statistical analyses. Consider, e. g., a data set which contains a compound twice, or a trivial analog of it. Both cases result in a lower estimate of the prediction error (due to increased sample size) without actually reducing the error, leading to overestimation of performance.

Table 1.5 Literature recommendations on the retrospective evaluation of virtual screening studies. ∗ Jain and Nicholls (2008), † Jorgensen (2006), ‡ Cleves and Jain (2008), § Good and Oprea (2008), ¶ Nicholls (2008), ‖ OECD (2004), # Gramatica (2007), ♣ Tropsha et al. (2003), ♢ McGaughey et al. (2007), ♠ Hristozov et al. (2007). References ∗, †, ‡, §, ¶ directly deal with retrospective virtual screening evaluation, ‖, #, ♣ deal with validation of QSAR / QSPR models, and ♢,♠ are reference studies.

Ref.	Recommendation
Data	
∗,¶,♢,♠	Publish usable primary data.
∗†,♢	Use or provide public data, or prove necessity of proprietary data.
†,♠	Use at least one common (benchmark) data set.
¶,♠	Choose number of actives, inactives, and targets using statistical criteria.
∗	Ensure negatives are not active.
∗,¶	Ensure negatives are not trivially separable from positives.
∗	Quantify the diversity of the actives.
∗,§,¶,♢	Use only one representative per chemical class or weight them.
‡	Use old, new, and serendipitous ligands to control human inductive bias.
∗	Use the same protocol to prepare all compounds.
♣	Use an external test set.
Reporting	
‖,#	Clearly state the measured/predicted endpoint, including the assay(s).
‖,#	Unambiguously describe the algorithm, including descriptors.
‖,#,♣	State the domain of applicability, or provide confidence estimates.
♣	Use cross-validation or bootstrapping.
#, ♣	Use Y-randomization.
‖,#	Report robustness of methods.
♢	Report performance of at least one baseline method.
†,♢,♠	Report performance of at least one commonly used method.
∗,♢	Report performance using default parameters as well.
∗,¶,♠	Report receiver operating characteristic area under curve.
∗,¶,♢	Report enrichment at 0.5 %, 1 %, 2 %, 5 %.
∗,¶	Use enrichment variant independent of active to inactive ratio.
∗,♠	Report Pearson's correlation and Kendall's tau.
∗	Provide error bars.
∗	Report accuracy of used experimental data.
‖,#	If possible, interpret the model in physico-chemical terms.

Inductive bias

Known ligands are the result of (incremental) drug development processes, heavily influenced by human expertise. The demonstrated bias of human experts towards molecular graph similarity has in turn led to an over-representation of topologically similar compounds in ligand data sets. The high topological similarity of drugs that bind the same target is therefore, at least in part, not a property of the target, but of the drug development process. This has been investigated in detail by Cleves and Jain (2008).

Ratio and number of actives and inactives

The variance in performance estimates depends on the number of actives and inactives. In evaluations of a single method on a single target, the number of actives dominates error bars, with a low decoy to active ratio being acceptable (a ratio of 4:1 increases ROC AUC 95 % confidence error bars by only 11 % compared to the limiting case of infinitely many decoys). The actual number of actives can be as low as 10. In method comparisons, target-to-target variance dominates, and a large number (on the order of 10^3) of targets is necessary for statistically significant statements. See Nicholls (2008) for details.

Data accuracy

It is meaningless to compare a property with greater precision than the accuracy of the experiment that measures the property. Therefore, one can not in general expect prediction accuracy to exceed the accuracy of the underlying experimental data, e. g., in regression.[16]

Oral stories of laboratories — even within the same organization — unable to reproduce each others measurements to within an order of magnitude are frequent. Similar observations have been reported in the literature.[17] From these, it is clear that regression requires a previous careful analysis of the error in target values. A data set of ligands compiled from various sources will often not be suitable.

Data composition

The test data should reflect, in character and difficulty, the operational application, especially if retrospective evaluation is done to assess potential prospective performance, i. e., if an absolute instead of a relative performance estimation is done.

1.4.2 Performance measures

Comparing virtual screening methods and assessing their usefulness requires a measure of performance. The latter evaluates a predictor on a data set, and is different from, but related to, the loss function, which evaluates a single prediction. A good performance measure depends only on properties of the method (Nicholls, 2008). We focus on ranking methods, as these are the most relevant to virtual screening (p. 24).

[16] Theoretically, this can happen if the measurement error is i. i. d. with zero mean — think of, e. g., points sampled from a line with independent and standard normally distributed measurement error. Given enough samples, linear regression will recover the correct line parameters.

[17] Rücker et al. (2006), for example, report a correlation coefficient of $r^2 = 0.48$ between K_i values (dissociation constants, p. 149) for the peroxisome proliferator-activated receptor γ of 61 compounds using a scintillation proximity assay and a classical solution scintillation assay. Reported K_i values for the drug rosiglitazone range from 47 to 230 nM.

Virtual screening-specific requirements

Virtual screening has special requirements on performance evaluation:

- *Early recognition problem:* In a general machine learning setting, performance on the whole data set is of interest, whereas in virtual screening only the top-ranked compounds are selected for assay tests. Performance on the first part of the ranked data set is therefore of higher interest than performance on the rest of the data.
- *False positives versus false negatives:* Inactive compounds predicted as active (type I errors, false positives) waste money, time, and manpower, whereas active compounds predicted as inactive (type II errors, false negatives) represent missed opportunities. The relative importance of the two error types depends on the project; a good performance measure will reveal the trade-off between the two.

Early enrichment is frequently thought to be important in virtual screening. This assumption is implicitly based on an assumed cost structure, i. e., an assignment of costs to true positives, false positives, true negatives, and false negatives. An example by Nicholls (2008) demonstrates the sensitivity of both the resulting cost effectiveness of a virtual screening method and the importance of early recognition towards the assumed cost structure. We are not aware of any study on this subject based on real cost estimates.

Enrichment factor

One of the simplest performance measures, the *enrichment factor* (Hawkins et al., 2008) at a given fraction $x \in [0, 1]$ of the data set, is given by

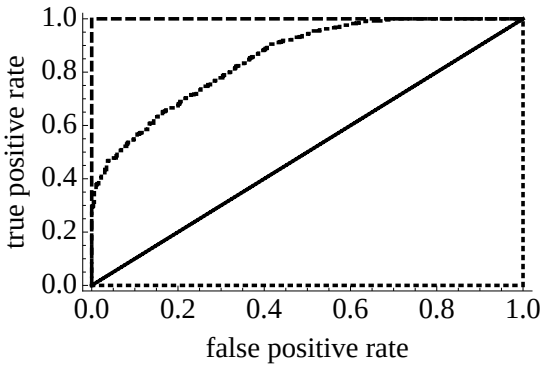
$$\frac{s_+}{s} \Big/ \frac{n_+}{n} = x^{-1} \frac{s_+}{n_+}, \tag{1.11}$$

where n is the number of samples in the data set, n_+ is the number of actives in the data set, $s = xn$ is the number of ranked samples considered, and s_+ is the number of actives in the first s ranked samples. It measures how many times more actives are found within the first s ranked compounds as compared to chance alone. Typical values for x are 0.005, 0.01, 0.02, 0.05 and 0.1. Other performance measures related to the enrichment factor are (Cornell, 2006)

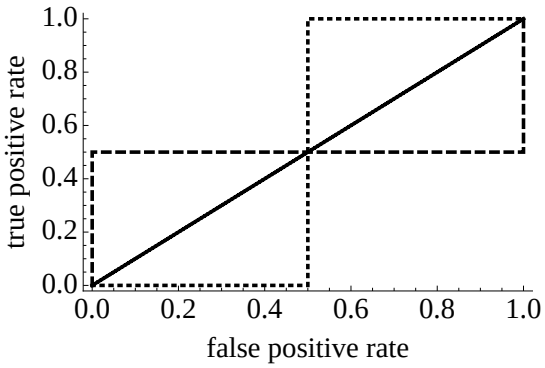
- The fraction s_+/s of actives found in the s top-ranked compounds.
- The screened data fraction x necessary to recover a given fraction s_+/s of actives.
- The maximum enrichment factor, $\max_{x \in [0,1]} x^{-1} s_+/n_+$.
- The fraction x of the data set where the maximum enrichment occurred.

Enrichment is computed easily and measures a quantity related to virtual screening success, but has drawbacks:

- *Dependence on number of actives:* Increasing n_+ for fixed x and n lowers the range of possible enrichment factors, making the statistic dependent on a data set property.
- *Dependence on cut-off:* The enrichment factor is a function of x, and evaluating it at only a few locations gives an incomplete picture of virtual screening performance.
- *No consideration of ties:* Equally ranked samples are not considered. If, e. g., all samples are ranked equal, s_+ is arbitrary, and the enrichment could be chosen at will between 0 and its maximum value n/n_+ (or x^{-1} if $n_+ < s$).



(a) ROC curve examples of a perfect (dashed line, upper left), a random (solid line, diagonal), and a worst-case (dotted line, lower right) ranker, as well as an instance of a learner who can predict values in $[0, 1]$ within $\pm 50\,\%$ (dash-dotted line, $n = 10^3$). Respective ROC AUCs are 1, $\frac{1}{2}$, 0, and 0.85.



(b) ROC curves for rankings with the positive labels at the beginning and end (dashed line), randomly distributed (solid line), and in the middle (dotted line). All curves have a ROC AUC of $\frac{1}{2}$.

Figure 1.7 Examples of receiver operating characteristic curves.

- *Order does not matter:* If $n_+ < s$, it does not matter if the actives are ranked at the beginning or at the end of the first s compounds.

The dependence on a data set property, the ratio of actives to inactives, prevents comparison of enrichment values between data sets. Nicholls (2008) suggests using *ROC enrichment*, where the fraction of actives seen along with a fraction x of the inactives — as opposed to a fraction x of the whole data set as in Equation 1.11 — is reported:

$$x^{-1}\frac{s'_+}{n_+}, \tag{1.12}$$

where s'_+ is the number of actives in the first s' ranked samples and s' is chosen to contain xn_- inactives, n_- being the total number of inactives in the data set. Equation 1.12 does not depend on the ratio of actives to inactives, and, allows analytic error estimation (Pepe, 2004), but still suffers from the other drawbacks.

Receiver operating characteristic

Originally from signal detection theory (Egan, 1975), the *receiver operating characteristic* (ROC), together with the area under its curve (ROC AUC), are well established performance measures for ranking methods. They are extensively used in several disciplines, including medicine and machine learning, and have been applied to evaluate virtual screening methods (Triballeau et al., 2005; Jain and Nicholls, 2008).

A ROC curve plots the fraction of correctly classified positive samples (also true positive rate, recall, sensitivity) on the ordinate over the fraction of incorrectly classified negative samples (also false positive rate, 1 - specificity) on the abscissa. It visualizes the performance of ranking algorithms by varying the decision threshold for positively classified samples over the occurring ranks; in this way, each unique rank creates a point in ROC space (Figure 1.7a; Algorithm 1.1a). The ROC AUC has advantages over enrichment and some other performance measures:

- *Independence of data set properties:* Although the number of samples influences estimation accuracy of the ROC AUC (the positives more than the negatives; Hanley and McNeil, 1982), its value itself is independent of the ratio of actives to inactives.
- *Stochastic interpretation:* The ROC AUC equals the probability of ranking a randomly chosen positive sample above a randomly chosen negative sample.
- *Analytical error estimation:* Due to equivalence with the Wilcoxon-Mann-Whitney test statistic, an error estimate of the ROC AUC can be computed analytically.

Still, the ROC AUC (Algorithm 1.1b) is a one-number summary of a ROC curve, which necessarily loses part of the information: Consider three rankings, without ties, where the first one places the positive samples at the beginning and end of the list, the second one places them randomly, and, the third one places them in the middle of the list. All three lists have the same ROC AUC of $\frac{1}{2}$ (Figure 1.7b), but only the first ranking is suitable for virtual screening, where only the first ranks will be experimentally confirmed.

As a global measure, the ROC AUC therefore does not reflect the possible importance of early hits in virtual screening. Variants proposed to remedy this include the Boltzmann-enhanced discrimination of receiver operating characteristic (BEDROC; Truchon and Bayly, 2007) and usage of semi-logarithmic plots for AUC calculations (pROC AUC; Clark and Webster-Clark, 2008). Both are based on the idea of an exponential weighting according to rank. Such approaches adapt the ROC AUC to early recognition, but partially abandon its advantages; furthermore, data sets containing more actives than the number of ranks considered early can cause saturation effects (Truchon and Bayly, 2007). In a study by Nicholls (2008), ROC AUC and BEDROC scores were highly correlated ($r^2 = 0.901$).

Performance measures for regression

Let y_i denote the true label of the i-th sample and let $\hat{y}_i$ denote its estimate. Performance measures for regression include the *mean squared error*

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{1.13}$$

and its root $\mathrm{RMSE} = \sqrt{\mathrm{MSE}}$, the *mean absolute error*

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{1.14}$$

(also *mean absolute deviation*, MAD), *cumulative histograms*, i. e., rank-frequency plots, and, Pearson's *correlation coefficient*

$$r = \frac{\mathrm{covar}(\boldsymbol{y}, \hat{\boldsymbol{y}})}{\sqrt{\mathrm{var}(\boldsymbol{y})\,\mathrm{var}(\hat{\boldsymbol{y}})}} = \frac{\sum_{i=1}^{n}\left(y_i - \frac{1}{n}\sum_{j=1}^{n} y_j\right)\left(\hat{y}_i - \frac{1}{n}\sum_{j=1}^{n}\hat{y}_j\right)}{\sqrt{\sum_{i=1}^{n}(y_i - \frac{1}{n}\sum_{i=1}^{n} y_i)^2 \sum_{i=1}^{n}(\hat{y}_i - \frac{1}{n}\sum_{i=1}^{n}\hat{y}_i)^2}}. \tag{1.15}$$

Its square r^2 equals the proportion of target value variance accounted for by the regression. As a performance measure for regression, it can be misleading (Figure 1.8).

Like the ROC AUC, these measures do not take the possible importance of early recognition into account, but various modifications exist, e. g., weighting based on utility (Torgo and Ribeiro, 2007).

Algorithm 1.1 Receiver operating characteristic curve, and its area.

(a) Receiver operating characteristic curve. We assume that larger scores $f(\cdot)$ are better, and that labels are either 0 (negatives) or 1 (positives). The idea is to define a function $h_i(\boldsymbol{x}) = 1$ iff $f(\boldsymbol{x}) > f(\boldsymbol{x}_i)$, and -1 otherwise, which classifies all samples with score above $f(\boldsymbol{x}_i)$ as positive. The rates of true and false positives are then computed using h_i, each pair yielding a point in ROC space, not necessarily unique due to ties. The if-statement addresses these. Linear interpolation of the points in P yields the ROC curve.

Input: input samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, labels $y_1, \ldots, y_n \in \{0, 1\}$, scoring function $f : \mathcal{X} \to \mathbb{R}$.
Output: set P of points on the ROC curve.

1 Set $n_+ \leftarrow \sum_{i=1}^{n} y_i$ and $n_- \leftarrow n - n_+$.
2 Sort $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, together with $y_1, \ldots, y_n$, in descending order by f.
3 Set $P \leftarrow \emptyset$, $\boldsymbol{tp}_i \leftarrow \sum_{j=1}^{i} y_j$ and $\boldsymbol{fp}_i \leftarrow i - \boldsymbol{tp}_i$.
4 For each $i \in [1, n)$,
5 If $f(\boldsymbol{x}_i) \neq f(\boldsymbol{x}_{i+1})$ then $P \leftarrow P \cup \left\{ \left(\frac{1}{n_-} \boldsymbol{fp}_i, \frac{1}{n_+} \boldsymbol{tp}_i \right) \right\}$.
6 $P \leftarrow P \cup \big\{(0,0), (1,1)\big\}$.

(b) Area under receiver operating characteristic curve. The vectors $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ correspond to statistics 1, 2, and 3 in Hanley and McNeil (1982). Line 3 computes the number u of unique ranks; ties are appropriately considered.

Input: input samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, labels $y_1, \ldots, y_n \in \{0, 1\}$, scoring function $f : \mathcal{X} \to \mathbb{R}$.
Output: area A under receiver operating characteristic curve.

1 Set $n_+ \leftarrow \sum_{i=1}^{n} y_i$ and $n_- \leftarrow n - n_+$.
2 Sort $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, together with $y_1, \ldots, y_n$, in ascending order by f.
3 Set $u \leftarrow |\{ f(\boldsymbol{x}_i) \mid 1 \leq i \leq n \}|$, $r \leftarrow 1$, and $a \leftarrow c \leftarrow \boldsymbol{0} \in \mathbb{R}^u$.
4 For each $i \in [1, n]$,
5 If $i > 1 \wedge f(\boldsymbol{x}_i) \neq f(\boldsymbol{x}_{i-1})$ then $r \leftarrow r + 1$.
6 If $y_i \neq 1$ then $\boldsymbol{a}_r \leftarrow \boldsymbol{a}_r + 1$ else $\boldsymbol{c}_r \leftarrow \boldsymbol{c}_r + 1$.
7 Set $\boldsymbol{b} \leftarrow \big(\sum_{j=i+1}^{u} \boldsymbol{c}_j \big)_{i=1,\ldots,u}$.
8 Set $A \leftarrow \big\langle \boldsymbol{a}, \boldsymbol{b} + \frac{\boldsymbol{c}}{2} \big\rangle / (n_- \cdot n_+)$.

Other performance measures

Many other performance measures exist. Some metrics like robust initial enhancement (RIE, Sheridan et al., 2001, Merck), cumulative probability (Bursulaya et al., 2003, Molsoft), and average number of outranking decoys (Friesner et al., 2006, Schrödinger) have historically been used only by the groups that invented them (Hawkins et al., 2008). They are not further treated here as they share some of the problems mentioned earlier, and, due to insufficient use, do not offer comparability with the results of others.

Scalar performance measures do not provide insight into the conditions, e. g., misclassification costs and class distributions, under which one predictor is superior to another. ROC plots partially address this issue. Alternatives include cost curves (Drummond and Holte, 2006) and precision-recall curves (Clark and Webster-Clark, 2008).

1.4.3 Statistical validation

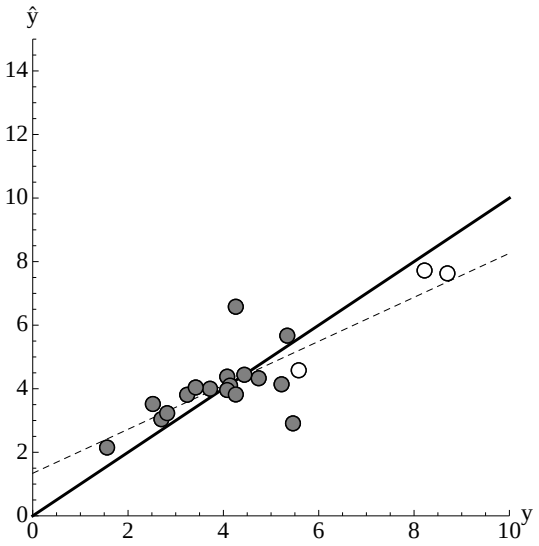
The error of a model on the training data is a measure of how well the model fits these data. A high training error indicates the inability of a model to capture the characteristics of a data set (Figure 1.9a), and a low training error is a prerequisite for successful learning, indicating sufficient capacity of the model class (Figures 1.9b, 1.9c). It is, however, not necessarily indicative of future performance, since a model class that is too complex can lead to over-fitting (rote learning, Figure 1.9c). Consequently, a good model has to have low training and test errors (Figure 1.9b), and retrospective evaluation requires performance measurements on both training and test data. For information on statistical learning theory, see Vapnik (1998, 2001).

In virtual screening, the number of available samples is usually a limiting factor, and generating new samples for testing purposes, or setting aside a substantial fraction of samples as a test set, is not always an option. In such situations, resampling-based statistical validation methods (Hjorth, 1994) can be used. Generalization performance often depends on model parameters, e. g., the kernel width σ of the Gaussian kernel, the parameter C in support vector machines, or, the number of principal components in principal component analysis. Statistical validation can therefore also be used for *model selection*, i. e., to guide the choice of model parameters.

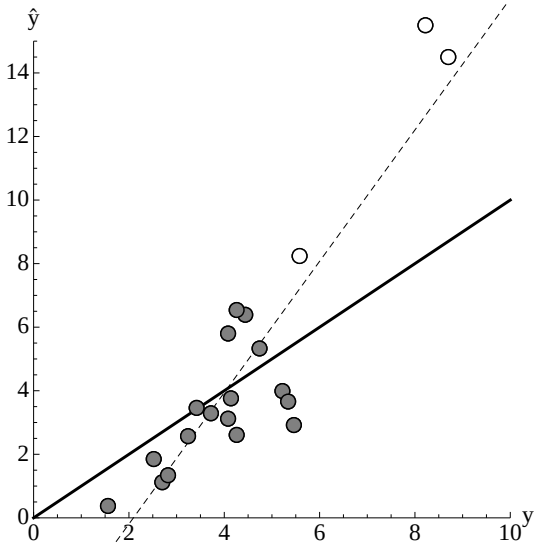
Cross-validation

In k-fold *cross-validation* (also *rotation estimation*; Kohavi, 1995), the training data set is partitioned into $k \in \{2, \ldots, n\}$ *folds* of approximately equal[18] size. Each fold serves as test data set once, with the other folds serving as training data. In this way, each sample is used $k-1$ times for training and once for testing. Common choices of k are 5, 7, and 10; the extreme cases are 2-fold cross-validation and *leave-one-out cross-validation* (LOOCV, $k = n - 1$). Other variants of cross-validation exist; in Chapter 4, we use a leave-k-clusters-out cross-validation strategy to counter over-estimation of performance due to structural correlations in the training data. Note that model selection, e. g., parameter optimization or feature selection, must be done separately for each fold. Cross-validation is different from, and on small data sets superior to, the hold-out (also split-sample) method sometimes used for early stopping in neural networks (Goutte, 1997). It is also different from jackknifing, a resampling scheme used to estimate the bias of a statistic.

[18] Let $m = n - k\lfloor n/k \rfloor$. There are $k - m$ folds of size $\lfloor n/k \rfloor$, and m folds of size $\lfloor n/k \rfloor + 1$.

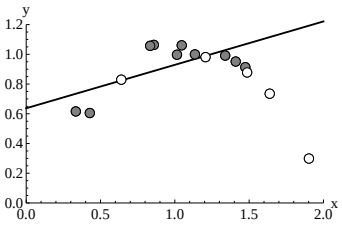


(a) Predictor A has $r = 0.82$ (all points) and $r = 0.55$ (only gray points).

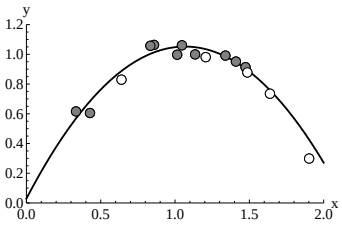


(b) Predictor B has $r = 0.90$ (all points) and $r = 0.64$ (only gray points). Removing the three worst predictions (opaque disks) lowers r.

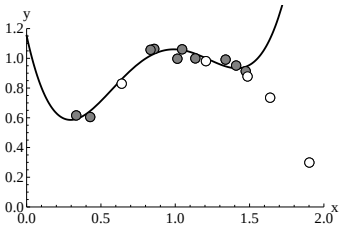
Figure 1.8 The correlation coefficient and regression performance; example after Sheiner and Beal (1981). Pairs $(y, \hat{y})$ of measurements and corresponding predictions are shown as gray disks for two predictors; the three points furthest from the diagonal (black line) in (b) are shown as opaque disks. Although A is clearly the better predictor, it's correlation coefficient is lower than that of B. This is due to the correlation coefficient measuring the best linear relationship between measurements and predictions (thin dashed lines), whereas in regression, one is interested in association along the diagonal.



(a) Linear model $y = 0.64 + 0.29x$ with RMSE of 0.124 (training data) and 0.443 (test data). The model class is not complex enough to capture the pattern in the data, resulting in high training and test errors (underfitting).



(b) Quadratic model $y = 0.03 + 1.92x - 0.90x^2$ with RMSE of 0.044 (training data) and 0.068 (test data). The complexity of the model class fits the data, resulting in low training and test error.



(c) Quartic model $y = 1.15 - 4.55x + 11.60x^2 - 9.89x^3 + 2.75x^4$ with RMSE of 0.036 (training data) and 0.939 (test data). The model class is too complex, resulting in the lowest training and highest test error (overfitting).

Figure 1.9 Model complexity and generalization error. On a training set of 10 points (gray disks), three increasingly complex models — linear (left), quadratic (middle), and quartic (right) — were fitted to minimize the root mean square error. 5 test points from the same distribution as the training data are shown as opaque disks.

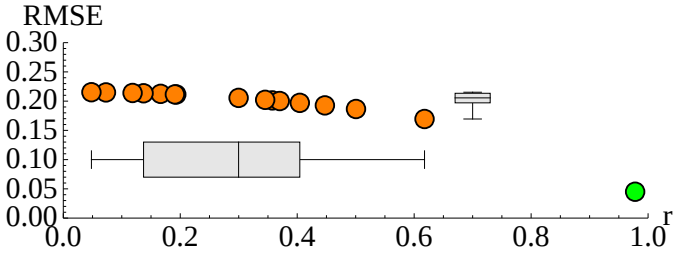


Figure 1.10 *y*-scrambling. Using the data and quadratic model from Figure 1.9b, performance as measured by correlation coefficient r (abscissa) and root mean square error RMSE (ordinate) are shown for the original labels (green disk) and 15 permutations of the labels (orange disks). Superpositioned are box-whisker plots showing range, median, and 25 % as well as 75 % quantiles. Note that correlation coefficient values for random labels spread from 0.05 up to 0.62, whereas the RMSE is more consistent.

Bootstrapping

Another resampling scheme, *bootstrapping* (Efron and Tibshirani, 1993, 1997), can be seen as a smoothed version of cross-validation. Repeated sampling with replacement from the training set is used to create data sets of the same size as the training data set. These are then used for training and performance estimation, averaging over all sets.

Stratification

Stratification is a sampling technique where one partitions the population (training data) into homogeneous strata, and then samples from within each stratum, thereby increasing representativeness of the overall sample. This can be used to reduce the high variance encountered with stochastic methods like cross-validation and bootstrapping. Stratification requires partitioning criteria. In classification, strata are naturally given by the classes, resulting in equal class proportions in each training fold. In regression, the distribution of the target labels should be similar in all folds.

y-scrambling

Good performance may be due to chance alone if the number of explanatory variables, e. g., descriptor dimension, is large relative to the number of samples. To guard against this, one can use *y-scrambling* (also *y-randomization*; Wold et al., 1995), where one repeatedly conducts the learning experiment in question with randomly permuted labels. The resulting performance measurements should be significantly different from the performance based on the original labels (Figure 1.10).

1.4.4 Other aspects

Drug development is necessarily a complex process, depending on many factors. We briefly mention some aspects directly related to virtual screening.

Drug properties

Desired activity on a given target is a necessary but not a sufficient requirement for a drug. Other important properties include absorption, distribution, metabolism, and

excretion (ADME), as well as toxicity. Important examples are aqueous solubility, human intestinal absorption, and oral bioavailability. Late stage ADME / toxicity-related failures are costly and frequent,[19] and efforts are being made to treat these issues earlier on in the drug development process (Li, 2001). This necessitates predictions of these properties (van de Waterbeemd and Gifford, 2003), as well as selection and optimization of compounds according to multiple criteria.

Domain of applicability

A prediction is only useful if it can be trusted. Many machine learning methods and bounds quantifying their generalization performance rely on the assumption that training and test data are independent and identically distributed. As argued in this chapter, this assumption is not valid in typical virtual screening scenarios, requiring ways to quantify the trust in a prediction, or, equivalently, to determine the *domain of applicability* of a model. Although there is consent on this necessity (Jaworska et al., 2003; OECD, 2004), the methodological aspects are not settled.

Some methods provide implicit estimates of prediction confidence, such as Gaussian processes (Subsection 4.2.4); other methods require an external domain of applicability model. These can be divided (Tetko et al., 2006) into those based on molecular similarity (Jaworska et al., 2005) and those based on the predicted property. The former use concepts like descriptor ranges, the presence of fragments, convex hull, or, probability densities to determine whether a test compound is similar enough to the training data to be reliably predicted by the model. The latter compute ensembles of models, using, e. g., different representations, different methods, or resampling schemes. They then analyze the variance of the predictions, or consider the variation in the model residuals.

Scaffold hopping

Ligand-based virtual screening is per se well-suited for the retrieval of close structural analogues. *Scaffold hopping*, the "identification of isofunctional molecular structures with significantly different molecular backbones" (Schneider et al., 1999), is more difficult for an approach based on the similarity principle. Strategies for ligand-based scaffold hopping include inferring information about the target from the ligands, e. g., pseudo-receptor models, and, using representations that sufficiently abstract from the structure graph, e. g., pharmacophore descriptors. The ability of ligand-based virtual screening methods to retrieve new chemotypes was empirically confirmed (Hristozov et al., 2007).

1.5 Conclusions

Ligand-based virtual screening, a problem relevant to drug development, is amenable to machine learning approaches. In this chapter, we give an overview of the problem and some of its aspects, in particular machine learning approaches and retrospective evaluation, and, investigate the role of molecular representation and associated (dis)similarity measures in detail.

[19] In a British study (Prentis et al., 1988), 39 % of failures in clinical development were attributed to inappropriate pharmacokinetics. After removal of a class of poorly bio-available anti-infectives, however, this number reduced to 7 % (Kennedy, 1997). Still, ADME / toxicity problems caused 24 % of all failures.

1.5.1 Summary

Virtual screening, the computational ranking of a compound data set with respect to a predicted property, is a cheminformatics problem relevant to the hit generation phase of drug development. Its ligand-based variant relies upon the similarity principle, which states that (structurally) similar compounds tend to have similar properties. Many machine learning approaches are applicable, including clustering, novelty detection, classification, and regression. We focus on the role of molecular representations, and investigate the effect of the dimensionality of chemical descriptor spaces.

1.5.2 Ligand-based virtual screening

We discuss issues in ligand-based virtual screening based on our exposition of the subject.

Advanced machine learning methods may benefit virtual screening

The applicability of machine learning methods to ligand-based virtual screening has been demonstrated many times, and some algorithms have become standard tools in the field, e. g., neural networks, decision trees, and support vector machines. However, machine learning, in particular kernel-based machine learning, is an area of intense research, and, consequently, many new developments. The recent successful applications of advanced kernel-based machine learning methods to ligand-based virtual screening, e. g., the studies by Schroeter et al. (2007) on aqueous solubility and by Schwaighofer et al. (2008) on metabolic stability using Gaussian processes, hint at the potential improvements in this direction. We back this claim in the following chapters.

Virtual screening lacks a standard protocol for retrospective evaluation

The development of virtual screening is hindered by the lack of a standard protocol for retrospective performance evaluation. This complicates the comparison of studies, a task further aggravated by the use of proprietary data sets. It also affects neighboring disciplines: In view of the increasing relevance of QSAR / QSPR models for regulatory affairs, in particular with regard to the European Union's REACH regulation, there is a growing need for reliable, accurate, and comparable evaluation and validation protocols.

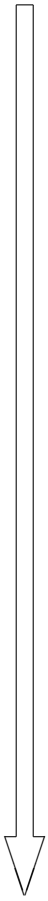
Although the need for such protocols is clear, there is no consensus on the details so far. Until then, we recommend reasonable adherence to the literature recommendations (Table 1.5); a possible ligand-based virtual screening workflow facilitating this is shown in Figure 1.11.

Performance measures

We recommend the use of the receiver operating characteristic, and the area under its curve, on the grounds given in Subsection 1.4.2. From the discussion there, it is clear that enrichment should be avoided, except for legacy comparisons to other studies. In situations where early enrichment is demonstrably relevant, weighted versions of ROC AUC like BEDROC (Truchon and Bayly, 2007) and pROC AUC (Clark and Webster-Clark, 2008), may be given in addition; even better, the ROC curves themselves may be shown.

Statistical validation

We recommend cross-validation or bootstrapping for retrospective evaluation. Cross-validation is almost unbiased (the only bias coming from reduced sample size), but



1 *Data set:*

1.1 Create or select a training data set.

1.2 Assess activity value quality; for classification, set activity cut-off.

1.3 Assess compound diversity; retain one compound per class or weight them.

1.4 Ensure inactives are not trivially separable from actives.

1.5 Preprocess compounds.

2 *Retrospective evaluation:*

2.1 Use cross-validation or bootstrapping to estimate performance.
Use an inner loop of cross-validation or bootstrapping to optimize parameters.

2.2 Create model for whole data set.

2.3 Determine domain of applicability.

3 *Prospective application:*

3.1 Create or select screening data set.

3.2 Prefilter by drug-likeness, assay and target-specific requirements.

3.3 Remove compounds not in domain of applicability.

3.4 Rank test data set using model from step 2.2.

3.5 Visually inspect and select ("cherry-pick") top-ranked compounds.

3.6 Test selected compounds in a bioassay.

Figure 1.11 A ligand-based virtual screening workflow.

has high variance, whereas bootstrapping has lower variance but can be overly optimistic. Other statistical criteria for model selection and performance estimation exist, e. g., Akaike's information criterion (AIC; Akaike, 1974; Bozdogan, 2000), Bayesian information criterion (BIC; Schwartz, 1978), minimum description length (Hansen and Yu, 2003), and, likelihood-based cross-validation (van der Laan et al., 2004). If used, additionally stating cross-validated results may facilitate study comparisons; 10-fold cross-validation is a popular choice.

In cases where the assumption of independence and identical distribution is severely violated, additional measures should be taken, e. g., modification of the training set by retaining only representatives from clusters, or, usage of a modified cross-validation procedure as in Subsection 4.2.5.

Structure-based virtual screening

The computational alternative to ligand-based virtual screening is structure-based virtual screening. With the advent of sophisticated structure elucidation methods, e. g., in-cell nuclear magnetic resonance spectroscopy (in-cell NMR; Sakakibara et al., 2009), requiring the three-dimensional structure of the target might become less restrictive. While structure-based approaches have attractive advantages like providing insight into the mechanism of action, they also have problems, e. g., limitations of crystal structures (Davis et al., 2008) and force fields (Kitchen et al., 2004). In a similar way, high-throughput screening has neither replaced virtual screening, nor was it replaced by it;

instead, synergistic uses have been put forward, like using virtual screening to suggest compounds for re-testing to reduce error rates in high-throughput screening. We expect ligand-based virtual screening to remain a valuable tool in drug development for the foreseeable future.

1.5.3 High-dimensional descriptor spaces

We discuss consequences of high descriptor dimensionality based on our investigation in Subsection 1.5.3.

Neighborhood computations

In neighborhood computations, the size of the neighborhood should depend on dimensionality. For k-nearest neighbor methods, this is automatically achieved by fixing the number of neighbors. When using ϵ-balls as neighborhoods, ϵ should be computed based on the data, e. g., by optimization as in the ad hoc-solution proposed by Tenenbaum et al. in their rejoinder to Balasubramanian (2002).

Structure in chemical data sets

It is common knowledge that chemical data sets exhibit structure due to analogue bias and inductive bias; we formalize this idea with a statistical test in our treatment of the empty space phenomenon. For some representations like auto-correlation vectors, descriptor space dimensions are always neither independent nor identically distributed.

Intrinsic dimensionality

Distance phenomena set in early and depend on the intrinsic dimensionality of the data set in the non-i. i. d. case, which, as discussed above, is rule rather than exception in chemical data sets. It is therefore important to determine the intrinsic dimensionality of the data; we pursue this further in Chapter 3.

1.5.4 Outlook

Virtual screening is a complex problem, and many aspects were only hinted at in this chapter, e. g., training compounds with activity on the same target but different binding mode, the problem of frequent hitters (also promiscuous binders; Roche et al., 2002), or, the screening of virtual combinatorial libraries.

We propose ideas for future research, in increasing order of speculativity:

- *Active learning for ligand-based virtual screening:* By allowing the learning algorithm to suggest compounds for intermediate assay testing, virtual screening can be turned into an active learning problem. Schüller and Schneider (2008) demonstrate this idea in virtual screening by experimentally testing the compounds in each generation of an evolutionary strategy optimization scheme. Such approaches should be particularly effective in combination with virtual combinatorial libraries. An advanced example in another area of science is given by King et al. (2009), who design a robot system that generates functional genomics hypotheses about the yeast *Saccharomyces cerevisiae* and experimentally tests them using microbial batch growth experiments.

- *Free-form learning for QSAR / QSPR models:* Schmidt and Lipson (2009) developed a learning system able to extract analytical expressions describing simple physical systems such as oscillators and pendulums from experimental data. Their approach is based on symbolic regression,[20] regularization, and partial derivatives for performance measurements. In a similar approach, one could try to learn explicit laws, in analytically closed form, describing structure-activity or structure-property relationships.
- *Structured output learning of conformations:* Three-dimensional virtual screening methods require knowledge about the conformational space inhabited by a compound. Based on crystallographic data, kernel-based learning of structured outputs (Bakir et al., 2007) could be used to predict biologically relevant ligand conformations. If further data is available, e. g., from molecular dynamics simulations, conformation distributions could be predicted.
- *Learning electron densities:* The single most important factor in virtual screening is the molecular representation. Quantum mechanical descriptions of molecules, such as electron densities of ground states, are well-founded, sufficient, and accurate, but computationally costly. Various degrees of approximation are used, e. g., different sets of basis functions, semi-empirical parametrizations, restricted treatment of electron correlation, and many others. An approximation based on semi-definite programming (Vandenberghe and Boyd, 1996; Helmberg, 2000) could be used for a kernel-based approach to ligand-based virtual screening founded in quantum mechanics.

References

Christopher Adams, Van Brantner. *Estimating the cost of new drug development: Is it really $ 802 million?* Health Affairs, 25(2): 420–428, 2006.

Charu Aggarwal, Alexander Hinneburg, Daniel Keim. *On the surprising behavior of distance metrics in high dimensional spaces.* In Jan Van den Bussche, Victor Vianu (editors), *Proceedings of the 8th International Conference on Database Theory (ICDT 2001), London, United Kingdom, January 4–6*, volume 1973 of *Lecture Notes in Computer Science*, 420–434. Springer, 2001.

Mark Aizerman, Emmanuel Braverman, Lev Rozonoer. *Theoretical foundations of the potential function method in pattern recognition learning.* Automation and Remote Control, 25: 821–837, 1964.

Hirotugu Akaike. *A new look at the statistical model identification.* IEEE Transactions on Automatic Control, 19(6): 716–723, 1974.

Juan Alvarez, Brian Shoichet (editors). *Virtual Screening In Drug Discovery.* CRC Press, Boca Raton, 2005.

Nachman Aronszajn. *Theory of reproducing kernels.* Transactions of the American Mathematical Society, 68(3): 337–404, 1950.

Assay Drug Dev. Technol.. *Current trends in high-throughput screening. Roundtable discussion.* Assay and Drug Development Technologies, 6(4): 491–504, 2008.

Jürgen Bajorath. *Understanding chemoinformatics: A unifying approach.* Drug Discovery Today, 9(1): 13–14, 2004.

Gökhan Bakir, Thomas Hofmann, Bernhard Schölkopf, Alexander Smola, Ben Taskar, Vishy Vishwanathan (editors). *Predicting Structured Data.* MIT Press, Cambridge, 2007.

Mukund Balasubramanian. *The Isomap algorithm and topological stability.* Science, 295(5552): 7a, 2002.

Richard Bellman. *Dynamic Programming.* Princeton University Press, Princeton, 1957.

Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, Uri Shaft. *When is "nearest neighbor" meaningful?* In *Proceedings of the 7th International Conference on Database Theory (ICDT 1999), Jerusalem, Israel, January 10–12*, volume 1540 of *Lecture Notes in Computer Science*, 217–235, 1999.

[20] Symbolic regression (Koza, 1992, Chapter 10) fits not only the parameters of an equation (model), but also the equation (model) itself using stochastic search algorithms like evolutionary computation.

Christopher Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, 1996.

Christopher Bishop. *Pattern Recognition and Machine Learning*. Springer, Berlin, 2006.

Hans-Joachim Böhm, Gisbert Schneider (editors). *Virtual Screening for Bioactive Molecules*. Wiley-VCH, Weinheim, Germany, 2000.

Bernhard Boser, Isabelle Guyon, Vladimir Vapnik. *A training algorithm for optimal margin classifiers*. In *Proceedings of the 5th Annual ACM Conference on Computational Learning Theory (COLT 1992), Pittsburgh, Pennsylvania, USA, July 27–29, 1992*, 144–152. Association for Computing Machinery, 1992.

Hamparsum Bozdogan. *Akaike's information criterion and recent developments in information complexity*. Journal of Mathematical Psychology, 44(1): 62–91, 2000.

Badry Bursulaya, Maxim Totrov, Ruben Abagyan, Charles Brooks III. *Comparative study of several algorithms for flexible ligand docking*. Journal of Computer-Aided Molecular Design, 17(11): 755–763, 2003.

Olivier Chapelle, Bernhard Schölkopf, Alexander Zien (editors). *Semi-Supervised Learning*. MIT Press, Cambridge, 2006.

Robert Clark, Daniel Webster-Clark. *Managing bias in ROC curves*. Journal of Computer-Aided Molecular Design, 22(3–4): 141–146, 2008.

Ann Cleves, Ajay Jain. *Effects of inductive bias on computational evaluations of ligand-based modeling and on drug discovery*. Journal of Computer-Aided Molecular Design, 22(3–4): 147–159, 2008.

Claude Cohen-Tannoudji, Bernard Diu, Franck Laloë. *Quantum Mechanics*. Wiley-VCH, Berlin, 2006.

Wendy Cornell. *Recent evaluations of high-throughput docking methods for pharmaceutical lead finding — consensus and caveats*. In David Spellmeyer (editor), *Annual Reports in Computational Chemistry*, volume 2, chapter 16, 297–324. Elsevier, 2006.

Nello Cristianini, John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, Cambridge, 2000.

Gabriele Cruciani, Patrizia Crivori, Pierre-Alain Carrupt, Bernard Testa. *Molecular fields in quantitative structure-permeation relationships: the VolSurf approach*. Journal of Molecular Structure (Theochem), 503(1–2): 17–30, 2000.

Andrew Davis, Stephen St-Gallay, Gerard Kleywegt. *Limitations and lessons in the use of X-ray structural information in drug design*. Drug Discovery Today, 13(19–20): 831–841, 2008.

Joseph DiMasi, Ronald Hansen, Henry Grabowski. *The price of innovation: new estimates of drug development costs*. Journal of Health Economics, 22(2): 151–185, 2003.

Dominique Douguet. *Ligand-based approaches in virtual screening*. Current Computer-Aided Drug Design, 4(3): 180–190, 2008.

Chris Drummond, Robert Holte. *Cost curves: An improved method for visualizing classifier performance*. Machine Learning, 65(1): 95–130, 2006.

Richard Duda, Peter Hart, David Stork. *Pattern Classification*. Wiley, New York, second edition, 2001.

Bradley Efron, Robert Tibshirani. *An Introduction to the Bootstrap*, volume 57 of *Monographs on Statistics and Applied Probability*. CRC Press, Boca Raton, 1993.

Bradley Efron, Robert Tibshirani. *Improvements on cross-validation: The .632+ bootstrap method*. Journal of the American Statistical Association, 92(438): 548–560, 1997.

James Egan. *Signal Detection Theorey and ROC Analysis*. Academic Press, 1975.

Uli Fechner, Gisbert Schneider. *Evaluation of distance metrics for ligand-based similarity searching*. ChemBioChem, 5(4): 538–540, 2004.

Miklos Feher. *Consensus scoring for protein-ligand interactions*. Drug Discovery Today, 11(9–10): 421–428, 2006.

Imola Fodor. *A survey of dimension reduction techniques*. *Technical Report UCRL-ID-148494*, Lawrence Livermore National Laboratory, Livermore, California, USA, 2002.

Damien François. *High-dimensional data analysis: optimal metrics and feature selection*. Ph.D. thesis, Université Catholique de Louvain, Louvain-la-Neuve, Belgium, 2007.

Damien François, Vincent Wertz, Michel Verleysen. *The concentration of fractional distances*. IEEE Transactions on Knowledge and Data Engineering, 19(7): 873–886, 2007.

Richard Friesner. *Ab initio quantum chemistry: Methodology and applications*. Proceedings of the National Academy of Sciences of the United States of America, 102(19): 6648–6653, 2005.

Richard Friesner, Robert Murphy, Matthew Repasky, Leah Frye, Jeremy Greenwood, Thomas Halgren, Paul Sanschagrin, Daniel Mainz. *Extra precision glide: Docking and scoring incorporating a model of hydrophobic enclosure for protein-ligand complexes*. Journal of Medicinal Chemistry, 49(21): 6177–6196, 2006.

Johann Gasteiger. *Chemoinformatics: A new field with a long tradition*. Analytical and Bioanalytical Chemistry, 384(1): 57–64, 2006.

Zoubin Ghahramani. *Unsupervised learning.* In Olivier Bousquet, Ulrike von Luxburg, Gunnar Rätsch (editors), *Advanced Lectures on Machine Learning. ML Summer Schools 2003*, volume 3176 of *Lecture Notes in Artificial Intelligence*, 72–112. Springer, 2004.

Arup Ghose, Vellarkad Viswanadhan (editors). *Combinatorial Library Design and Evaluation.* CRC Press, Boca Raton, 2001.

Andrew Good, Tudor Oprea. *Optimization of CAMD techniques 3. virtual screening enrichment studies: a help or hindrance in tool selection?* Journal of Computer-Aided Molecular Design, 22(3–4): 169–178, 2008.

Cyril Goutte. *Note on free lunches and cross-validation.* Neural Computation, 9(6): 1245–1249, 1997.

Paola Gramatica. *Principles of QSAR models validation: internal and external.* QSAR & Combinatorial Science, 26(5): 694–701, 2007.

Isabelle Guyon, André Elisseeff. *An introduction to variable and feature selection.* Journal of Machine Learning Research, 3: 1157–1182, 2003.

Richard Hamming. *Coding and Information Theory.* Prentice-Hall, 1980.

James Hanley, Barbara McNeil. *The meaning and use of the area under a receiver operating characteristic (ROC) curve.* Radiology, 143(1): 29–36, 1982.

Mark Hansen, Bin Yu. *Model selection and the principle of minimum description length.* Journal of the American Statistical Association, 96(454): 746–774, 2003.

Trevor Hastie, Robert Tibshirani, Jerome Friedman. *The Elements of Statistical Learning. Data Mining, Inference, and Prediction.* Springer, New York, 2003.

Paul Hawkins, Geoffrey Skillman, Anthony Nicholls. *Comparison of shape-matching and docking as virtual screening tools.* Journal of Medicinal Chemistry, 50(1): 72–82, 2006.

Paul Hawkins, Gregory Warren, Geoffrey Skillman, Anthony Nicholls. *How to do an evaluation: pitfalls and traps.* Journal of Computer-Aided Molecular Design, 22(3–4): 179–190, 2008.

Christoph Helmberg. *Semidefinite programming for combinatorial optimization. Technical Report ZR-00-34*, Konrad-Zuse-Zentrum Berlin, 2000.

Alexander Hinneburg, Charu Aggarwal, Daniel Keim. *What is the nearest neighbor in high dimensional spaces?* In Amr El Abbadi, Michael Brodie, Sharma Chakravarthy, Umeshwar Dayal, Nabil Kamel, Gunter Schlageter, Kyu-Young Whang (editors), *Proceedings of the 26th International Conference on Very Large Databases (VLDB 2000), Cairo, Egypt, September 10–14*, 506–515. Morgan Kaufmann, 2000.

Urban Hjorth. *Computer Intensive Statistical Methods. Validation, Model Selection, and Bootstrap.* CRC Press, Boca Raton, 1994.

Thomas Hofmann, Bernhard Schölkopf, Alexander Smola. *Kernel methods in machine learning.* Annals of Statistics, 36(6): 1171–1220, 2008.

Dimitar Hristozov, Tudor Oprea, Johann Gasteiger. *Virtual screening applications: A study of ligand-based methods and different structure representations in four different scenarios.* Journal of Computer-Aided Molecular Design, 21(10–11): 617–640, 2007.

Michael Hutter. *Separating drugs from nondrugs: A statistical approach using atom pair distributions.* Journal of Chemical Information and Modeling, 47(1): 186–194, 2009.

Ajay Jain, Anthony Nicholls. *Recommendations for evaluation of computational methods.* Journal of Computer-Aided Molecular Design, 22(3–4): 133–139, 2008.

William Janzen (editor). *High Throughput Screening: Methods and Protocols.* Humana Press, Totowa, 2004.

Joanna Jaworska, M Comber, C Auer, CJ van Leeuwen. *Summary of a workshop on regulatory acceptance of (Q)SARs for human health and environmental endpoints.* Environmental Health Perspectives, 111(10): 1358–1360, 2003.

Joanna Jaworska, Nina Nikolova-Jeliazkova, Tom Aldenberg. *QSAR applicabilty domain estimation by projection of the training set descriptor space: A review.* Alternatives to Laboratory Animals, 33(5): 445–459, 2005.

Mark Johnson, Gerald Maggiora (editors). *Concepts and Applications of Molecular Similarity.* Wiley, New York, 1990.

William Jorgensen. *QSAR/QSPR and proprietary data.* Journal of Chemical Information and Modeling, 46(3): 937, 2006.

Peter Jurs. *Quantitative structure-property relationships.* In Johann Gasteiger (editor), *Handbook of Chemoinformatics: From Data to Knowledge*, volume 3, chapter 1.2, 1314–1135. Wiley, 2003.

Tony Kennedy. *Managing the drug discovery / development interface.* Drug Discovery Today, 2(10): 436–444, 1997.

Thomas Kerr. *Misstatements of the test for positive semidefinite matrices.* Journal of Guidance, Control and Dynamics, 13(3): 571–572, 1990.

Ross King, Jem Rowland, Stephen Oliver, Michael Young, Wayne Aubrey, Emma Byrne, Maria Liakata, Magdalena Markham, Pinar Pir, Larisa Soldatova, Andrew Sparkes, Kenneth Whelan, Amanda Clare. *The automation of science.* Science, 324(5923): 85–89, 2009.

Douglas Kitchen, Hélène Decornez, John Furr, Jürgen Bajorath. *Docking and scoring in virtual screening for drug discovery: Methods and applications.* Nature Reviews Drug Discovery, 3(11): 935–949, 2004.

Gerhard Klebe. *Virtual ligand screening: Strategies, perspectives and limitations.* Drug Discovery Today, 11(13–14): 580–594, 2006.

Ron Kohavi. *A study of cross-validation and bootstrap for accuracy estimation and model selection.* In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI 1995), Montréal, Québec, Canada, August 20–25*, 1137–1145. Morgan Kaufmann, 1995.

Sotos Kotsiantis. *Supervised machine learning: A review of classification techniques.* Informatica, 31(3): 249–268, 2007.

John Koza. *Genetic Programming. On the Programming of Computers By Means of Natural Selection.* MIT Press, Cambridge, 1992.

Romano Kroemer. *Molecular modelling probes: Docking and scoring.* Biochemical Society Transactions, 31(5): 980–984, 2003.

Hugo Kubinyi. *Similarity and dissimilarity: A medicinal chemist's view.* Perspectives in Drug Discovery and Design, 9-11: 225–252, 1998.

Hugo Kubinyi. *QSAR in drug design.* In Johann Gasteiger (editor), *Handbook of Chemoinformatics: From Data to Knowledge*, volume 4, chapter 4.2, 1532–1554. Wiley, 2003.

Hugo Kubinyi. *Success stories of computer-aided design.* In Sean Ekins (editor), *Computer Applications in Pharmaceutical Research and Development*, 377–424. Wiley, New York, 2006.

Albert Li. *Screening for human ADME / Tox properties in drug discovery.* Drug Discovery Today, 6(7): 357–366, 2001.

David MacKay. *Information theory, Inference, and Learning Algorithms.* Cambridge University Press, Cambridge, 2003.

Peter Mager. *Theoretical approaches to drug design and biological activity: Critical comments to the use of mathematical methods applied to univariate and multivariate quantitative structure-activity relationships (QSAR).* Medicinal Research Reviews, 2(1): 93–121, 1982.

Yvonne Martin, James Kofron, Linda Traphagen. *Do structurally similar molecules have similar biological activity?* Journal of Medicinal Chemistry, 45(19): 4350–4358, 2002.

Georgia McGaughey, Robert Sheridan, Christopher Bayly, Chris Culberson, Constantine Kreatsoulas, Stacey Lindsley, Vladimir Maiorov, Jean-Francois Truchon, Wendy Cornell. *Comparison of topological, shape, and docking methods in virtual screening.* Journal of Chemical Information and Modeling, 47(4): 1504–1519, 2007.

Donald McQuarrie. *Quantum Chemistry.* University Science Books, Sausalito, 2007.

James Mercer. *Functions of positive and negative type, and their connection with the theory of integral equations.* Philosophical Transactions of the Royal Society of London, 209: 415–446, 1909.

Carl Meyer. *Matrix Analysis and Applied Linear Algebra.* Society for Industrial and Applied Mathematics, Philadelphia, 2001.

Gerry Moss. *Basic terminology of stereochemistry (IUPAC recommendations 1996).* Pure and Applied Chemistry, 68(12): 2193–2222, 1996.

Paul Newbold, David Harvey. *Forecast combination and encompassing.* In Michael Clements, David Henry (editors), *A companion to economic forecasting*, chapter 12, 268–283. Blackwell, Oxford, 2002.

Anthony Nicholls. *What do we know and when do we know it?* Journal of Computer-Aided Molecular Design, 22(3–4): 239–255, 2008.

OECD. *OECD principles for the validation, for regulatory purposes, of (quantitative) structure-activity relationship models.* In *37th Joint Meeting of the Chemicals Committee and Working Party on Chemicals, Pesticides and Biotechnology, November 2004.* OECD, 2004.

John Overington, Bissan Al-Lazikani, Andrew Hopkins. *How many drug targets are there?* Nature Reviews Drug Discovery, 5(12): 993–996, 2006.

Margaret Pepe. *The Statistical Evaluation of Medical Tests for Classification and Prediction.* Oxford University Press, Oxford, 2004.

Tuan Pham, Ajay Jain. *Parameter estimation for scoring protein-ligand interactions using negative training data.* Journal of Medicinal Chemistry, 49(20): 5856–5868, 2006.

Prentis, Lis, Walker. *Pharmaceutical innovation by the seven UK-owned pharmaceutical companies (1964–1985).* British Journal of Clinical Pharmacology, 25(3): 387–396, 1988.

Public Citizen. *Rx R & D Myths: The Case Against the Drug Industry's R & D "Scare Card"*, 2001.

Olivier Roche, Petra Schneider, Jochen Zuegge, Wolfgang Guba, Manfred Kansy, Alexander Alanine, Konrad Bleicher, Franck Danel, Eva-Maria Gutknecht, Mark Rogers-Evans, Werner Neidhart, Henri Stalder, Michael Dillon, Eric Sjögren, Nader Fotouhi, Paul Gillespie, Robert Goodnow, William Harris, Phil Jones, Mikio Taniguchi, Shinji Tsujii, Wolfgang von der Saal, Gerd Zimmermann, Gisbert Schneider. *Development of a virtual screening method for identification of „frequent hitters" in compound libraries.* Journal of Medicinal Chemistry, 45(1): 137–142, 2002.

Sebastian Rohrer, Knut Baumann. *Impact of benchmark data set topology on the validation of virtual screening methods: Exploration and quantification by spatial statistics.* Journal of Chemical Information and Modeling, 48(4): 704–718, 2008.

Lior Rokach, Oded Maimon. *Data Mining with Decision Trees. Theory and Applications.* World Scientific Publishing, Toh Tuck Link, 2008.

Christoph Rücker, Marco Scarsi, Markus Meringer. *2D QSAR of PPARγ agonist binding and transactivation.* Bioorganic & Medicinal Chemistry, 14(15): 5178–5195, 2006.

Matthias Rupp, Petra Schneider, Gisbert Schneider. *Distance phenomena in high-dimensional chemical descriptor spaces: Consequences for similarity-based approaches.* Journal of Computational Chemistry, 30(14): 2285–2296, 2009.

Stuart Russell, Peter Norvig. *Artificial Intelligence: A Modern Approach.* Prentice Hall, second edition, 2002.

Daisuke Sakakibara, Atsuko Sasaki, Teppei Ikeya, Junpei Hamatsu, Tomomi Hanashima, Masaki Mishima, Masatoshi Yoshimasu, Nobuhiro Hayashi, Tsutomu Mikawa, Markus Wälchli, Brian Smith, Masahiro Shirakawa, Peter Güntert, Yutaka Ito. *Protein structure determination in living cells by in-cell NMR spectroscopy.* Nature, 458(7234): 102–105, 2009.

Frank Sams-Dodd. *Target-based drug discovery: Is something wrong?* Drug Discovery Today, 10(2): 139–147, 2005.

Michael Schmidt, Hod Lipson. *Distilling free-form natural laws from experimental data.* Science, 324(5923): 81–85, 2009.

Gisbert Schneider, Karl-Heinz Baringhaus. *Molecular Design. Concepts and Applications.* Wiley, Weinheim, 2008.

Gisbert Schneider, Uli Fechner. *Computer-based de novo design of drug-like molecules.* Nature Reviews Drug Discovery, 4(8): 649–663, 2005.

Gisbert Schneider, Werner Neidhart, Thomas Giller, Gerard Schmid. *"Scaffold-hopping" by topological pharmacophore search: A contribution to virtual screening.* Angewandte Chemie International Edition, 38(19): 2894–2896, 1999.

Bernhard Schölkopf, Alexander Smola. *Learning with Kernels.* MIT Press, Cambridge, 2002.

Timon Schroeter, Anton Schwaighofer, Sebastian Mika, Antonius Ter Laak, Detlev Sülzle, Ursula Ganzer, Nikolaus Heinrich, Klaus-Robert Müller. *Estimating the domain of applicability for machine learning QSAR models: A study on aqueous solubility of drug discovery molecules.* Journal of Computer-Aided Molecular Design, 21(9): 485–498, 2007.

Andreas Schüller, Gisbert Schneider. *Identification of hits and lead structure candidates with limited resources by adaptive optimization.* Journal of Chemical Information and Modeling, 48(7): 1473–1491, 2008.

Anton Schwaighofer, Timon Schroeter, Sebastian Mika, Katja Hansen, Antonius ter Laak, Philip Lienau, Andreas Reichel, Nikolaus Heinrich, Klaus-Robert Müller. *A probabilistic approach to classifying metabolic stability.* Journal of Chemical Information and Modeling, 48(4): 785–796, 2008.

Gideon Schwartz. *Estimating the dimension of a model.* Annals of Statistics, 6(2): 461–464, 1978.

David Scott, James Thompson. *Probability estimation in higher dimensions.* In James Gentle (editor), *Computer Science and Statistics: Proceedings of the 15th Symposium on the Interface (Interface 1983), Houston, Texas, USA, March 17–18*, 173–179. North-Holland Publishing Company, 1983.

Markus Seifert, Martin Lang. *Essential factors for successful virtual screening.* Mini Reviews in Medicinal Chemistry, 8(1): 63–72, 2008.

John Shawe-Taylor, Nello Cristianini. *Kernel methods for pattern analysis.* Cambridge University Press, New York, first edition, 2004.

Lewis Sheiner, Stuart Beal. *Some suggestions for measuring predictive performance.* Journal of Pharmacokinetics and Pharmacodynamics, 9(4): 503–512, 1981.

Robert Sheridan, Suresh Singh, Eugene Fluder, Simon Kearsley. *Protocols for bridging the peptide to nonpeptide gap in topological similarity searches.* Journal of Chemical Information and Computer Sciences, 41(5): 1395–1406, 2001.

Matthew Sills, Donna Weiss, Quynhchi Pham, Robert Schweitzer, Xiang Wu, Jinzi Wu. *Comparison of assay technologies for a tyrosine kinase assay generates different results in high throughput screening.* Journal of Biomolecular Screening, 7(3): 191–214, 2002.

Olivier Sperandio, Maria Miteva, Bruno Villoutreix. *Combining ligand- and structure-based methods in drug design projects.* Current Computer-Aided Drug Design, 4(3): 250–258, 2008.

Ingo Steinwart, Don Hush, Clint Scovel. *An explicit description of the reproducing kernel hilbert spaces of Gaussian RBF kernels.* IEEE Transactions on Information Theory, 52(10): 4635–4643, 2006.

Yusuf Tanrikulu, Gisbert Schneider. *Pseudoreceptor models in drug design: Bridging ligand- and receptor-based virtual screening.* Nature Reviews Drug Discovery, 7(8): 667–677, 2008.

Igor Tetko, Pierre Bruneau, Hans-Werner Mewes, Douglas Rohrer, Gennadiy Poda. *Can we estimate the accuracy of ADME-Tox predictions?* Drug Discovery Today, 11(15–16): 700–707, 2006.

Gareth Thomas. *Fundamentals of Medicinal Chemistry.* Wiley, New York, first edition, 2003.

Roberto Todeschini, Viviana Consonni. *Handbook of Molecular Descriptors.* Wiley-VCH, Weinheim, Germany, first edition, 2000.

John Topliss, Robert Edwards. *Chance factors in studies of quantitative structure-activity relationships.* Journal of Medicinal Chemistry, 22(10): 1238–1244, 1979.

Luis Torgo, Rita Ribeiro. *Utility-based regression.* In Joost Kok, Jacek Koronacki, Ramon Lopez de Mantaras, Stan Matwin, Dunja Mladenič, Andrzej Skowron (editors), *Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2007), Warsaw, Poland, September 17–21*, volume 4702 of *Lecture Notes in Artificial Intelligence*, 597–604. Springer, 2007.

Nicolas Triballeau, Francine Acher, Isabelle Brabet, Jean-Philippe Pin, Hugues-Olivier Bertrand. *Virtual screening workflow development guided by the "receiver operating characteristic" curve approach. Application to high-throughput docking on metabotropic glutamate receptor subtype 4.* Journal of Medicinal Chemistry, 48(7): 2534–2547, 2005.

Alexander Tropsha, Paola Gramatica, Vijay Gombar. *The importance of being earnest: Validation is the absolute essential for successful application and interpretation of QSPR models.* QSAR & Combinatorial Science, 22(1): 69–77, 2003.

Jean-François Truchon, Christopher Bayly. *Evaluating virtual screening methods: Good and bad metrics for the "early recognition" problem.* Journal of Chemical Information and Modeling, 47(2): 488–508, 2007.

Han van de Waterbeemd, Eric Gifford. *ADMET in silico modelling: Towards prediction paradise?* Nature Reviews Drug Discovery, 2(3): 192–204, 2003.

Mark van der Laan, Sandrine Dudoit, Sunduz Keles. *Asymptotic optimality of likelihood based cross-validation.* Statistical Applications in Genetics and Molecular Biology, 3(1): Article 3, 2004.

Lieven Vandenberghe, Stephen Boyd. *Semidefinite programming.* SIAM Review, 38(1): 49–95, 1996.

Vladimir Vapnik. *Statistical Learning Theory.* Wiley, New York, 1998.

Vladimir Vapnik. *The Nature of Statistical Learning Theory.* Springer, second edition, 2001.

Vellarkad Viswanadhan, Arup Ghose, Ganapathi Revankar, Roland Robins. *Atomic physicochemical parameters for three dimensional structure directed quantitative structure-activity relationships. 4. Additional parameters for hydrophobic and dispersive interactions and their applications for an automated superposition of certain naturally occurring nucleoside antibiotics.* Journal of Chemical Information and Computer Sciences, 29(3): 163–172, 1989.

Patrick Walters, Matthew Stahl, Mark Murcko. *Virtual screening — an overview.* Drug Discovery Today, 3(4): 160–178, 1998.

Gregory Warren, Webster Andrews, Anna-Maria Capelli, Brian Clarke, Judith LaLonde, Millard Lambert, Mika Lindvall, Neysa Nevins, Simon Semus, Stefan Senger, Giovanna Tedesco, Ian Wall, James Woolven, Catherine Peishoff, Martha Head. *A critical assessment of docking programs and scoring functions.* Journal of Medicinal Chemistry, 49(20): 5912–5931, 2006.

David Weisgerber. *Chemical abstracts service chemical registry system: History, scope, and impacts.* Journal of the American Society for Information Science, 48(4): 349–360, 1997.

Peter Willett. *Chemical similarity searching.* Journal of Chemical Information and Computer Sciences, 38(6): 983–996, 1998.

Peter Willett. *Enhancing the effectiveness of ligand-based virtual screening using data fusion.* QSAR & Combinatorial Science, 25(12): 1143–1152, 2006a.

Peter Willett. *Similarity-based virtual screening using 2D fingerprints.* Drug Discovery Today, 11: 1046–1053, 2006b.

Peter Willett, John Barnard, Geoffrey Downs. *Chemical similarity searching.* Journal of Chemical Information and Computer Sciences, 38(6): 983–996, 1998.

Svante Wold, Lennart Eriksson, Sergio Clementi. *Statistical validation of QSAR results.* In *Chemometric Methods in Molecular Design*, Methods and Principles in Medicinal Chemistry, 309–318. Wiley-VCH, 1995.

Kurt Wüthrich. *NMR studies of structure and function of biological macromolecules (Nobel lecture).* Angewandte Chemie International Edition, 42(29): 3340–3363, 2003.

Ling Xue, Jeffrey Godden, Jürgen Bajorath. *Database searching for compounds with similar biological activity using short binary bit string representations of molecules.* Journal of Chemical Information and Computer Sciences, 39(5): 881–886, 1999.

Ling Xue, Jeffrey Godden, Jürgen Bajorath. *Evaluation of descriptors and mini-fingerprints for the identification of molecules with similar activity.* Journal of Chemical Information and Computer Sciences, 40(5): 1227–1234, 2000.

All models are wrong,
some are useful.

George Box

Chapter 2

A molecular kernel based on iterative graph similarity and optimal assignments

The most important factor in virtual screening are the data and their representation. While vector representations are popular with computational approaches, the molecular structure graph is a universally used chemical representation. We introduce a kernel defined directly on the annotated molecular structure graph. The annotation allows for the incorporation of domain- and problem-specific knowledge. The graph kernel itself is based on iterative graph similarity and optimal assignments. We give an iterative algorithm for its computation, prove convergence of the algorithm and the uniqueness of the solution, provide an upper bound on the required number of iterations necessary to achieve a desired precision, and discuss its positive semidefiniteness. A retrospective evaluation using support vector machine classification and regression on pharmaceutical and toxicological data sets shows encouraging results.

2.1 Introduction

Molecular representation and corresponding (dis)similarity measure are crucial choices in a virtual screening study (Section 1.3). Many approaches focus on vector representations, e. g., descriptors and fingerprints; consequently, popular molecular (dis)similarity measures include metrics, similarity coefficients, and kernels on vector spaces.

2.1.1 Structured molecular representations

Other approaches use more structured representations, e. g., strings and graphs (Table 2.1). Often, these are not compared directly, but are first converted into vectors, e. g., using fingerprints, hashing, or binning, and are then compared using vector-based (dis)similarity measures. Such conversions are not always chemically motivated, and information can be lost or noise added in the process.

Table 2.1 Examples of structured molecular representations and formats. See Engel (2003) for further information on the representation of chemical compounds.

Structure	Examples
Strings	Simplified molecular input line entry specification (SMILES; Weininger, 1988).
	Molecular query language (MQL, Proschak et al., 2007).
	International chemical identifier (InChi; Heller and McNaught, 2009).
Graphs	Adjacency list, e. g., structured data format (SDF; Symyx, 2007).
	Graph Laplacian (also admittance matrix, Kirchhoff matrix).
Trees	See Subsection 2.2.3.
Grammars	RNA secondary structure (Searls, 2002).
Distributions	Electron densities, e. g., based on quantum chemistry (McQuarrie, 2007).
Various	Shape descriptors, e. g., shapelets (Proschak et al., 2008).

An alternative is to directly compare structured representations, avoiding the conversion into vectors. In kernel-based learning, based on the work of Haussler (1999), kernels on structured data like strings (Joachims, 2002; Vishwanathan and Smola, 2003), labeled ordered trees (Collins and Duffy, 2002), or, probability distributions (Jebara et al., 2004), have been developed. See Gärtner (2009) for information on structured data kernels.

2.1.2 Graph theory and notation

We recapitulate basic terminology. See Diestel (2005) for a graph theory introduction.

Graphs

A *graph*[1] $G = (V, E)$ consists of a set of *vertices* (also *nodes*) $V = \{v_1, \ldots, v_{|V|}\}$ and a set of *edges* $E = \{e_1, \ldots, e_{|E|}\}$. The edges are either ordered vertex pairs (v, v') if the graph is *directed*, or, unordered vertex pairs $\{v, v'\}$ if the graph is *undirected*. For our purposes, we do not allow self-loops and multiple edges between two vertices. Note that for intuition's sake, we draw graph edges as in chemical structure graphs; see Figure 2.10 (p. 86) for an example. In directed graphs, the number $\left|\left\{v' \in V \;\middle|\; (v', v) \in E\right\}\right|$ of incoming neighbors of v is its *in-degree*, and the number $\left|\left\{v' \in V \;\middle|\; (v, v') \in E\right\}\right|$ of outgoing neighbors is its *out-degree*; in undirected graphs, the *degree* of v is the number $\left|\left\{v' \in V \;\middle|\; \{v, v'\} \in E\right\}\right|$ of its neighbors.

Labels

A *labeled graph* has labels (over some arbitrary but fixed domain $\mathcal{L}$) attached to its vertices, edges, or both; the labels encode additional information. We denote with $\operatorname{label}(v) \in \mathcal{L}$ the label of a vertex v and with $\operatorname{label}(e) \in \mathcal{L}$ the label of an edge e. To measure similarity between labels, let k_v and k_e be positive definite kernels defined on the set $\mathcal{L}$ of vertex and edge labels. To shorten notation, we set $k_v(v, v') = k_v\big(\operatorname{label}(v), \operatorname{label}(v')\big)$ and $k_e(e, e') = k\big(\operatorname{label}(e), \operatorname{label}(e')\big)$. In the following, we assume that these kernels have unit range, $0 \leq k_v, k_e \leq 1$.

[1] Graph theory was started by Euler (1736, see Alexanderson, 2006). The word "graph" was introduced by Sylvester (1878), based on the term "graphical notation", as the chemical structure graph was called then. See Rouvray (1991) for information on the origins of chemical graph theory.

Matrix representation

The *adjacency matrix* $\boldsymbol{A} \in \mathbb{R}^{|V|\times|V|}$ of G is given by $\boldsymbol{A}_{i,j} = 1_{\{\{v_i,v_j\}\in E\}}$. The entries $\left(\boldsymbol{A}^k\right)_{i,j}$ of its k-th power, $k \geq 1$, give the number of walks of length k from vertex v_i to vertex v_j. For directed graphs, $\boldsymbol{A}_{i,j} = 1_{\{(v_i,v_j)\in E\}}$. Let $l_1, \ldots, l_l$ denote the possible vertex labels. The *label vertex matrix* $\boldsymbol{L} \in \mathbb{R}^{|\mathcal{L}|\times|V|}$ of G is given by $\boldsymbol{L}_{i,j} = 1_{\{l_i=\text{label}(v_j)\}}$. The k-th diagonal element of $\boldsymbol{L}\boldsymbol{L}^T$ gives the number of vertices with label l_k. Combined, the entry $\left(LAL^T\right)_{ij}$ gives the number of edges between vertices labeled with l_i and l_j.

Properties

A *walk* of edge-length k is a sequence $v_1e_1v_2e_2\ldots e_kv_{k+1}$ of vertices and edges in G with $e_i = \{v_i, v_{i+1}\}$. The vertex-length of a walk is one more than its edge length. A *path* is a walk with distinct, i. e., pairwise different, vertices. A graph with $V = \{v_1, \ldots, v_k\}$ and $E = \big\{\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{k-1}, v_k\}\big\} \cup \big\{\{v_k, v_1\}\big\}$ is a *cycle*. A graph is *connected* iff there is a path between each pair of its vertices. A graph $G' = (V', E')$ is a *subgraph* of G iff $V' \subseteq V$ and $E' \subseteq E$. A graph G is *isomorphic* to another graph G' iff there is an isomorphism between V and V' which preserves edge structure, i. e., if there is a permutation π such that $\{v_i, v_j\} \in E \Leftrightarrow \{v'_{\pi(i)}, v'_{\pi(j)}\} \in E'$.

Trees

A connected graph without cycles is a *tree*; those of its vertices with degree 1 are called *leaves*. In a *rooted tree*, one of the leaves is called the trees *root*. For directed graphs, all vertices have in-degree 1, except the root; the vertices with out-degree 0 are leaves. The *depth* of a tree node is the length of a path from the root to the node. A perfectly depth-balanced tree of order h is a tree where each leaf has depth h. The *branching cardinality* of a tree is one less than its number of leaf nodes. A graph whose components are trees is a *forest*.

Treewidth

Treewidth (Robertson and Seymour, 1986) is a measure of the tree-likeness of a graph. A tree-decomposition maps the vertices of a graph to subtrees of a tree such that the subtrees of adjacent vertices intersect. Let $G = (V, E)$ be a graph, let T denote the vertices of a tree, and let $V_1, \ldots, V_{|T|} \subseteq V$. A *tree-decomposition* $(T, \{V_1, \ldots, V_{|T|}\})$ satisfies

$$V = \bigcup_{t\in T} V_t, \tag{2.1}$$

$$\forall\{u, v\} \in E : \exists\, t \in T : u, v \in V_t, \tag{2.2}$$

$$\forall t, t', t'' \in T \text{ with } t' \text{ on the path from } t \text{ to } t'' \ : V_t \cap V_{t''} \subseteq V_{t'}. \tag{2.3}$$

Equations 2.1 and 2.2 state that G is the union of the subgraphs induced by the parts V_i, and Equation 2.3 enforces a tree-like organization of the parts. The *width* of a tree decomposition is given by

$$\max\big\{|V_t| - 1 \,\big|\, t \in T\big\}, \tag{2.4}$$

and the *treewidth* of G is the smallest width of any tree decomposition of G.

Product graphs

Let $G = (V, E)$ and $G' = (V', E')$ denote two graphs. In the *direct product graph*, vertices correspond to pairs of vertices from G and G', and edges are present only if edges exist for both corresponding vertex pairs. Formally, $G_\times = (V_\times, E_\times)$ with

$$V_\times = V \times V' = \big\{(v, v') \,\big|\, v \in V \wedge v' \in V'\big\}, \tag{2.5}$$

$$E_\times = \Big\{\big\{(u, u'), (v, v')\big\} \,\Big|\, \{u, v\} \in E \wedge \{u', v'\} \in E'\Big\}. \tag{2.6}$$

A walk in $G_\times$ corresponds to two walks, one in G and one in G'.

Spectral graph theory

Let $\boldsymbol{M}$ be a symmetric matrix with eigenvalues $\lambda_1, \ldots, \lambda_n$. The *spectral radius*

$$\rho(\boldsymbol{M}) = \max_{1 \le i \le n} |\lambda_i| \tag{2.7}$$

is the largest magnitude of the eigenvalues of $\boldsymbol{M}$. If $\boldsymbol{M}$ is real and component-wise non-negative, e. g., the adjacency matrix of an undirected graph, then $\rho(\boldsymbol{M})$ is an eigenvalue of $\boldsymbol{M}$, called the *Perron root*.

2.1.3 Characteristics of molecular graphs

In *molecular graphs* (also *structure graphs*), vertices correspond to atoms, and edges correspond to covalent bonds. Such graphs possess distinct characteristics that can be exploited in the design of specialized graph kernels.

Graph type

Molecular graphs are simple graphs, i. e., they are undirected,[2] and have neither self-loops nor multiple edges. They are connected (for salts and ions, only the largest fragment is kept). Vertices and edges are annotated with element and bond type information. Often, there are additional annotations in the form of descriptors, e. g., E-state indices (Kier and Hall, 1999).

Many organic compounds are *planar* (Rücker and Meringer, 2002), i. e., they can be embedded in the plane without two edges crossing. Some graph theoretical problems are computationally easier on planar graphs (Nishizeki and Chiba, 1988).

Size

Molecular graphs can in general be very large, e. g., the muscle protein titin has approximately $4.23 \cdot 10^5$ atoms. In ligand-based virtual screening, however, only small molecules are considered.[3] Hydrogen atoms can often be treated implicitly and therefore do not have to be represented as vertices. In, e. g., the `COBRA` data set (p. 88), the median molecular graph size is 28, and no graph has more than 98 vertices (Table 2.2a, Figure 2.1a). The small size of molecular graphs admits otherwise infeasible algorithms, e. g., algorithms with cubic runtime in the limit and a small constant factor.

[2] It is valid to model molecular graphs as directed graphs, e. g., as in Mahé and Vert (2009); for our purposes, there is no advantage to this.

[3] *Biopharmaceuticals* (also *biologicals*), i. e., medicinal products containing biotechnology-derived proteins as active substances (European Medicines Agency, 2005), such as erythropoietin, insulin, and growth hormone, are an exception. They differ from conventional small-molecule drugs, among other aspects, in size and complexity of the active substance (Roger and Mikhail, 2007).

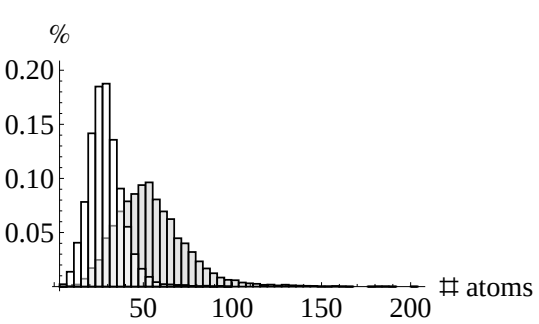
Table 2.2 Statistics of molecular graph properties of the COBRA data set (p. 88; 10 848 molecules, 590 778 atoms). std. dev. = standard deviation, H = with explicit hydrogens, no H = with implicit hydrogens.

(a) Molecule size (number of atoms).

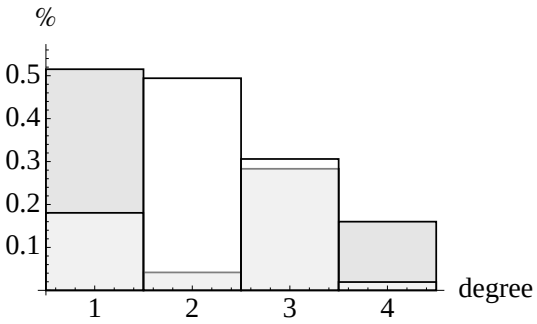
Molecule size	H	no H
Mean	54.46	29.19
Std. dev.	19.77	9.77
Median	52	28
Minimum	7	4
Maximum	203	98

(b) Vertex degree (number of covalent bonds).

Vertex degree	H	no H
Mean	2.09	2.16
Std. dev.	1.20	0.73
Median	1	2
Minimum	1	1
Maximum	4	4



(a) Molecule size (number of atoms).



(b) Vertex degree (number of covalent bonds).

Figure 2.1 Histograms of molecular graph properties for the COBRA data set (p. 88; 10 848 molecules, 590 778 atoms). Shown are histograms with implicit hydrogens (white) and with explicit hydrogens (gray).

Maximum vertex degree

In molecular graphs, the vertex degree is limited by a small constant, the maximum valency number, which is 7 for non-metals such as, e. g., Iodine. The *coordination number*, the number of neighboring atoms linked to a central atom, can be up to 12 for solid-phase metals.[4] Since we are interested mainly in covalent binding, we neglect such higher coordination numbers. Indeed, the maximum valency number found in the COBRA data set (p. 88) was 4 (Table 2.2b, Figure 2.1b), and the maximum valency number observed in several large vendor libraries was 5 (Rupp et al., 2007). Moreover, the average vertex degree was consistently slightly above 2, both with and without hydrogens, which we attribute to the dominance of carbon ring systems in such data sets. Note that most metal-based therapeutics (Hambley, 2007) can be modeled with graphs of vertex degree 7 or less. The small vertex degrees of molecular graphs admit algorithms which would otherwise be infeasible, e. g., algorithms with runtime exponential in the vertex degree and a small constant factor.

[4]Highest possible coordination numbers are related to the generalized Gregory-Newton problem of kissing spheres. Excluding fullerenes and similar structures, coordination numbers up to 12 can be realized in solid phases. For liquid and gas phases, higher coordination numbers are possible. See Hermann et al. (2007) for further information and an example ($PbHe_{15}^{2+}$ in gas phase).

2.2 Graph kernels

The molecular graph is an established and intuitive structured representation of molecules. Several *graph kernels*, i. e., positive definite measures of similarity between graphs, were introduced for direct comparison of (molecular) graphs for kernel-based learning. A *complete* graph kernel is injective modulo graph isomorphism, i. e., it separates all non-isomorphic graphs. The computation of inner products in the feature space indexed by all subgraphs, which would allow such kernels, is NP-hard[5] (Gärtner et al., 2003). Graph kernels therefore trade in separation capability for computational efficiency. For further information on graph kernels, see Borgwardt (2007); Vishwanathan et al. (2010).

Note that graph kernels are different from, but related to, kernels with graph-structured input spaces, i. e., kernels between vertices of a graph, such as diffusion kernels (Kondor and Vert, 2004), the regularized Laplacian kernel (Smola and Kondor, 2003), or, the von Neumann kernel (Kandola et al., 2003).

2.2.1 Convolution kernels

Many kernels for structured data, including graph kernels, are based on the idea of convolution kernels (Haussler, 1999). Assume that a sample $x \in \mathcal{X}$ can be decomposed into parts $x_1, \ldots, x_d \in \mathcal{X}_1, \ldots, \mathcal{X}_d$, e. g., a decomposition of a graph into subgraphs. The relation R indicates possible decompositions, where $R(x, x_1, \ldots, x_d)$ means that x can be decomposed into $x_1, \ldots, x_d$. Given positive definite kernels $k_i : \mathcal{X}_i \times \mathcal{X}_i \to \mathbb{R}$, $1 \leq i \leq d$, the *convolution kernel*

$$(k_1 \times \cdots \times k_d)(x, x') = \sum_R \prod_{i=1}^{d} k_i(x_i, x_i') \tag{2.8}$$

is positive definite for finite R (Haussler, 1999). The sum runs over all decompositions indicated by R; if a sample can not be decomposed, the sum is zero. Convolution kernels can be generalized to *mapping kernels* (Shin and Kuboyama, 2008), where the sum in Equation 2.8 is over transitive subsets of the cross product. Random walk-based graph kernels, tree-based graph kernels, and cyclic pattern kernels are convolution kernels.

2.2.2 Random walk kernels

Label sequence kernels (Gärtner et al., 2003; Kashima et al., 2004) are based on the similarity of random walks on graphs. They can be seen as kernels on label sequences marginalized with respect to these random walks, and are also called *marginalized graph kernels*. Specific kernels differ in the employed random walk model and the kernels used to compare the vertex and edge labels.

Random walks

A *random walk* on a graph $G = (V, E)$ can be constructed by first choosing a start vertex $x_1 \in V$ according to an initial probability distribution p_s; subsequent vertices x_i are chosen from the neighbors of x_{i-1} according to a transitional probability distribution p_t conditional on x_{i-1}, or, the random walk ends with probability $p_q(x_{i-1})$. Note that

[5] Even approximating a complete graph kernel is as hard as the graph isomorphism problem (Ramon and Gärtner, 2003), whose complexity is unknown but believed to be between P and NP (Johnson, 2005). In general, one can not expect graph kernels to efficiently learn concepts that are hard to compute.

label sequences of infinite length can occur. The probability of a random walk instance $x = x_1 \cdots x_l$ is

$$\mathbb{P}(x \,|\, G) = p_s(x_1)\Big(\prod_{i=2}^{l} p_t(x_i|x_{i-1})\Big)p_q(x_l). \tag{2.9}$$

A default choice for p_s, p_t, and p_q is to set $p_s(v_i) = |V|^{-1}$, $p_q(v_i) = c$ for a small constant $0 < c \leq 1$, and, $p_t(v_i|x_{i-1}) = \frac{1-c}{d(x_{i-1})}$, where $d(v)$ is the (out-)degree of v.

Label sequences

The *label sequence* $h_x = \text{label}\,(x_1)\,\text{label}\,(\{x_1, x_2\})\,\text{label}\,(x_2)\cdots\text{label}\,(x_l)$ is the sequence of alternating vertex and edge labels generated by traversing the random walk x (Figure 2.2). The probability of a label sequence is the sum of the probabilities of all random walks that generate it,

$$\mathbb{P}(h \,|\, G) = \sum_{x} 1_{\{h_x = h\}}\mathbb{P}(x \,|\, G)\,. \tag{2.10}$$

Graph kernel

A kernel on label sequences of equal length is given by the product of the label kernels,

$$k_z(h_x, h'_x) = k_v(h_1, h'_1)\prod_{i=1}^{l-1} k_e(h_{2i}, h'_{2i})k_v(h_{2i+1}, h'_{2i+1}). \tag{2.11}$$

For different lengths, $k_z(h_x, h'_x) = 0$. A *label sequence kernel* for graphs is given by the expectation of k_z over all possible label sequences:

$$k(G, G') = \sum_{h,h'} k_z(h, h')\mathbb{P}(h \,|\, G)\,\mathbb{P}\big(h' \,\big|\, G'\big)\,. \tag{2.12}$$

Kashima et al. (2004) show that for non-negative k_v and k_e, Equation 2.11, and therefore Equation 2.12, is positive definite. The latter is an example of a *marginalized kernel* (Tsuda et al., 2002), i. e., a kernel between visible and hidden variables — here, graphs and random walks — computed by taking the expectation over the hidden variables.

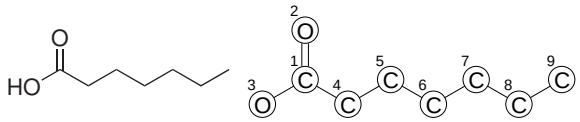
An alternative formulation using matrix power series and the product graph is given by Borgwardt et al. (2007), based on the observation that a walk in the product graph $G_\times$ corresponds to two walks, one in G and one in G'. Let $\boldsymbol{W} \in \mathbb{R}^{|V||V'|\times|V||V'|}$,

$$\boldsymbol{W}_{\substack{(i-1)|V|+j,\\(i'-1)|V'|+j'}} = \begin{cases} k_e\big(\{v_i, v_j\}, \{v'_{i'}, v'_{j'}\}\big) & \text{if } \{v_i, v_j\} \in E \wedge \{v'_{i'}, v'_{j'}\} \in E' \\ 0 & \text{otherwise,} \end{cases} \tag{2.13}$$

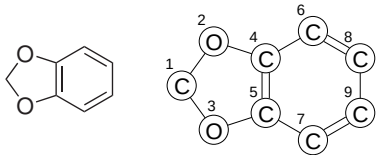
denote the *weight matrix* of $G_\times$, and define start and stop distributions on $G_\times$ by letting $\boldsymbol{p}_\times = \boldsymbol{p} \otimes \boldsymbol{p'}$ and $\boldsymbol{q}_\times = \boldsymbol{q} \otimes \boldsymbol{q'}$. An (edge) label sequence kernel is given by

$$k(G, G') = \sum_{i \geq 0} \lambda^i\, \boldsymbol{q}_\times^T \boldsymbol{W}^i \boldsymbol{p}_\times = \boldsymbol{q}_\times^T (\boldsymbol{I} - \lambda \boldsymbol{W})^{-1} \boldsymbol{p}_\times, \tag{2.14}$$

where $\lambda > 0$ is a constant small enough to ensure convergence. Note that vertex labels can be encoded into edge labels, e. g., by using edge labels from $\Sigma_V \times \Sigma_E \times \Sigma_V$, where Σ_V and Σ_E are the sets of vertex and edge labels, respectively.



(a) Heptanoic acid (structure graph, left, and annotated graph, right).



(b) Benzodioxole (structure graph, left, and annotated graph, right).

Figure 2.2 Random walks on molecular graphs. Considering only element type vertex labels, the random walks 12131456 (a) and 53124689 (b) both produce the same label sequence COCOCCCC. Taking bond type edge labels into account (s = single, d = double) leads to different label sequences CdOdCsOsCsCsCsC and CsOsCsOsCsCdCsC of the same random walks. The tottering walks 13131414 and 42424646 reproduce the first label sequence, but use only three vertices.

Computation

For acyclic graphs, e. g., reduced molecular graphs (Xu and Johnson, 2001, 2002), Equation 2.12 can be computed using topological sorting and dynamic programming in time $O\big(c\,c'\,|V|\,|V'|\big)$, where c, c' are the maximum vertex degrees in G and G'.

Molecular graphs themselves are cyclic due to, e. g., carbon ring structures. For cyclic graphs, Equation 2.12 can be computed by solving a system of linear equations, or, equivalently, by inverting a sparse $|V|\,|V'|\times|V|\,|V'|$ matrix. In both cases, the number of non-zero coefficients is upper-bounded by $c\,c'\,|V|\,|V'|$. For molecular graphs, $c\,c' \leq 49$, and, in almost all scenarios, even $c\,c' \leq 25$. The solution exists if a convergence condition on the involved probabilities and kernels is met. For random walk models with constant termination probabilities $p_q(\cdot) = \gamma$, the requirement is

$$k_v(\cdot,\cdot)k_e(\cdot,\cdot) < \frac{1}{(1-\gamma)^2}, \tag{2.15}$$

which is met by vertex and edge label kernels with $0 \leq k_v, k_e \leq 1$. The solution can be computed using matrix power series (Gärtner, 2003), an iterative method (Kashima et al., 2004), the Sylvester or Lyapunov equation, or, conjugate gradient methods (Borgwardt et al., 2007).

Tottering

A random walk $x = x_1 \cdots x_l$ can immediately revisit a vertex, i. e., $x_i = x_{i+2}$ for some $i \in [1, l-2]$, a behavior called *tottering* (Mahé et al., 2004, Figure 2.2). Such excursions are likely to be uninformative and to add noise to the model, especially because the ratio of tottering to non-tottering walks increases rapidly. Tottering can be prevented by switching to second-order Markov random walks, i. e., by conditioning the transition probabilities on the last two visited vertices:

$$\mathbb{P}(x \,|\, G) = p_s(x_1)p_t(x_2|x_1)\Big(\prod_{i=3}^{l} p_t(x_i|x_{i-2}, x_{i-1})\Big)p_q(x_l). \tag{2.16}$$

As a default, one can choose p_s, p_q, and $p_t(x_i|x_{i-1})$ as before, set $p_t(v_i|x_{i-2}, x_{i-1}) = \frac{1-c}{d(x_{i-1})-1}$ for $v_i \neq x_{i-2}$, and 0 otherwise. The corresponding label sequence kernel can be computed using the algorithms for Equation 2.12 on a transformed graph $G'' = (V'', E'')$.

The latter is constructed by inserting an additional vertex for each (directed) edge in the original graph $G = (V, E)$,

$$V'' = V \cup E, \tag{2.17}$$

$$\begin{aligned} E'' = &\left\{ \big(v, (v,u)\big) \;\middle|\; v \in V, (v,u) \in E \right\} \cup \\ &\left\{ \big((u,v),(v,w)\big) \;\middle|\; (u,v),(v,w) \in E, u \neq w \right\}, \end{aligned} \tag{2.18}$$

together with an appropriate labeling

$$\operatorname{label}\big(v''\big) = \begin{cases} \operatorname{label}(v'') & \text{if } v'' \in V \\ \operatorname{label}(v) & \text{if } v'' = (u,v) \in E \end{cases}, \tag{2.19}$$

$$\operatorname{label}\big((u'', v'')\big) = \operatorname{label}\big(v''\big) \text{ for } u'' \in V \cup E \wedge v'' \in E. \tag{2.20}$$

Mahé et al. (2004) show that there is a bijection between the non-tottering walks on G and the tottering walks on G''. The graph transformation increases the complexity of computing the label sequence kernel by a factor of

$$\frac{\big(|V| + |E|\big)\big(|V'| + |E'|\big)}{|V|\,|V'|}, \tag{2.21}$$

with $G = (V, E)$ and $G' = (V', E')$ the original input graphs (Mahé and Vert, 2009).

Variants

An early variant of random walk-based graph kernels, where only walks of equal length with matching first and last vertex label were counted, was introduced by Gärtner (2003). Later, extensions to transition graphs were introduced by Gärtner et al. (2003), e. g., to Markov chains (Diaconis, 2009), where edges are labeled with transition probabilities, and, to non-contiguous label sequences, where gaps are allowed (but penalized) when matching label sequences. Random walk-based graph kernels can also be extended to graphs with multiple edges (Kashima et al., 2004). Contextual information can be embedded into the labels using the Morgan index, which improves computation time by decreasing the number of common paths while still giving comparable performance on test data sets (Mahé et al., 2004).

2.2.3 Tree pattern kernels

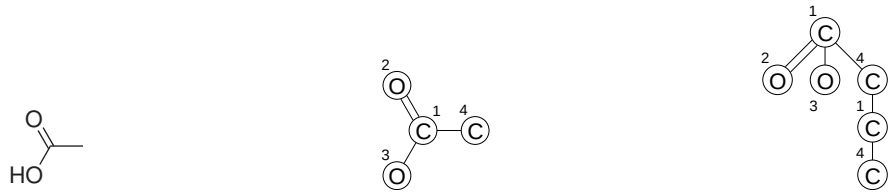
Tree-based graph kernels (Mahé and Vert, 2009) are based on the idea of comparing subtrees of the graphs.

Tree patterns

Let $G = (V, E)$ be a graph and let $T = (W, F)$, $W = \{w_1, \ldots, w_t\}$ be a rooted directed tree. A *tree pattern* of G with respect to T consists of vertices $v_1, \ldots, v_t \in G$ such that

$$\begin{aligned} &\operatorname{label}(v_i) = \operatorname{label}(w_i) \text{ for } 1 \leq i \leq t \\ \text{and } &\{v_i, v_j\} \in E \wedge \operatorname{label}\big(\{v_i, v_j\}\big) = \operatorname{label}\big((w_i, w_j)\big) \text{ for } (w_i, w_j) \in W \\ \text{and } &j \neq k \Leftrightarrow v_j \neq v_k \text{ for } (w_i, w_j), (w_i, w_k) \in W \end{aligned} \tag{2.22}$$

Each vertex in tree T is assigned a vertex from graph G such that edges and labels match. The $v_1, \ldots, v_t$ need not be distinct, as long as vertices assigned to sibling vertices in T are distinct (Figure 2.3). The *tree pattern counting function* $\psi(G, T)$ returns the number of times the tree pattern T occurs in the graph G, i. e., the number of distinct tuples $(v_1, \ldots, v_t)$ which are tree patterns of T in G.



(a) Acetic acid. **(b)** Annotated graph. **(c)** Tree pattern.

Figure 2.3 Tree patterns. Shown are the structure graph of acetic acid (a), a corresponding annotated graph (b), and, a tree pattern contained in it (c). The numbers in (c) indicate the assigned vertices from (b). Note that vertices 1 and 4 appear twice.

Graph kernel

Let $G = (V, E)$, $G' = (V', E')$ be graphs, let $\mathcal{T}$ be a set of trees, and, let $w : \mathcal{T} \to \mathbb{R}_+$ weight the trees in $\mathcal{T}$. Then

$$k(G, G') = \sum_{T \in \mathcal{T}} w(T)\psi(G, T)\psi(G', T) \tag{2.23}$$

is positive definite as it corresponds to a weighted inner product in the feature space indexed by the trees in $\mathcal{T}$.

Balanced trees

Let $\mathcal{B}_h$ denote the set of balanced trees of order h. By weighting tree patterns according to their size or their branching cardinality, two different kernels are derived, the *size-based balanced tree-pattern kernel of order h*,

$$k^h_{\text{size}}(G, G') = \sum_{T \in \mathcal{B}_h} \lambda^{|T|-h}\psi(G, T)\psi(G', T), \tag{2.24}$$

where $|T|$ denotes the number of vertices in T, and, the *branching-based balanced tree-pattern kernel of order h*,

$$k^h_{\text{branch}}(G, G') = \sum_{T \in \mathcal{B}_h} \lambda^{\text{branch}(T)}\psi(G, T)\psi(G', T), \tag{2.25}$$

where branch(T) denotes the branching cardinality of T. The parameter λ controls the weight put on complex tree patterns: more weight is put on them for $\lambda > 1$, and less for $\lambda < 1$. Since $|T| - h \geq \text{branch}(T)$, the weighting is more pronounced in Equation 2.24 than in Equation 2.25.

In the limit of $\lambda \to 0$, only linear trees have non-zero weight, and the two kernels converge to the walk counting kernel

$$k(G, G') = \sum_{\substack{w \in \mathcal{W}_{h-1}(G) \\ w' \in \mathcal{W}_{h-1}(G')}} 1_{\{\text{label}(w)=\text{label}(w')\}}, \tag{2.26}$$

where $\mathcal{W}_{h-1}(G)$ denotes the set of all walks of length $h - 1$ in G.

General trees

The branching-based kernel of Equation 2.25 can be extended to arbitrary trees of depth up to h by

$$k_{\text{branch}}^{\leq h}(G, G') = \sum_{T \in \mathcal{T}_{\leq h}} \lambda^{\text{branch}(T)} \psi(G, T) \psi(G', T), \tag{2.27}$$

where $\mathcal{T}_{\leq h}$ denotes the set of all trees with depth h or less. The feature space of this kernel is indexed by all trees in $\mathcal{T}_{\leq h}$. Although Equation 2.24 can be generalized in the same manner, this loses the proper weighting scheme.

Computation

Mahé and Vert (2009) show how to compute Equations 2.24, 2.25 and 2.27 using dynamic programming and the notion of neighborhood matching sets in time $O\big(|V|\,|V'|\,h\,d^{2d}\big)$, where d denotes the maximum vertex degree.

Tottering

Recall that graph vertices may be used more than once when matching tree patterns (Equation 2.22). This causes the equivalent of tottering in random walk-based graph kernels. In the computation of Equations 2.24, 2.25 and 2.27, all trees up to a given depth are enumerated by recursively extending depth 2 trees. In this process, it is possible that a vertex appears both as a parent and as a child of another vertex.

Mahé and Vert (2009) modify the tree pattern counting function ψ to prevent tottering by introducing additional constraints. In order to retain the algorithms for the computation of Equations 2.24, 2.25 and 2.27, they transform the input graphs, replacing edges with additional vertices. Their transformation does not change the maximum vertex out-degree, but increases the size of the graphs, leading to an increase in runtime by a factor of

$$\frac{(|V| + |E|)(|V'| + |E'|)}{|V|\,|V'|}. \tag{2.28}$$

2.2.4 Cyclic pattern kernels

Introduced by Horváth et al. (2004), *cyclic pattern graph kernels* are based on the idea of mapping graphs to sets of cyclic and tree pattern strings that are compared using the intersection kernel.

Intersection kernel

Let U be a set, and let 2^U denote the set of all subsets of U. The *intersection kernel*

$$k_\cap : 2^U \times 2^U \to \mathbb{R}, \qquad k_\cap(S, S') = \big|S \cap S'\big| \tag{2.29}$$

is positive definite since for all c_i, $c_j \in \mathbb{R}$

$$\sum_{i,j} c_i c_j \,|S_i \cap S_j| = \sum_{i,j} c_i c_j \sum_{u \in U} 1_{\{u \in S_i\}} 1_{\{u \in S_j\}} = \sum_{u \in U} \Big(\sum_i c_i 1_{\{u \in S_i\}} \Big)^2 \geq 0. \tag{2.30}$$

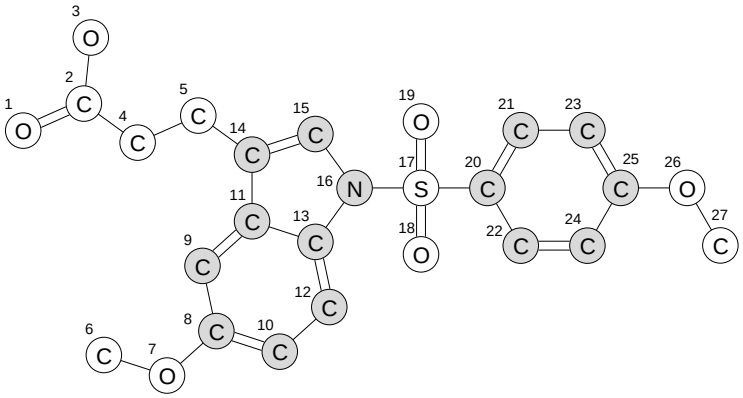


Figure 2.4 Cyclic and tree patterns of indeglitazar (Compound 44, p. 153). Shown are vertices belonging to simple cycles (shaded) and to bridges (white).

Cyclic and tree patterns

A subgraph is a *simple cycle* if it is connected and each vertex has degree 2. Let $S(G)$ denote the set of simple cycles in a graph G. An edge not belonging to a simple cycle is a *bridge*. We denote the subgraph of all bridges in G, which is a forest, by $\mathcal{B}(G)$ (Figure 2.4). Let π be a mapping, computable in polynomial time, from the set of labeled simple cycles and trees to label strings that is injective modulo isomorphism. Note that such a mapping can always be constructed (Horváth et al., 2004; Zaki, 2005). The sets of *cyclic* and *tree patterns* are given by

$$\mathcal{C}(G) = \big\{\pi(C) \,\big|\, C \in S(G)\big\}, \tag{2.31}$$

$$\mathcal{T}(G) = \big\{\pi(T) \,\big|\, T \text{ is a connected component of } \mathcal{B}(G)\big\}. \tag{2.32}$$

The *cyclic pattern kernel* is given by

$$k(G, G') = k_\cap\big(\mathcal{C}(G), \mathcal{C}(G')\big) + k_\cap\big(\mathcal{T}(G), \mathcal{T}(G')\big). \tag{2.33}$$

Computing Equation 2.33 is at least as hard as counting simple cycles in a graph, which is computationally not tractable if P $\neq$ NP (Flum and Grohe, 2004). For graphs with a small number of simple cycles, Equation 2.33 can be computed via enumeration of $\mathcal{B}, \mathcal{T}, S, \mathcal{C}$ (Horváth et al., 2004).

Bounded tree width

The number of cyclic and tree patterns in a graph can be exponential in $|V|$ (Horváth et al., 2004), leading to computational infeasibility of the cyclic pattern kernel for general graphs. The restriction of inputs to graphs with few simple cycles can be relaxed to graphs of bounded treewidth, for which many NP-complete problems become tractable (Bodlaender, 1993). For graphs of constant bounded treewidth, Equation 2.33 can be computed in time polynomial in $\max\big\{|V|, |V'|, |\mathcal{C}(G)|, |\mathcal{C}(G')|\big\}$ (Horváth, 2005).

Relevant cycles

An alternative relaxation is to consider a different class of cycles. Horváth (2005) uses algebraic graph theory to compute the cyclic pattern kernel on monotone increasing subsets of simple cycles generated by relevant cycles (Plotkin, 1971), with a runtime bound similar to the one in the previous paragraph, but with different cyclic patterns. While the number of relevant cycles is typically cubic in $|V|$ for molecular graphs (Gleiss and Stadler, 1999), it is still exponential in the worst case.

2.2.5 Optimal assignment kernels

Optimal assignment graph kernels (Fröhlich et al., 2005a) are based on the idea of an optimal assignment of vertices between graphs.

Optimal assignments

Let $G = (V, E)$ and $G' = (V', E')$ be two undirected labeled graphs. Based upon a measure $k_{G,G'}$ of similarity[6] between the (labels of) vertices of G and G', the *optimal assignment kernel* injectively assigns the vertices of the smaller graph to vertices of the larger graph such that the total similarity between the assigned vertices is maximized:

$$k_{\mathrm{oa}}(G, G') = \begin{cases} \max\limits_{\pi} \sum\limits_{i=1}^{|V|} k_{G,G'}(v_i, v'_{\pi(i)}) & \text{if } |V| \leq |V'|, \\ \max\limits_{\pi} \sum\limits_{i=1}^{|V'|} k_{G,G'}(v_{\pi(i)}, v'_i) & \text{otherwise.} \end{cases} \tag{2.34}$$

The maximum in the first case is over all possible assignments π of the vertices in V to vertices in V', i. e., all prefixes of length $|V|$ of permutations of size $|V'|$; in the second case, the roles of V and V' are exchanged. Since $k_{\mathrm{oa}}(G, G') = k_{\mathrm{oa}}(G', G)$, we assume from now on without loss of generality that $|V| < |V'|$. Equation 2.34 then becomes

$$k_{\mathrm{oa}}(G, G') = \max_{\pi} \sum_{i=1}^{|V|} k_{G,G'}(v_i, v'_{\pi(i)}). \tag{2.35}$$

To prevent the value of the kernel depending on the size $|V|$ of the smaller graph, the *normalized optimal assignment kernel* with values in $[0, 1]$ may be used:

$$k_{\mathrm{noa}}(G, G') = \frac{k_{\mathrm{oa}}(G, G')}{\sqrt{k_{\mathrm{oa}}(G, G) k_{\mathrm{oa}}(G', G')}}. \tag{2.36}$$

Positive definiteness

Optimal assignment kernels were proposed by Fröhlich et al. (2005a,b) in the context of machine learning and cheminformatics. However, their proof that k_{oa} is positive definite is wrong, as the inequality at the end of their proof is bounded into the wrong direction. Furthermore, their proof idea (induction over matrix size based on properties of 2×2 matrices) fails, as there are 3×3 matrices which are not positive definite, although all their 2×2 submatrices are; consider, e. g., the following matrices and their smallest eigenvalues (shown below each matrix):

$$\begin{array}{cccccc} \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} & \left(\begin{smallmatrix} 1 & 1 \\ 1 & 1 \end{smallmatrix}\right) & \left(\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right) & \left(\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right) & \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right) & \\ 0 & 1 & 1 & 1 & & \\ \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right) & \left(\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right) & \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right) & \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right) & \left(\begin{smallmatrix} 1 & 1 \\ 1 & 1 \end{smallmatrix}\right) & . \\ 1 - \sqrt{2} & 1 & 1 & 1 & 1 & 0 \end{array}$$

The 3×3 matrix is not positive definite, as its smallest eigenvalue is negative, but all submatrices of size 2×2 are.

In general, whether an optimal assignment is positive definite or not depends on the underlying vertex similarity $k_{G,G'}$. Vert (2008) shows that k_{oa} is positive definite for $k_{G,G'}(v, v') = 1$, and gives an example of $k_{G,G'}$ for which k_{oa} is not positive definite.

[6] Note that we allow $k_{G,G'} : V \times V' \to \mathbb{R}_+$ to depend on G and G'.

Computation

Algorithmically, Equation 2.35 can be computed in two steps: First, the matrix of pairwise vertex similarities

$$\boldsymbol{X} = \left(k_{G,G'}(v_i, v'_j)\right)_{\substack{i=1,\ldots,|V| \\ j=1,\ldots,|V'|}} \tag{2.37}$$

is calculated. Then, an optimal assignment (one column assigned unambiguously to each row) is computed, e. g., using the Kuhn-Munkres assignment algorithm (also Hungarian algorithm; Kuhn, 1955; Munkres, 1957; Bourgeois and Lassalle, 1971) in time $O(|V'|^3)$.

Similarity matrix

Let k_v and k_e be non-negative bounded kernels defined on the labels of vertices and edges, respectively. Default choices for k_v and k_e are the *Dirac kernel*

$$k(x, y) = 1_{\{x=y\}} \tag{2.38}$$

for discrete labels and the Gaussian kernel (p. 30) for continuous labels.

Fröhlich et al. (2005b) start by defining $k_{G,G'}(v, v')$ using the mean similarity $k_0(v_i, v'_j)$ between all neighbors of v_i and v'_j,

$$k_{G,G'}(v_i, v'_j) = k_v(v_i, v'_j) + k_0(v_i, v'_j), \tag{2.39}$$

$$k_0(v_i, v'_j) = \frac{1}{|v_i|\,|v'_j|} \sum_{\substack{v \in n(v_i) \\ v' \in n(v'_j)}} k_v\big(v, v'\big) k_e\big(\{v_i, v\}, \{v'_j, v'\}\big), \tag{2.40}$$

where $n(v)$ denotes the set of all neighbors of vertex v. They extend this definition to include all neighbors up to a given topological distance $L \geq 0$ using the recursion

$$k_r(v_i, v'_j) = \frac{1}{|v_i|\,|v'_j|} \sum_{\substack{v \in n(v_i) \\ v' \in n(v'_j)}} k_{r-1}(v, v') \tag{2.41}$$

and a decay function $\gamma : \mathbb{N}_0 \to \mathbb{R}$ to limit the influence of distant neighbors:

$$k_{G,G'}(v_i, v'_j) = k_v(v_i, v'_j) + \sum_{r=0}^{L} \gamma(r) k_r(v_i, v'_j). \tag{2.42}$$

Since vertex degrees are bounded in molecular graphs, Equations 2.40, 2.41 and 2.42 can be computed in constant time for finite topological distances L. For $\gamma(r) = \gamma^r$ with $0 < \gamma < 1$, the sum in Equation 2.42 converges for $L \to \infty$. In another publication, they replace the mean with an optimal assignment of the neighbors (Fröhlich et al., 2005a).

2.2.6 Other graph kernels

Several other approaches to graph kernels have been proposed.

Complement graph

Let $k(G, G')$ denote the random walk kernel from Equation 2.14. Borgwardt et al. (2007) propose to use the composite graph kernel

$$k_c(G, G') = k(G, G') + k(\bar{G}, \bar{G}'), \tag{2.43}$$

where $\bar{G} = \big(V, V \times V \setminus E\big)$ denotes the complement graph of G. This kernel also takes the absence of an edge in both graphs into consideration, which is useful for modeling protein-protein interaction networks.

Fingerprint kernels

Ralaivola et al. (2005) propose kernels based on common walks of bounded length, which they compute explicitly using depth-first search, molecular fingerprinting (Raymond and Willett, 2002), and suffix trees (Ukkonen, 1995). Let $\mathcal{P}$ denote the set of all labeled walks of bond-length up to d, let $\phi_w(G)$ indicate either the presence of the walk w in G (if binary features are used) or the number of times w occurs in G (if counting features are used). Let $\phi(G) = \big(\phi_w(G)\big)_{w \in \mathcal{P}}$ denote the mapping into the feature space indexed by such walks, and let $k(G, G') = \langle \phi(G), \phi(G') \rangle$ denote the inner product in this space. The authors use the derived *Tanimoto kernel*

$$k_t(G, G') = \frac{k(G, G')}{k(G, G) + k(G', G') - k(G, G')} \tag{2.44}$$

and the related *minmax kernel*

$$k_m(G, G') = \frac{\sum_{w \in \mathcal{P}} \min\big\{\phi_w(G), \phi_w(G')\big\}}{\sum_{w \in \mathcal{P}} \max\big\{\phi_w(G), \phi_w(G')\big\}}. \tag{2.45}$$

Enumeration by depth-first search of all walks of length up to d, starting from each vertex, takes time in $O\big(|V|\,|E|\big)$, generation of the suffix tree takes time in $O\big(d\,|V|\,|E|\big)$, and, enumeration and lookup of all walks in G' takes time in $O\big(d\,|V'|\,|E'|\big)$, yielding an overall runtime complexity of $O\big(d\big(|V|\,|E| + |V'|\,|E'|\big)\big)$.

Path kernels

The *all-paths kernel* (Borgwardt and Kriegel, 2005) is related to random walk kernels (Subsection 2.2.2), but uses paths instead of walks, i. e., walks without repetition of vertices. Let $\mathcal{P}, \mathcal{P}'$ denote the set of all paths on the graphs G, G', and let k_p denote a kernel on paths. The all-paths kernel

$$k(G, G') = \sum_{\substack{p \in \mathcal{P} \\ p' \in \mathcal{P}'}} k_p(p, p') \tag{2.46}$$

is positive definite, but its computation is NP-hard. Restriction to shortest paths renders Equation 2.46 computationally feasible, and can be achieved by considering transformed graphs $\tilde{G} = (V, \tilde{E})$, $\tilde{G}' = (V', \tilde{E}')$, with edges between all vertices connected by a path. The edges are labeled with shortest path distances, which are computable in polynomial time (all-pairs shortest-path problem; Cormen et al., 2001), resulting in an overall runtime of $O\big(|V|^2\,|V'|^2\big)$.

Edit distance kernels

The *edit distance* (Levenshtein, 1966; Gusfield, 1997) between two graphs is the minimum number of vertex and edge insertions, deletions, and substitutions required to transform one graph into another. Graph edit distance computation requires time exponential in the number of vertices, but can be efficiently approximated (Neuhaus and Bunke, 2004). Let $d(G, G')$ denote the non-negative and symmetric edit distance between the two graphs G and G'. Neuhaus and Bunke (2006) propose the use of

$$k_{G_0}(G, G') = \frac{1}{2}\big(d^2(G, G_0) + d^2(G', G_0) - d^2(G, G')\big), \tag{2.47}$$

where the graph G_0 is called *zero graph* because it has the role of origin. Equation 2.47 relates the direct distance between G and G' to the distance between G and G' via the zero graph. Extending it to multiple graphs using a set I of zero graphs as

$$k_I^+(G, G') = \sum_{G_0 \in I} k_{G_0}(G, G') \quad \text{and} \quad k_I^*(G, G') = \prod_{G_0 \in I} k_{G_0}(G, G') \tag{2.48}$$

improves performance in practice. Equations 2.47 and 2.48 are positive definite if $-d^2$ is conditionally positive definite, which is not the case for edit distances in general. An advantage of edit distances is their robustness against noise in the input graphs.

2.2.7 Applications in cheminformatics

Kernels on structured data, and graphs in particular, have been successfully applied to various problems in cheminformatics, including ligand-based virtual screening.

Structured data

An example of an application in cheminformatics using kernels on non-graph structured data is given by Gärtner et al. (2004), who use structured data kernels and a 1-nearest neighbor classifier to improve classification accuracy to 95 % in diterpene structure elucidation from ^{13}C nuclear magnetic resonance spectra, an improvement of 10 % compared to the state of the art at that time.

Graphs

Examples of applications in bio- and cheminformatics using kernels on graphs:

- Borgwardt et al. (2005) use a modified random walk kernel, hyperkernels (Ong et al., 2005), and support vector machines for protein function prediction. Classification accuracy for enzymes versus non-enzymes was 84.04 %, a significant improvement over previous vector-based approaches.
- Menchetti et al. (2005) propose weighted decomposition kernels for molecules, using local topology and graph complements. They achieve performance similar to other graph-based approaches in the predictive toxicology challenge (Toivonen et al., 2003), and, discriminating active versus moderately active compounds in a screen for activity against human immunodeficiency virus. Ceroni et al. (2007) later incorporate spatial information into weighted decomposition kernels for molecules.
- Borgwardt et al. (2006) employ graph representations of proteins, random walk kernels, and kernel mean discrepancy to statistically test whether two protein samples are from

the same distribution, with excellent results (no errors on a significance level of 0.05). Gretton et al. (2006) improve the test, and apply it to protein homology detection on a KDD cup (Caruana and Joachims, 2004) data set, with state of the art performance.

- Airola et al. (2008) introduce the all-dependency-paths graph kernel for the automated extraction of protein-protein interactions from scientific literature (text mining), with state of the art results (ROC AUC between 0.80 and 0.85 on five public text corpora).

For further examples, see Jain et al. (2005).

2.3 Iterative similarity optimal assignment graph kernel

Quantitative measures of graph similarity, including, but not limited to, the graph kernels of the previous section, are required in many contexts. Consequently, the graph isomorphism problem (Köbler et al., 1993; Johnson, 2005), as well as its generalizations, edit distance and maximum common subgraph / minimum common supergraph, have been intensively investigated (Conte et al., 2004). Starting from iterative methods for the similarity of general graphs, we develop a vertex scoring scheme tailored to molecular graphs. Combining this scheme with optimal assignment kernels, we introduce the iterative similarity optimal assignment graph kernel.

2.3.1 Iterative graph similarity

In one particular approach to graph similarity, vertices (and edges) in two graphs are considered similar if their respective neighborhoods are similar. This recursive definition naturally leads to iterative computation schemes for pairwise vertex (and edge) similarity scores, where initial similarities are repeatedly updated, propagating information according to the graph structures, until convergence occurs.

Several methods based on this approach have been developed, e. g., Kleinberg (1999); Melnik et al. (2002); Jeh and Widom (2002); Heymans and Singh (2003); Leicht et al. (2006); Zager and Verghese (2008). We essentially follow Zager and Verghese (2008) in our exposition.

Hub and authority scores in a single graph

The hypertext induced topic selection algorithm (HITS; Kleinberg, 1999) for scoring internet search queries is an iterative scheme for scoring the vertices of a single directed graph $G = (V, E)$. Each vertex v is associated with a *hub score* $h^{(k)}(v)$ and an *authority score* $a^{(k)}(v)$, where k is the iteration number. In each iteration, the hub score of a vertex is the sum of the authority scores of the vertices it points to, and its authority score is the sum of the hub scores of vertices pointing to it,

$$\tilde{h}^{(k)}(v) = \sum_{(v,u)\in E} a^{(k-1)}(u), \qquad \tilde{a}^{(k)}(v) = \sum_{(u,v)\in E} h^{(k-1)}(u), \tag{2.49}$$

constituting a mutually reinforcing relation between hubs and authorities. In each iteration, a normalization step is necessary to keep the scores from unlimited growth,

$$h^{(k)}(v) = \frac{\tilde{h}^{(k)}(v)}{\left\|\tilde{h}^{(k)}\right\|_2}, \qquad a^{(k)}(v) = \frac{\tilde{a}^{(k)}(v)}{\left\|\tilde{a}^{(k)}\right\|_2}. \tag{2.50}$$

Let $\boldsymbol{A}$ denote the adjacency matrix of G, let $v_1, \ldots, v_{|V|}$ denote its vertices, and set

$$\boldsymbol{x}^{(k)} = \left(h^{(k)}(v_1), \ldots, h^{(k)}(v_{|V|}), a^{(k)}(v_1), \ldots, a^{(k)}(v_{|V|})\right)^T. \tag{2.51}$$

Equation 2.49 can be written in matrix form,

$$\tilde{\boldsymbol{x}}^{(k)} = \begin{pmatrix} 0 & \boldsymbol{A} \\ \boldsymbol{A}^T & 0 \end{pmatrix} \boldsymbol{x}^{(k-1)} = \boldsymbol{M}\boldsymbol{x}^{(k-1)}, \qquad \boldsymbol{x}^{(k)} = \frac{\tilde{\boldsymbol{x}}^{(k)}}{\left\|\tilde{\boldsymbol{x}}^{(k)}\right\|_2}. \tag{2.52}$$

The convergence behavior of Equation 2.52 depends on spectral properties of $\boldsymbol{M}$ (Blondel et al., 2004): If $-\rho(\boldsymbol{M})$ is not an eigenvalue of $\boldsymbol{M}$, then

$$\lim_{k\to\infty} \boldsymbol{x}^{(k)} = \frac{\boldsymbol{\Pi}\boldsymbol{x}^{(0)}}{\left\|\boldsymbol{\Pi}\boldsymbol{x}^{(0)}\right\|_2}, \tag{2.53}$$

where $\boldsymbol{\Pi}$ is the orthogonal projector on the invariant subspace associated with the Perron root $\rho(\boldsymbol{M})$. If $-\rho(\boldsymbol{M})$ is an eigenvalue of $\boldsymbol{M}$, the odd and even iterates converge to different limits,

$$\lim_{k\to\infty} \boldsymbol{x}^{(2k)} = \frac{\boldsymbol{\Pi}\boldsymbol{x}^{(0)}}{\left\|\boldsymbol{\Pi}\boldsymbol{x}^{(0)}\right\|_2}, \qquad \lim_{k\to\infty} \boldsymbol{x}^{(2k+1)} = \frac{\boldsymbol{\Pi}\boldsymbol{M}\boldsymbol{x}^{(0)}}{\left\|\boldsymbol{\Pi}\boldsymbol{M}\boldsymbol{x}^{(0)}\right\|_2}. \tag{2.54}$$

In both cases, convergence limits depend on the initial value $\boldsymbol{x}^{(0)}$. Kleinberg (1999) assumes the case of Equation 2.53, an assumption not necessarily valid in practice, and sets $\boldsymbol{x}^{(0)} = \mathbf{1}_{|V|}$, ensuring convergence to the solution with largest possible sum norm $\|\cdot\|_1$.

Generalization to two graphs

Blondel et al. (2004) view Kleinberg's algorithm as a comparison of G with the prototype hub-authority graph $\left(\{h, a\}, \{(h, a)\}\right)$, ⓗ→ⓐ, where $h^{(k)}(v)$ and $a^{(k)}(v)$ are interpreted as the similarity of v to h and a, respectively.

They generalize this to a similarity measure between two arbitrary directed graphs $G = (V, E)$ and $G' = (V', E')$ by introducing a similarity score $x_{v,v'}$ for each pair of vertices $(v, v') \in V \times V'$ and updating according to

$$\tilde{\boldsymbol{x}}^{(k)}_{v,v'} = \sum_{\substack{(u,v)\in E \\ (u',v')\in E'}} \boldsymbol{x}^{(k-1)}_{u,u'} + \sum_{\substack{(v,u)\in E \\ (v',u')\in E'}} \boldsymbol{x}^{(k-1)}_{u,u'}, \tag{2.55}$$

which adds the similarities of all pairs of incoming and outgoing neighbors, respectively. Note that, as in Equation 2.54, a normalization step $\boldsymbol{x}^{(k)} = \tilde{\boldsymbol{x}}^{(k)} / \left\|\tilde{\boldsymbol{x}}^{(k)}\right\|_2$ is required. Equation 2.55 can be written in matrix form as

$$\tilde{\boldsymbol{X}}^{(k)} = \boldsymbol{A}^T \boldsymbol{X}^{(k-1)} \boldsymbol{A}' + \boldsymbol{A} \boldsymbol{X}^{(k-1)} \boldsymbol{A}'^T, \tag{2.56}$$

where $\boldsymbol{A}$ and $\boldsymbol{A}'$ are the adjacency matrices of G and G'. Concatenating the columns of $\boldsymbol{X}$ into $\mathrm{vec}(\boldsymbol{X})$ and using $\mathrm{vec}(\boldsymbol{C}\boldsymbol{X}\boldsymbol{D}) = (\boldsymbol{D}^T \otimes \boldsymbol{C})\,\mathrm{vec}(\boldsymbol{X})$ (Horn and Johnson, 1991, p. 254), where $\otimes$ denotes the Kronecker matrix product (also tensor or direct product), one gets

$$\tilde{\boldsymbol{x}}^{(k)} = (\boldsymbol{A}'^T \otimes \boldsymbol{A}^T + \boldsymbol{A}' \otimes \boldsymbol{A})\boldsymbol{x}^{(k-1)} = \boldsymbol{M}\boldsymbol{x}^{(k-1)}. \tag{2.57}$$

The convergence of this equation is subject to the same conditions as Equation 2.52. Blondel et al. (2004) set $\boldsymbol{x}^{(0)} = \mathbf{1}_{|V|\,|V'|}$ and use the limit of the even iterations.

Coupled vertex-edge scoring

Zager and Verghese (2008) extend this approach by introducing similarity scores for edges. They define two edges to be similar if their respective source and terminal vertices are similar, i. e., they couple edge similarity to vertex similarity. Let $\boldsymbol{S}, \boldsymbol{T} \in \{0,1\}^{|V|\times|E|}$ denote the *source-edge* and *terminus-edge* matrices,

$$\boldsymbol{S}_{v,e} = 1_{\{\exists u \in V\colon e=(v,u)\}}, \qquad \boldsymbol{T}_{v,e} = 1_{\{\exists u \in V\colon e=(u,v)\}}. \tag{2.58}$$

Note that $\boldsymbol{D_S} = \boldsymbol{S}\boldsymbol{S}^T$ and $\boldsymbol{D_T} = \boldsymbol{T}\boldsymbol{T}^T$ are diagonal, with the out- and in-degrees of the vertices on the diagonal, respectively. Let $\boldsymbol{y}_{e,e'}$ denote the edge score between $e \in E$ and $e' \in E'$. The coupled update equation

$$\begin{aligned}\tilde{\boldsymbol{y}}^{(k)}_{(u,v),(u',v')} &= \boldsymbol{x}^{(k-1)}_{u,u'} + \boldsymbol{x}^{(k-1)}_{v,v'} \\ \tilde{\boldsymbol{x}}^{(k)}_{v,v'} &= \sum_{\substack{(v,u)\in E\\(v',u')\in E'}} \boldsymbol{y}^{(k-1)}_{(v,u),(v',u')} + \sum_{\substack{(u,v)\in E\\(u',v')\in E'}} \boldsymbol{y}^{(k-1)}_{(u,v),(u',v')}\end{aligned} \tag{2.59}$$

updates the edge scores using the vertex scores and vice versa. Again, a normalization step is required in each iteration. The update can be given in matrix form,

$$\begin{aligned}\tilde{\boldsymbol{Y}}^{(k)} &= \boldsymbol{S}^T\boldsymbol{X}^{(k-1)}\boldsymbol{S}' + \boldsymbol{T}^T\boldsymbol{X}^{(k-1)}\boldsymbol{T}', \\ \tilde{\boldsymbol{X}}^{(k)} &= \boldsymbol{S}\boldsymbol{Y}^{(k-1)}\boldsymbol{S}'^T + \boldsymbol{T}\boldsymbol{Y}^{(k-1)}\boldsymbol{T}'^T,\end{aligned} \tag{2.60}$$

and, using column concatenation, as a matrix-vector product

$$\tilde{\boldsymbol{y}}^{(k)} = \big(\boldsymbol{S}'^T \otimes \boldsymbol{S}^T + \boldsymbol{T}'^T \otimes \boldsymbol{T}^T\big)\boldsymbol{x}^{(k-1)} = \boldsymbol{N}\boldsymbol{x}^{(k-1)}, \tag{2.61a}$$

$$\tilde{\boldsymbol{x}}^{(k)} = \big(\boldsymbol{S}' \otimes \boldsymbol{S} + \boldsymbol{T}' \otimes \boldsymbol{T}\big)\boldsymbol{y}^{(k-1)} = \boldsymbol{N}^T\boldsymbol{y}^{(k-1)}. \tag{2.61b}$$

Concatenating $\boldsymbol{x}$ and $\boldsymbol{y}$ into a single vector $\boldsymbol{z} = \binom{\boldsymbol{x}}{\boldsymbol{y}}$, Equation 2.59 can be expressed as a single matrix update:

$$\tilde{\boldsymbol{z}}^{(k)} = \begin{pmatrix} 0 & \boldsymbol{N}^T \\ \boldsymbol{N} & 0 \end{pmatrix} \boldsymbol{z}^{(k-1)} = \boldsymbol{M}\boldsymbol{z}^{(k-1)}. \tag{2.62}$$

Zager and Verghese (2008) show that for arbitrary $\boldsymbol{x}^{(0)}$ and $\boldsymbol{y}^{(0)} = \alpha\boldsymbol{N}\boldsymbol{x}^{(0)}$ with $\alpha > 0$, Equation 2.62 converges to a unique non-negative solution. Inserting Equation 2.61a into Equation 2.61b, and using $\boldsymbol{A} = \boldsymbol{S}\boldsymbol{T}^T$ and $(\boldsymbol{C}\otimes\boldsymbol{D})(\boldsymbol{E}\otimes\boldsymbol{F}) = \boldsymbol{C}\boldsymbol{E}\otimes\boldsymbol{D}\boldsymbol{F}$) gives

$$\tilde{\boldsymbol{x}}^{(k)} = \big(\boldsymbol{N}^T\boldsymbol{N}\big)\boldsymbol{x}^{(k-2)} = \big(\boldsymbol{A}'\otimes\boldsymbol{A} + \boldsymbol{A}'^T\otimes\boldsymbol{A}^T + \boldsymbol{D}'_{\boldsymbol{S}}\otimes\boldsymbol{D_S} + \boldsymbol{D}'_{\boldsymbol{T}}\otimes\boldsymbol{D_T}\big)\boldsymbol{x}^{(k-2)}, \tag{2.63}$$

the first part of which is identical to Equation 2.57. Coupled vertex-edge similarity therefore differs from the update of Blondel et al. (2004) in additional diagonal terms that amplify the scores of highly connected vertices.

2.3.2 Iterative similarity for molecular graphs

The coupled vertex-edge update (Equation 2.63) has several desirable properties: It converges to a unique limit, independent of the initialization value, and, it has a succinct matrix notation, making it amenable to analysis and enabling a simple implementation via iterated matrix multiplication.

With regard to the comparison of molecular graphs, however, it has several shortcomings: It is based exclusively on graph topology and does not take any labeling of vertices or bonds into account, which is indispensable for chemical similarity measures as the graph topology alone does not provide enough information. Molecular graph properties (Part 2.1.3), in particular bounded vertex degrees, are not exploited.

Update equation

We propose the following update equation that has the desirable properties of Equation 2.63, but is tailored towards the requirements of ligand-based virtual screening, i. e., the comparison of small labeled undirected graphs with bounded vertex degree:

$$\boldsymbol{X}^{(n)}_{v,v'} = (1-\alpha)k_v(v,v') + \alpha \max_{\pi} \frac{1}{|v'|} \sum_{\{v,u\}\in E} \boldsymbol{X}^{(n-1)}_{u,\pi(u)} k_e\big(\{v,u\},\{v',\pi(u)\}\big) \tag{2.64a}$$

for $|v| < |v'|$ and

$$\boldsymbol{X}^{(n)}_{v,v'} = (1-\alpha)k_v(v,v') + \alpha \max_{\pi} \frac{1}{|v|} \sum_{\{v',u'\}\in E'} \boldsymbol{X}^{(n-1)}_{\pi(u'),u'} k_e\big(\{v,\pi(u')\},\{v',u'\}\big) \tag{2.64b}$$

for $|v| \geq |v'|$. The maximum in Equation 2.64a is over all possible assignments of neighbors of v to neighbors of v', i. e., over all length $|v|$ prefixes of length $|v'|$ permutations; in Equation 2.64b, the roles of v and v' are exchanged. In other words, Equation 2.64 optimally assigns the neighbors of the vertex with smaller degree to the neighbors of the vertex with larger degree, based on the similarity values of the previous iteration. The parameter $\alpha \in (0,1)$ weights the constant and the recursive parts of the equation. A normalization step is not necessary.

Equation 2.64 obviously takes vertex and edge labels into account. In the following, we introduce a succinct matrix notation, prove that the corresponding iteration converges to a unique limit independent of the initialization value $\boldsymbol{X}^{(0)}$, and, show that it exploits the bounded degree of molecular graphs.

Matrix notation

To obtain Equation 2.64 in matrix form, note that the graph neighborhood structures are fixed for the computation. This renders $k_v(v,v')$, $|v|^{-1}$, $|v'|^{-1}$, $\big\{u \bigm| \{v,u\} \in E\big\}$, $\big\{u' \bigm| \{v',u'\} \in E'\big\}$, and $k_e(\{u,v\},\{u',v'\})$ constants depending only on u, v, u', v', and predetermines the case in Equation 2.64 for each combination of v and v'. Let $\boldsymbol{x}^{(n)} = \operatorname{vec}\big(\boldsymbol{X}^{(n)}\big)$, and let

$$\boldsymbol{k_v} = \big(k_v(v_1,v'_1), k_v(v_2,v'_1), \ldots, k_v(v_{|V|},v'_1), k_v(v_1,v'_2), \ldots, k_v(v_{|V|},v'_{|V'|})\big) \tag{2.65}$$

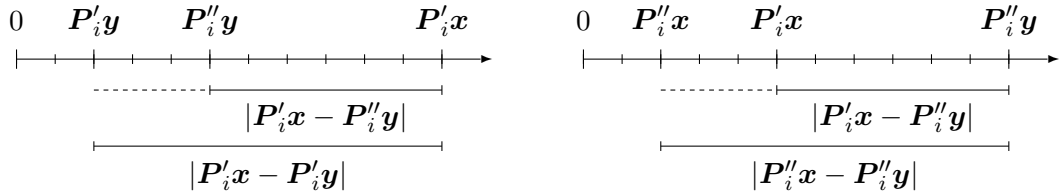
denote the corresponding vectorization of k_v. We encode the neighbor assignments of a single iteration into a $|V|\,|V'| \times |V|\,|V'|$ square matrix $\boldsymbol{P}$ as follows: Each row corresponds to a specific neighbor assignment in Equation 2.64, e. g., row $(j-1)|V|+i$, which corresponds to entry $\boldsymbol{X}^{(n)}_{i,j}$, contains one possible assignment of neighbors of v_i to neighbors of v'_j, or, vice versa, depending on $|v_i|$ and $|v'_j|$. The non-zero entries of $\boldsymbol{P}$ are the corresponding products of k_e and $|v'_j|^{-1}$ or $|v_i|^{-1}$, respectively. Equation 2.64 can be written as

$$\boldsymbol{x}^{(n)} = (1-\alpha)\boldsymbol{k_v} + \alpha \max_{\boldsymbol{P}} \boldsymbol{P}\boldsymbol{x}^{(n-1)}. \tag{2.66}$$

The maximum is over all matrices $\boldsymbol{P}$ compliant with the graph neighborhood structure. For the formal determination of the maximum, we compare two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ using

$$\boldsymbol{a} < \boldsymbol{b} \Leftrightarrow \forall i : \boldsymbol{a}_i \leq \boldsymbol{b}_i \wedge \exists i : \boldsymbol{a}_i < \boldsymbol{b}_i. \tag{2.67}$$

This corresponds to the component-wise determination of the maximum in Equation 2.64.



(a) Case $\boldsymbol{P}'_i\boldsymbol{x} \geq \boldsymbol{P}''_i\boldsymbol{y}$. **(b)** Case $\boldsymbol{P}'_i\boldsymbol{x} < \boldsymbol{P}''_i\boldsymbol{y}$.

Figure 2.5 The two cases of Equation 2.69. In case (a), replacing $\boldsymbol{P}''_i\boldsymbol{y}$ by $\boldsymbol{P}'_i\boldsymbol{y}$ can only result in a lower value, since $\boldsymbol{P}''_i$ maximizes $\boldsymbol{P}''_i\boldsymbol{y}$. Consequently, $|\boldsymbol{P}'_i\boldsymbol{x} - \boldsymbol{P}''_i\boldsymbol{y}| \leq |\boldsymbol{P}'_i\boldsymbol{x} - \boldsymbol{P}'_i\boldsymbol{y}|$. In case (b), the roles of $\boldsymbol{P}'_i\boldsymbol{x}$ and $\boldsymbol{P}''_i\boldsymbol{y}$ are exchanged. In both cases, $|\boldsymbol{P}'_i\boldsymbol{x} - \boldsymbol{P}''_i\boldsymbol{y}| \leq |\boldsymbol{P}_i\boldsymbol{x} - \boldsymbol{P}_i\boldsymbol{y}|$.

Convergence

We can now state the main theorem of this chapter.

Theorem. *For any $\boldsymbol{x}^{(0)} \geq 0$, the iteration given by Equation 2.64 converges to the unique solution of $\boldsymbol{x} = (1-\alpha)\boldsymbol{k}_{\boldsymbol{v}} + \alpha \max_{\boldsymbol{P}} \boldsymbol{P}\boldsymbol{x}$.*

Proof. Let $\boldsymbol{M} = \left\{\boldsymbol{x} \in \mathbb{R}^{|V||V'|} \;\middle|\; \boldsymbol{x}_i \geq 0\right\}$ denote the non-negative orthant, and let $f : \boldsymbol{M} \to \boldsymbol{M}$, $\boldsymbol{x} \mapsto (1-\alpha)\boldsymbol{k}_{\boldsymbol{v}} + \alpha \max_{\boldsymbol{P}} \boldsymbol{P}\boldsymbol{x}$. We show that f is a contraction mapping on $\boldsymbol{M}$, that is $\|f(\boldsymbol{x}) - f(\boldsymbol{y})\| \leq \lambda \|\boldsymbol{x} - \boldsymbol{y}\|$ for some positive $\lambda < 1$ and some norm $\|\cdot\|$. Let

$$\boldsymbol{P}' = \arg\max_{\boldsymbol{P}} \boldsymbol{P}\boldsymbol{x}, \qquad \boldsymbol{P}'' = \arg\max_{\boldsymbol{P}} \boldsymbol{P}\boldsymbol{y}, \tag{2.68}$$

i. e., $\boldsymbol{P}'$ and $\boldsymbol{P}''$ are the matrices that maximize $\boldsymbol{P}\boldsymbol{x}$ and $\boldsymbol{P}\boldsymbol{y}$ componentwise. Define $\boldsymbol{P}$ by setting

$$\boldsymbol{P}_i = \begin{cases} \boldsymbol{P}'_i & \text{if } \boldsymbol{P}'_i\boldsymbol{x} \geq \boldsymbol{P}''_i\boldsymbol{y} \\ \boldsymbol{P}''_i & \text{if } \boldsymbol{P}'_i\boldsymbol{x} < \boldsymbol{P}''_i\boldsymbol{y} \end{cases}, \tag{2.69}$$

where $\boldsymbol{P}_i$ denotes the i-th row of $\boldsymbol{P}$. Note that $|\boldsymbol{P}'_i\boldsymbol{x} - \boldsymbol{P}''_i\boldsymbol{y}| \leq |\boldsymbol{P}_i\boldsymbol{x} - \boldsymbol{P}_i\boldsymbol{y}|$ (Figure 2.5), giving

$$\begin{aligned} \left\|f(\boldsymbol{x}) - f(\boldsymbol{y})\right\|_\infty &= \alpha \left\|\max_{\boldsymbol{P}} \boldsymbol{P}\boldsymbol{x} - \max_{\boldsymbol{P}} \boldsymbol{P}\boldsymbol{y}\right\|_\infty \\ &= \alpha \left\|\boldsymbol{P}'\boldsymbol{x} - \boldsymbol{P}''\boldsymbol{y}\right\|_\infty \\ &\leq \alpha \left\|\boldsymbol{P}(\boldsymbol{x} - \boldsymbol{y})\right\|_\infty \\ &\leq \alpha \left\|\boldsymbol{x} - \boldsymbol{y}\right\|_\infty. \end{aligned} \tag{2.70}$$

The last line follows from a property of $\boldsymbol{P}$: At most $\min\left\{|v|, |v'|\right\}$ (the number of assigned neighbors) entries in the i-th row of $\boldsymbol{P}$ are not zero, and every such entry contains the factor $1/\max\left\{|v|, |v'|\right\}$, so the i-th component of $\boldsymbol{P}\boldsymbol{x}$ can be at most $\max_i |\boldsymbol{x}_i|$.

Since f is a contraction mapping defined on the complete metric space $\boldsymbol{M}$, the proposition follows from Banach's fixed point theorem (Granas and Dugundji, 2003). □

Number of iterations

The following lemma states a number of iterations k sufficient for the computation of $\boldsymbol{x}^{(k)}$ to a desired precision ϵ. Due to several inequalities used, the number of necessary iterations will, in general, be lower than k.

Lemma. *For given $\epsilon > 0$,*

$$\left\|\boldsymbol{x}^{(k)} - \lim_{n\to\infty} \boldsymbol{x}^{(n)}\right\|_\infty \leq \epsilon \text{ after at most } k = \left\lceil \log_\alpha \frac{(1-\alpha)\epsilon}{\left\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^{(1)}\right\|_\infty} \right\rceil \text{ iterations.}$$

Proof. We want to find $k \geq 0$ such that $\left\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k+m)}\right\| < \epsilon$ for all $m \geq 0$. By repeated application of the triangle inequality we get

$$\left\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k+m)}\right\|_\infty \leq \sum_{l=0}^{m-1} \left\|\boldsymbol{x}^{(k+l)} - \boldsymbol{x}^{(k+l+1)}\right\|_\infty. \tag{2.71}$$

Using the proof from the previous page gives

$$\begin{aligned} \left\|\boldsymbol{x}^{(p)} - \boldsymbol{x}^{(p+1)}\right\|_\infty &\leq \alpha \left\|\boldsymbol{x}^{(p-1)} - \boldsymbol{x}^{(p)}\right\|_\infty \\ &\leq \alpha^2 \left\|\boldsymbol{x}^{(p-2)} - \boldsymbol{x}^{(p-1)}\right\|_\infty \\ &\leq \dots \\ &\leq \alpha^p \left\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^{(1)}\right\|_\infty. \end{aligned} \tag{2.72}$$

Combining Equations 2.71 and 2.72, applying the geometric series yields

$$\begin{aligned} \left\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k+m)}\right\|_\infty &\leq \sum_{l=0}^{m-1} \alpha^{k+l} \left\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^{(1)}\right\|_\infty \\ &\leq \alpha^k \left\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^{(1)}\right\|_\infty \sum_{l\geq 0} \alpha^l \\ &= \frac{\alpha^k}{1-\alpha} \left\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^{(1)}\right\|_\infty. \end{aligned} \tag{2.73}$$

If $\boldsymbol{x}^{(0)} = \boldsymbol{x}^{(1)}$, we are done. Otherwise, solving for k gives

$$\frac{\alpha^k}{1-\alpha}\left\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^{(1)}\right\|_\infty \leq \epsilon \iff k \geq \log_\alpha \frac{(1-\alpha)\epsilon}{\left\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^{(1)}\right\|_\infty}. \tag{2.74}$$

Since k is integer, the proposition follows. □

2.3.3 A kernel for molecular graphs

Combining the ideas of optimal assignment kernels (Subsection 2.2.5) and iterative similarity for molecular graphs (Equation 2.64) leads to the *iterative similarity optimal assignment kernel* (ISOAK),

$$k(G, G') = \max_\pi \sum_{i=1}^{|V|} k_x(v_i, v'_{\pi(i)}), \qquad k_x(v_i, v'_j) = \left(\lim_{n\to\infty} \boldsymbol{x}^{(n)}\right)_{(j-1)|V|+i}. \tag{2.75}$$

Note that we assume $|V| \leq |V'|$ (as for Equations 2.34 and 2.35).

Computation

The vertex similarity scores k_x can be computed iteratively to a given precision ϵ using Equation 2.66 and Lemma 2.3.2. In each iteration and for each pair of vertices $(v, v') \in V \times V'$, an optimal assignment π of the neighbors of v to the neighbors of v' (or vice versa) is computed, based on the similarity values of the previous iteration (Algorithm 2.1). The final optimal vertex assignment can be done using the Kuhn-Munkres assignment algorithm (Kuhn, 1955; Munkres, 1957); Bourgeois and Lassalle (1971) give an extension of the algorithm to non-quadratic matrices (Algorithm 2.2).

Figures 2.10 and 2.11 show examples of ISOAK similarity and assignments.

Algorithm 2.1 Iterative similarity optimal assignment kernel. A Java (version 1.5, Sun microsystems, `www.sun.com`) implementation is available at `www.mrupp.info`.

(a) Normalization uses (b) to compute $k(G, G')/\sqrt{k(G, G)k(G', G')}$ (Equation 2.36). If $k_v(v, v) = k_e(e, e) = 1$ for all $v \in V$, $e \in E$, then $k(G, G) = |V|$ by Equation 2.64.

Input: graphs $G=(V, E), G'=(V', E')$, vertex and edge kernels k_v, k_e, parameters α, ϵ.
Output: ISOAK $k(G, G')$ normalized to $[0, 1]$.

1 Use (b) to compute $k(G, G')$, $k(G, G)$, and $k(G', G')$.
2 Return $k(G, G')/\sqrt{k(G, G)k(G', G')}$.

(b) ISOAK computation. For implementation purposes, one can use column-wise linearized indices, where $\boldsymbol{X}_{i,j}$ corresponds to entry $(j-1)|V|+i$ in $\mathrm{vec}(\boldsymbol{X})$. The neighborhood assignments (prefixes of length $|v_i|$ of permutations of $\{1, \ldots, |v'_j|\}$, or vice versa; see Sedgewick (1977) for a survey and analysis of permutation generation algorithms), as well as k_v and k_e can be precomputed.

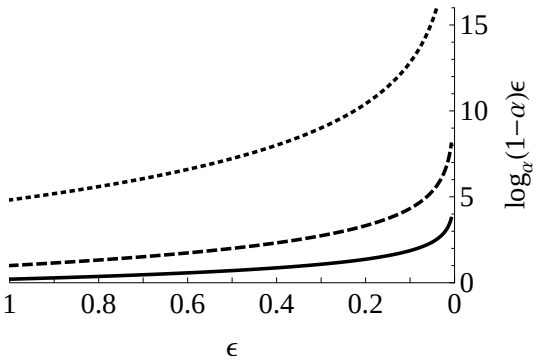
Input: graphs $G=(V, E), G'=(V', E')$, vertex and edge kernels k_v, k_e, parameters α, ϵ.
Output: ISOAK $k(G, G')$, similarity matrix $\boldsymbol{X}^{(k)}$, optimal assignment $(j_1, \ldots, j_n)$.

1 Set $\boldsymbol{X}^{(0)} \leftarrow \mathbf{1}_{|V|\times|V'|}$, $k \leftarrow 0$.
2 Do
3 Set $k \leftarrow k + 1$.
4 For each $(i, j) \in |V| \times |V'|$ do
5 Set $\boldsymbol{X}^{(k)}_{i,j} \leftarrow 0$.
6 If $|v_i| < |v'_j|$, for each neighborhood assignment π,
  set $\boldsymbol{X}^{(k)}_{i,j} \leftarrow \max\{\boldsymbol{X}^{(k)}_{i,j}, |v'_j|^{-1} \sum_{\{v_i,u\}\in E} \boldsymbol{X}^{(k-1)}_{u,\pi(u)} k_e(\{v_i, u\}, \{v'_j, \pi(u)\})\}$.
7 If $|v_i| \geq |v'_j|$, for each neighborhood assignment π,
  set $\boldsymbol{X}^{(k)}_{i,j} \leftarrow \max\{\boldsymbol{X}^{(k)}_{i,j}, |v_i|^{-1} \sum_{\{v'_j,u'\}\in E'} \boldsymbol{X}^{(k-1)}_{\pi(u'),u'} k_e(\{v_i, \pi(u')\}, \{v'_j, u'\})\}$.
8 Set $\boldsymbol{X}^{(k)}_{i,j} \leftarrow (1-\alpha)k_v(v_i, v'_j) + \alpha \boldsymbol{X}^{(k)}_{i,j}$.
9 until $k \geq \lceil \log_\alpha(1-\alpha)\epsilon / \left\|\mathrm{vec}(\boldsymbol{X}^{(0)}) - \mathrm{vec}(\boldsymbol{X}^{(1)})\right\|_\infty \rceil$.
10 Use Algorithm 2.2 to compute an optimal assignment of $-\boldsymbol{X}^{(k)}$.

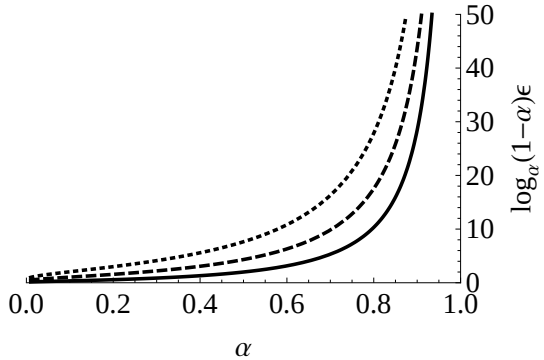
Algorithm 2.2 Optimal assignment algorithm (also Hungarian algorithm; Kuhn, 1955, 1956; Munkres, 1957), in an extended version for rectangular matrices (Bourgeois and Lassalle, 1971). A Java (version 1.5, Sun microsystems, `www.sun.com`) implementation is available at `www.mrupp.info`. The algorithm distinguishes zero entries of the matrix as primed $0'$ and starred $0^\star$; rows can be covered.

Input: matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n'}$, $n \leq n'$.
Output: assignment $(j_1, \ldots, j_n)$ minimizing $\sum_{i=1}^{n} \boldsymbol{A}_{i,j_i}$.

Initially, no lines are covered, no zeros are starred or primed.
1 From each row, subtract its minimum.
2 For all $\boldsymbol{A}_{i,j}$: If $\boldsymbol{A}_{i,j} = 0$ and there is no $0^\star$ in row i or column j, then set $\boldsymbol{A}_{i,j} \leftarrow 0^\star$.
3 Cover each column with a $0^\star$ in it.
If n columns are covered, return their indices $(j_1, \ldots, j_n)$ as the optimal assignment.
4 If all zeros are covered, go to 6, else prime a non-covered $\boldsymbol{A}_{i,j} = 0$.
If there is no $\boldsymbol{A}_{i,k} = 0^\star$ in row i, go to step 5,
else cover row i, un-cover column k, and go to step 4.
5 Construct a series of alternating $0'$ and $0^\star$:
Set $\boldsymbol{z}_0 \leftarrow (i, j)$. *$\boldsymbol{z}_0$ is the $0'$ found in step 4.*
Set $\boldsymbol{z}_1$ to the $0^\star$ in $\boldsymbol{z}_0$'s column (if any), set $\boldsymbol{z}_2$ to the $0'$ in $\boldsymbol{z}_1$'s row, and so on.
Un-star each $0^\star$ and star each $0'$ in the sequence.
Erase all $0'$ in $\boldsymbol{A}$, un-cover every row, and go to step 3.
6 Set h to the smallest non-covered element in $\boldsymbol{A}$. *h was un-covered in step 4.*
Add h to all covered rows, then subtract h from all non-covered columns.
Go to step 4.



(a) Dependence on ϵ for $\alpha = \frac{1}{4}$ (solid line), $\alpha = \frac{1}{2}$ (dashed line), and $\alpha = \frac{3}{4}$ (dotted line).



(b) Dependence on α for $\epsilon = \frac{1}{2}$ (solid line), $\epsilon = \frac{1}{10}$ (dashed line), and $\epsilon = \frac{1}{100}$ (dotted line).

Figure 2.6 ISOAK runtime dependence on parameters ϵ and α.

Runtime

The number of assignments π that need to be considered for each vertex pair is

$$|v'|\big(|v'|-1\big)\cdots\big(|v'|-|v|+1\big)=\frac{|v'|!}{\big(|v'|-|v|\big)!}. \tag{2.76}$$

This leads to a factorial worst-case runtime complexity for general graphs. In molecular graphs, the vertex degree is bounded by a small constant (p. 63), up to which the corresponding assignments can be precomputed, allowing the determination of each vertex assignment π in constant time. In this way, update Equation 2.66 exploits the bounded degree of molecular graphs.

We assume constant runtime complexity for k_v and k_e; therefore, each iteration takes time in $O\big(|V|\,|V'|\big)$. By Lemma 2.3.2, at most $k=\left\lceil\log_\alpha\big((1-\alpha)\epsilon/\|\boldsymbol{x}^{(0)}-\boldsymbol{x}^{(1)}\|_\infty\big)\right\rceil$ iterations are necessary. From $\|\boldsymbol{x}^{(0)}-\boldsymbol{x}^{(1)}\|_\infty\leq 1$ and $\alpha\in(0,1)$, it follows that $\log_\alpha(\|\boldsymbol{x}^{(0)}-\boldsymbol{x}^{(1)}\|_\infty)\geq 0$, and the computation of the similarity matrix $\boldsymbol{X}$ takes time in

$$O\big(|V|\,|V'|\,\log_\alpha\big((1-\alpha)\epsilon\big)\big). \tag{2.77}$$

Computing the final optimal assignment (Algorithm 2.2) has cubic runtime, resulting in a total runtime complexity of

$$O\big(\max\{|V|,|V'|\}^3+|V|\,|V'|\,\log_\alpha\big((1-\alpha)\epsilon\big)\big). \tag{2.78}$$

With the convention $|V|<|V'|$, this simplifies to

$$O\big(|V'|^3+|V'|^2\log_\alpha\big((1-\alpha)\epsilon\big)\big). \tag{2.79}$$

Runtime depends polynomially on the size of the input graphs, and increases exponentially for $\epsilon\to 0$ (Figure 2.6a) and $\alpha\to 1$ (Figure 2.6b) as the approximation approaches the exact solution and the complete topology of the graphs is considered. Empirical runtimes agree with Equation 2.79 ($r^2=1$, RMSE $=0.016$, Figure 2.7).

Positive definiteness

Optimal assignments are not positive definite in general (p. 71), and whether Equation 2.75 constitutes a proper kernel or not depends on the measure of pairwise vertex similarity k_x. We do not know if the ISOAK update rule (Equation 2.64) always leads to positive definite optimal assignments.

We have checked all kernel matrices computed with Algorithm 2.1 in several experiments for positive definiteness with the eigenvalue criterion (p. 30). Slightly negative eigenvalues were encountered, but might be partly or entirely numerical artifacts. Figure 2.8 shows the distribution of the smallest eigenvalues over the parameter α. Most of the negative smallest eigenvalues are close to zero; for $\alpha\to 1$ all eigenvalues are non-negative, indicating that the recursive similarity part of Equation 2.75 might be positive definite in general.

In practice, matrices with negative smallest eigenvalue λ_n can be corrected by adding a constant σ to their diagonal. This shifts their eigenspectrum,

$$(\boldsymbol{M}+\sigma\boldsymbol{I})\boldsymbol{x}=\boldsymbol{M}\boldsymbol{x}+\sigma\boldsymbol{x}=(\lambda+\sigma)\boldsymbol{x}, \tag{2.80}$$

and adding $\sigma=|\lambda_n|$ to their diagonal renders them positive definite. A disadvantage of this correction is that adding to the diagonal can worsen the performance of kernel algorithms due to diagonal dominance (Greene and Cunningham, 2006).

Expressiveness

According to Equation 2.77, the runtime of Algorithm 2.1 increases with α and $1/\epsilon$. Such an increase in computation time should lead to an improvement in discriminative power. To quantify this, we introduce the *expressiveness* $\chi(k, D)$ of a normalized graph kernel k on a data set D as the fraction of separated graph pairs,

$$\chi(k, D) = \frac{\left|\left\{\{G, G'\} \,\middle|\, G, G' \in D, k(G, G') \neq 1\right\}\right|}{\frac{1}{2}|D|(|D| - 1)} \in [0, 1]. \tag{2.81}$$

On all the data sets investigated in Section 2.4 and for all the used vertex and edge kernels, expressivity increased monotonically with α (Figure 2.9). Since runtime also increases with α, this parameter directly controls the trade-off between separation power and computational complexity. For the used data sets, note that expressivity saturates already for small values of $\alpha \approx 0.2$.

2.4 Retrospective evaluation

We retrospectively evaluate (Section 1.4) the performance of ISOAK in multiple virtual screening experiments using support vector machines (Shawe-Taylor and Cristianini, 2004) for binary classification and regression on public, proprietary, and benchmark data sets. We analyze the results and compare them to those of optimal assignment kernels (Subsection 2.2.5) from the literature. For a prospective application, see Chapter 4.

2.4.1 Data sets

We tested our graph kernel on 12 different data sets (Table 2.3). These included public (`drug-nondrug`, `ptcfm`, `ptcmm`, `ptcfr`, `ptcmr`) as well as proprietary data (`cobra:ache`, `cobra:cox2`, `cobra:dhfr`, `cobra:fxa`, `cobra:ppar`, `cobra:thrombin`, `bbb`) coming from four different sources. 11 data sets are binary classification problems and one (`bbb`) is a regression problem. In all data sets, duplicate molecules and molecules that could not be processed by some of the used software were removed.

Drugs versus non-drugs

Drug-likeness, i. e., the similarity of a compound to known drugs in terms of physicochemical properties like solubility and lipophilicity, is an important concept in drug discovery (Leeson and Springthorpe, 2007). It has been characterized in various ways; a prominent example is the rule of five (Lipinski et al., 1997), which states that for small molecules, notwithstanding substrates of biological transporters, "poor absorption or permeation are more likely when there are more than 5 hydrogen-bond donors, more than 10 hydrogen-bond acceptors, the molecular weight is over 500, or, the [computed] Log P (MLog P) is over 5 (4.15)".

We compiled the `drug-nondrug` data set using the DrugBank repository (Wishart et al., 2008, `www.drugbank.ca`, $n_+ = 809$ drugs) and randomly sampled compounds from the Sigma-Aldrich catalog (`www.sigmaaldrich.com`; $n_- = 734$ assumed non-drugs).

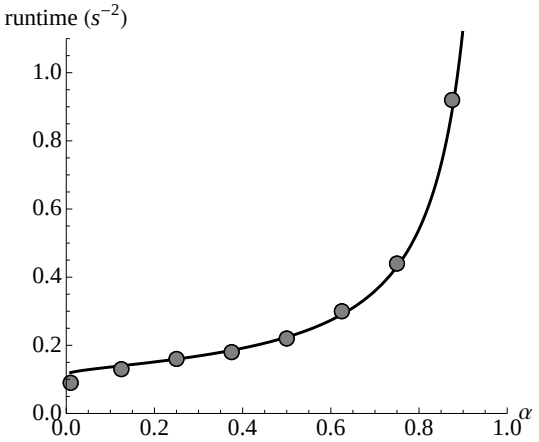


Figure 2.7 Empirical ISOAK runtimes (gray disks) and a fit to Equation 2.79 (solid line). Runtimes are in units of $10^{-2}\,\mathrm{s}$ for a Java (version 1.5, Sun microsystems, `www.sun.com`) implementation running on a Xeon processor (2.2 GHz, 3 GB RAM, Intel, `www.intel.com`). Runtime values for each α are averages over several data sets and choices of k_v and k_e, with $\epsilon = \frac{1}{100}$. The average runtime over all computed kernel values was $1.88 \pm 0.17\,\mathrm{s}^{-2}$. The fit $0.1042 + 0.0157 \log_\alpha(1-\alpha)\frac{1}{100}$ has $r^2 = 1$ and RMSE $= 0.016$.

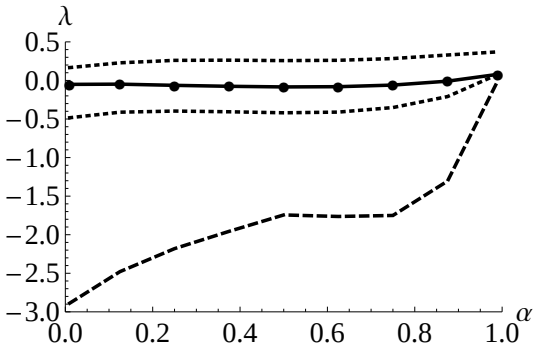


Figure 2.8 Smallest eigenvalues of ISOAK matrices, plotted against parameter α. Shown are the mean (solid line), mean $\pm$ standard deviation of negative and positive values, respectively (dotted lines), and, the minimum (dashed line), computed at a precision of $\epsilon = 10^{-3}$ over all parametrizations and all data sets used in Section 2.4 ($n = 8 \cdot 12 = 96$); $\alpha \in \{\frac{1}{100}, \frac{1}{8}, \frac{2}{8}, \frac{3}{8}, \frac{4}{8}, \frac{5}{8}, \frac{6}{8}, \frac{7}{8}, \frac{99}{100}\}$.

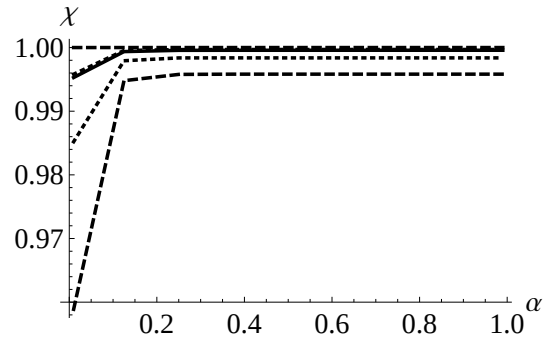
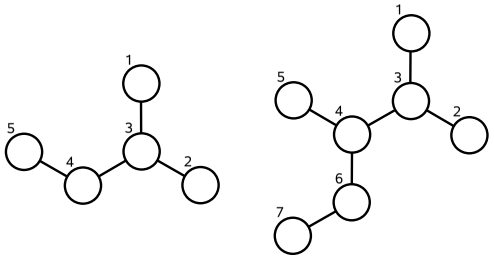


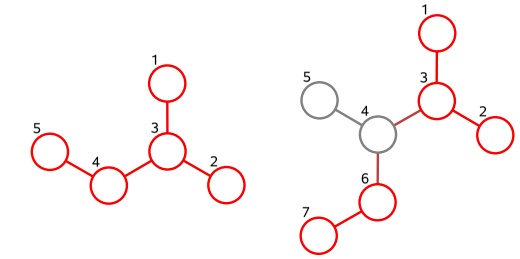
Figure 2.9 Expressiveness of ISOAK, plotted against parameter α. Shown are the mean (solid line), mean $\pm$ standard deviation of negative and positive values, respectively (dotted lines), and, the minimum and maximum values (dashed lines), computed to a precision of $\epsilon = 10^{-3}$ over all data sets and parametrizations used in Section 2.4 ($n = 8 \cdot 12 = 96$); $\alpha \in \{\frac{1}{100}, \frac{1}{8}, \frac{2}{8}, \frac{3}{8}, \frac{4}{8}, \frac{5}{8}, \frac{6}{8}, \frac{7}{8}, \frac{99}{100}\}$.



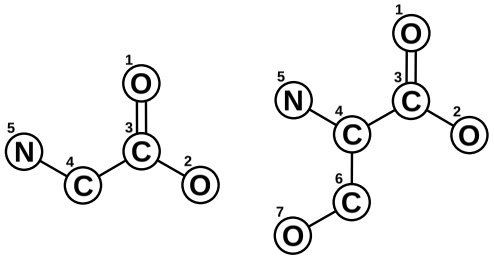
(a) Topological graphs. Only the graph is used.

	1	2	3	4	5	6	7
1	0.98	0.98	0.66	0.66	0.97	0.73	0.89
2	0.98	0.98	0.66	0.66	0.97	0.73	0.89
3	0.63	0.63	0.96	0.93	0.64	0.78	0.66
4	0.74	0.74	0.81	0.81	0.73	0.98	0.70
5	0.90	0.90	0.64	0.66	0.91	0.70	0.99

(b) Pairwise vertex similarities. Boxes indicate vertex assignments.



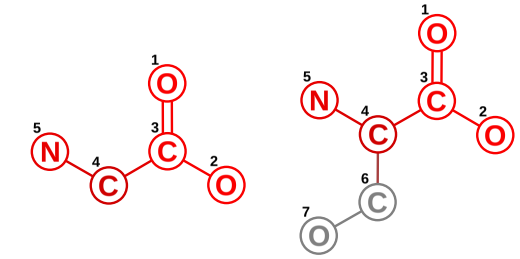
(c) Vertex assignments, color-coded from red (highest similarity) to black (lowest similarity); gray vertices were not assigned.



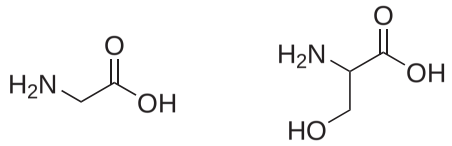
(d) Annotated graphs. Vertices are annotated with element type (C,O,N), bonds with bond type (single, double).

	1	2	3	4	5	6	7
1	0.98	0.50	0.00	0.00	0.00	0.00	0.50
2	0.50	0.98	0.11	0.16	0.34	0.17	0.89
3	0.00	0.11	0.96	0.68	0.14	0.78	0.13
4	0.00	0.24	0.67	0.81	0.17	0.77	0.20
5	0.00	0.33	0.14	0.13	0.91	0.20	0.38

(e) Pairwise vertex similarities. Boxes indicate vertex assignments.

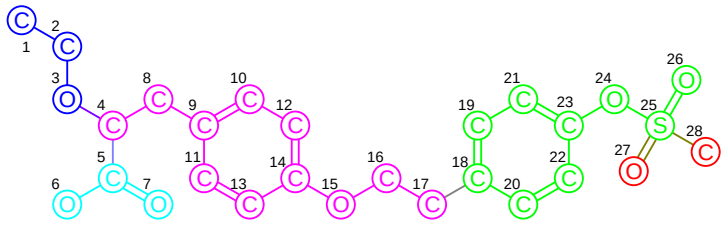


(f) Vertex assignments, color-coded from red (highest similarity) to black (lowest similarity); gray vertices were not assigned.

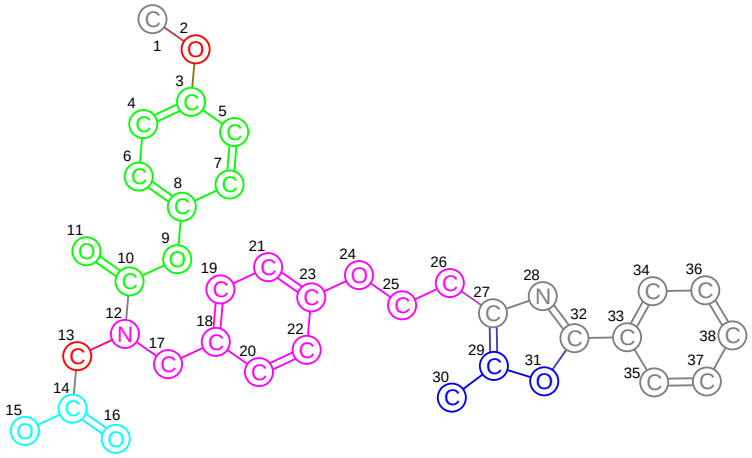


(g) Chemical structure graphs of the two amino acids glycine (left) and serine (right).

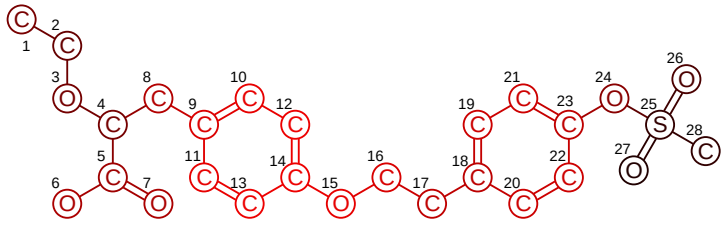
Figure 2.10 ISOAK on glycine and serine (g). Two computations are shown, based on topological graphs (a)–(c) and on graphs annotated with element/bond types (d)–(f). In both cases, $\alpha = 0.5$ and $\epsilon = 0.01$. Vertex assignments are shown boxed (b,e) and color-coded (c,f) from red (highest similarity) to black (lowest similarity); gray vertices were not assigned.



(a) Tesaglitazar annotated graph.



(b) Muraglitazar annotated graph.



(c) Assignment strength, color-coded from red (highest similarity) to black (lowest similarity).

Figure 2.11 ISOAK on tesaglitazar (a, Compound 38, p. 152) and muraglitazar (b, Compound 37, p. 152), with Dirac kernels on vertex element type and edge bond type annotation, $\alpha = 0.875$, $\epsilon = 0.01$. Colors indicate assigned components, with gray indicating unassigned vertices. Assignment strength is shown separately (c). The overall normalized similarity is 0.57.

COBRA subsets

The COBRA database (version 6.1; Schneider and Schneider, 2003) is a commercial data set containing compounds collected from the literature, annotated with activity on biological targets and interaction mode. Subsets `cobra:ache` (n_-=58, n_+=92), `cobra:cox2` (n_-=136, n_+=126), `cobra:dhfr` (n_-=60, n_+=60), `cobra:fxa` (n_-=228, n_+=221), `cobra:ppar` (n_-=94, n_+=92), and `cobra:thrombin`(n_-=185, n_+=186) were used.

For each data set, all compounds belonging to the respective class were taken as positive samples, and an identical number of molecules was randomly selected from the database as negative samples. In this, the negative samples problem (p. 28) was accepted as a concession to conventional procedure and the ability to directly compare results with those in the literature.

Predictive toxicology challenge

The predictive toxicology challenge 2000–2001 (Helma and Kramer, 2003) dealt with prediction of carcinogenicity in rodents, based on data from the national toxicology program (NTP) of the United States department of health and human services. We used the training data sets `ptcfm` (female mice, $n_- = 202$, $n_+ = 135$), `ptcmm` (male mice, $n_- = 204$, $n_+ = 118$), `ptcfr` (female rats, $n_- = 224$, $n_+ = 118$), and `ptcmr` (male rats, $n_- = 186$, $n_+ = 146$); all are binary classification problems.

Blood-brain barrier permeability

The *blood-brain barrier* (BBB; Edwards, 2001; Cecchelli et al., 2007) is one of several mechanisms regulating the exchange of substances between blood and the central nervous system (CNS). The BBB is formed by the endothelial cells of cerebral capillaries, regulating access to the CNS to protect it against changes in the hematic environment. Predicting a compounds ability to permeate the BBB is important in drug development, as drugs targeted at the CNS have to cross the BBB, whereas peripherally acting drugs should not do so in order to prevent CNS-mediated side effects. We use the `bbb` (n=115) data set published by Hou and Xu (2003), a regression problem.

2.4.2 Representation

All data sets were treated identically. Molecular graphs did not include hydrogen atoms.

ISOAK parametrizations

In total, 8 different combinations of vertex and edge annotations, and thus of vertex and edge kernels, were used (Table 2.4). For discrete labels, we used the Dirac kernel (Equation 2.38). For continuous labels, we used the Gaussian kernel (p. 30) with the kernel width σ set to the standard deviation of the labels in a data set. Pharmacophore types were computed using the molecular query language (MQL; Proschak et al., 2007, Table 2.5). Gasteiger-Marsili partial charges (Gasteiger and Marsili, 1980) were computed using the PETRA software (version 3.11, Molecular networks, `www.molecular-networks.com`).

Baseline representations

Besides the different ISOAK parametrizations, we used two established vectorial descriptors as baseline representations, the CATS2D descriptor (p. 154) and Ghose-Crippen fragment descriptors (p. 156). With vectorial descriptors, we used two standard kernels, the homogeneous polynomial kernel (p. 30) and the Gaussian kernel (p. 30).

Table 2.3 ISOAK retrospective evaluation data sets.

	Samples		
Name	neg.	pos.	Description
`drug-nondrug`	734	809	Known and desirable bioactivity
`cobra:ache`	58	92	Acetylcholinesterase inhibitors
`cobra:cox2`	136	126	Cyclooxygenase-2 inhibitors
`cobra:dhfr`	60	60	Dihydrofolate reductase inhibitors
`cobra:fxa`	228	221	Factor Xa inhibitors
`cobra:ppar`	94	92	Peroxisome proliferator-activated receptor agonists
`cobra:thrombin`	185	186	Thrombin inhibitors
`ptcfm`	202	135	Predictive toxicology challenge, female mice subset
`ptcmm`	204	118	Predictive toxicology challenge, male mice subset
`ptcfr`	224	118	Predictive toxicology challenge, female rats subset
`ptcmr`	186	146	Predictive toxicology challenge, male rats subset
`bbb`	115		Blood-brain barrier

Table 2.4 ISOAK parametrizations. All possible eight combinations of the listed vertex and edge kernels were used.

Abbrev.	Description
Vertex kernel	
none	$k_v(v, v') = 1$. No vertex kernel.
delement	Dirac kernel (Equation 2.38) with element types as labels.
dppp	Dirac kernel with potential pharmacophore points (Table 2.5) as labels.
echarge	Gaussian kernel (p. 30) with Gasteiger-Marsili partial charges as labels.
Edge kernel	
none	$k_e(e, e') = 1$. No edge kernel.
dbond	Dirac kernel using covalent bond type (single, double, triple) as label.

Table 2.5 Molecular query language (MQL; Proschak et al., 2007) definitions of the used potential pharmacophore points (PPP).

PPP	MQL definition
lipophilic	C[!bound(~Hetero)], Cl, Br, I
positive	*[charge>0], N[allHydrogens>1]
negative	*[charge<0], O=P'~O', O=S'~O', O=C'~O'[allHydrogens=1|charge<0], O[allHydrogens=1|charge<0]~C'=O
acceptor	O, N[allHydrogens=0]
donor	O[allHydrogens=1&!bound(-C=O)], N[allHydrogens>0]

2.4.3 Support vector machines

Many good introductions to support vector machines (SVMs) exist, and we limit our exposition to the basic ideas. Bennett and Campbell (2000) provide an intuitive, geometric introduction to SVMs; for details, see the text books by Cristianini and Shawe-Taylor (2000); Steinwart and Christmann (2008). For reviews of SVMs in chemistry and computational biology, see Ivanciuc (2007) and Ben-Hur et al. (2008), respectively.

Separating hyperplanes

Consider a binary classification problem with training samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$ and associated labels $y_1, \ldots, y_n \in \{-1, 1\}$. The decision function

$$f(\boldsymbol{x}) = \operatorname{sgn} g(\boldsymbol{x}), \qquad g(\boldsymbol{x}) = \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b, \tag{2.82}$$

depends on a vector $\boldsymbol{w} \in \mathbb{R}^d$ and a bias $b \in \mathbb{R}$ that determine orientation and offset of a discriminant (hyper)plane (Figure 2.12a). The hyperplane is orthogonal to $\boldsymbol{w}$ and the bias b translates it along $\boldsymbol{w}$. New samples $\boldsymbol{x}'$ will be classified as negative or positive depending on which side of the hyperplane they lie, i. e., whether $g(\boldsymbol{x}') < 0$ or $g(\boldsymbol{x}') > 0$.

Maximum margin

A finite number of linearly separable training samples can be separated by infinitely many hyperplanes, i. e., $\boldsymbol{w}$ and b such that $f(\boldsymbol{x}_i) = y_i$. Intuitively, the one that generalizes best to new samples[7] is the one farthest from all training samples, i. e., the one with maximum *margin* $d_- + d_+$, where d_- and d_+ are the shortest distances between the hyperplane and the negative and positive samples, respectively. Since the decision function f is invariant under positive rescalings of g, we can require

$$\begin{aligned} g(\boldsymbol{x}_i) = \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b \geq +1 \quad \text{if } y_i = 1 \qquad (2.83a) \\ g(\boldsymbol{x}_i) = \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b \leq -1 \quad \text{if } y_i = -1 \qquad (2.83b) \end{aligned} \quad ,$$

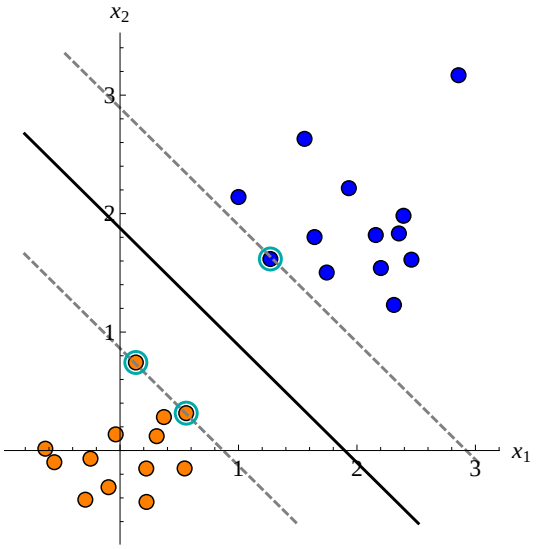
with equality for at least one training sample $\boldsymbol{x}_+$ and $\boldsymbol{x}_-$, respectively. Equation 2.83 defines two hyperplanes, parallel to the separating hyperplane, with no training samples between them. The distance between the projections (Meyer, 2001) of $\boldsymbol{x}_+$ and $\boldsymbol{x}_-$ onto the weight vector $\boldsymbol{w}$ equals the margin,

$$d_- + d_+ = \left\| \frac{\langle \boldsymbol{w}, \boldsymbol{x}_+ \rangle}{\langle \boldsymbol{w}, \boldsymbol{w} \rangle} \boldsymbol{w} - \frac{\langle \boldsymbol{w}, \boldsymbol{x}_- \rangle}{\langle \boldsymbol{w}, \boldsymbol{w} \rangle} \boldsymbol{w} \right\| = \left\| \frac{\langle \boldsymbol{w}, \boldsymbol{x}_+ - \boldsymbol{x}_- \rangle}{\langle \boldsymbol{w}, \boldsymbol{w} \rangle} \boldsymbol{w} \right\| = \frac{2}{\langle \boldsymbol{w}, \boldsymbol{w} \rangle} \|\boldsymbol{w}\| = \frac{2}{\|\boldsymbol{w}\|}, \tag{2.84}$$

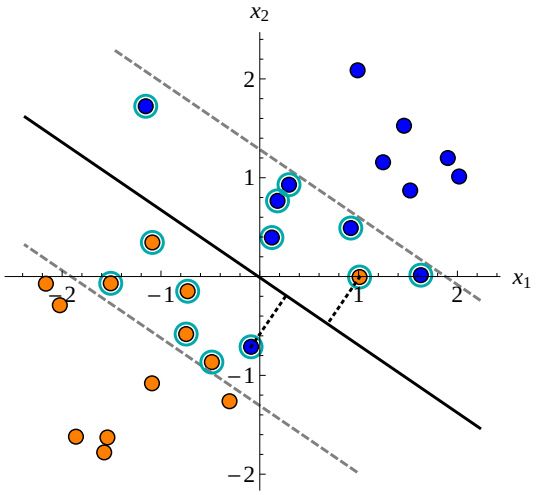
where $\langle \boldsymbol{w}, \boldsymbol{x}_+ - \boldsymbol{x}_- \rangle = 2$ follows from subtracting Equation 2.83b from Equation 2.83a. Maximizing the margin is equivalent to minimizing its inverse $\frac{1}{2} \|\boldsymbol{w}\|^2$, resulting in the *primal hard-margin* formulation of SVMs:

$$\min_{\boldsymbol{w} \in \mathbb{R}^d, b \in \mathbb{R}} \frac{1}{2} \|\boldsymbol{w}\|^2 \quad \text{such that} \quad y_i \big(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b \big) \geq 1 \text{ for } i \in \{1, \ldots, n\}. \tag{2.85}$$

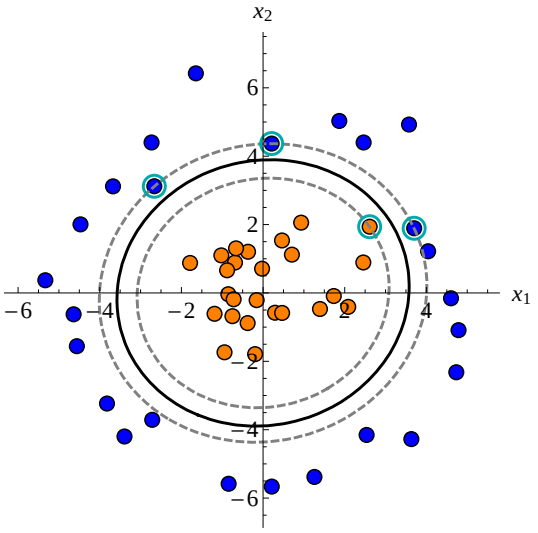
[7] Drawn from the same distribution as the training samples. Relaxation of this assumption leads to the problem of *covariate shift* (Shimodaira, 2000; Zadrozny, 2004; Sugiyama et al., 2007a,b). The latter is relevant to ligand-based virtual screening because training and (prospective) test samples are often from different data sets, i. e., from different distributions (Subsection 1.4.1).



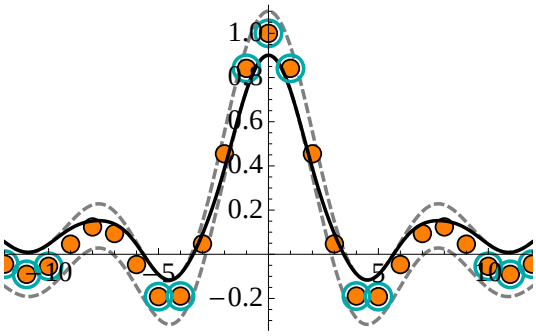
(a) Linear separable case. Two times 13 samples drawn i. i. d. from two Gaussian distributions $\mathcal{N}_1\big((0,0),0.5\big)$ and $\mathcal{N}_2\big((2,2),0.5\big)$; 3 support vectors; $w = (0.97, 0.98)$, $b = -1.85$.



(b) Linear inseparable case. Two times 13 samples drawn i. i. d. from two Gaussian distributions $\mathcal{N}_1\big(-(1,1),1\big)$ and $\mathcal{N}_2\big((1,1),1\big)$; 13 support vectors; $w = (0.53, 0.77)$, $b = 0$. For the 2 misclassified samples, the distance to the separating hyperplane is shown (dotted lines).



(c) Non-linear case. 25 i. i. d. samples from a Gaussian distribution $\mathcal{N}\big((0,0),1\big)$ and 25 samples spaced evenly on a circle of radius 5 with i. i. d. Gaussian noise $\mathcal{N}\big((0,0),0.5\big)$. SVM with degree 2 homogeneous polynomial kernel $k(x,x') = \langle x,x'\rangle^2$; 4 support vectors.



(d) Regression. 25 equidistant samples of $f(x) = \frac{1}{x}\sin x$. SVM with $C = 1$, $\epsilon = 0.1$, and a Gaussian kernel with $\sigma = 1$; 13 support vectors.

Figure 2.12 Support vector machine classification (a, b, c) and regression (d) examples. In (a), (b), (c), samples from two classes (orange and blue disks), the separating hyperplane (solid line), the margins (gray dashed lines), and the support vectors (encircled in light blue) are shown. In (d), samples from a function (orange disks), the ϵ-tube around the regressand (gray dashed lines), the regressor (solid line), and support vectors (encircled in light blue) are shown.

Soft margins

In practice, data sets are often not linearly separable due to noise and non-linear patterns in the data. The latter can be addressed with the kernel trick (p. 28); the former is handled by introducing *slack variables* $\xi_i \geq 0$ that measure the extent to which a sample $\boldsymbol{x}_i$ violates the hyperplane (Figure 2.12b). This leads to the *primal soft-margin* formulation of SVMs,

$$\min_{\boldsymbol{w}\in\mathbb{R}^d, b\in\mathbb{R}} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i^p \quad \text{such that} \quad y_i\big(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b\big) + \xi_i \geq 1, \;\; \xi_i \geq 0, \tag{2.86}$$

where p is either 1 (hinge loss) or 2 (quadratic loss), and $0 < C < \infty$ is a parameter controlling the trade-off between margin size and tolerated error.

Non-linear case

The linear SVM described so far uses only inner products of the training samples and the weight vector $\boldsymbol{w}$. Applying the kernel trick (p. 28) requires a different representation of the decision function f, since $\boldsymbol{w}$ is now a vector in feature space and can not be explicitly computed. The representer theorem (Kimeldorf and Wahba, 1971; Schölkopf and Smola, 2002) guarantees the existence of a representation of f as a kernel expansion over the training samples,

$$f(x') = \operatorname{sgn} g(x'), \qquad g(x') = b + \sum_{i=1}^{n} y_i\alpha_i k(x', x_i). \tag{2.87}$$

The $\alpha_i \in [0, C]$ are the solution coefficients. A training sample x_i is called a *support vector*[8] iff $\alpha_i > 0$. The Karush-Kuhn-Tucker conditions (Karush, 1939; Kuhn and Tucker, 1951; Boser et al., 1992; Cortes and Vapnik, 1995) state that $\alpha_i > 0$ iff $y_i\, g(x_i) \leq 1$, i. e., support vectors are training samples that lie on the hyperplane or violate it (Figure 2.12).

Computation

Usually, Equation 2.86 is solved for the α_i by applying Lagrange multipliers to its dual formulation, but the solution can also be computed efficiently in the primal using gradient descent (Chapelle, 2007).

Regression

Support vector regression (Vapnik, 1995; Smola and Schölkopf, 2004) directly extends soft margin SVMs (Equation 2.86) by dropping the sgn in the decision function and requiring $|y_i - g(\boldsymbol{x}_i)|$ to be bounded,

$$\min_{\boldsymbol{w}\in\mathbb{R}^d, b\in\mathbb{R}} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\max\{0, |y_i - g(\boldsymbol{x}_i)| - \epsilon\}. \tag{2.88}$$

The used *ϵ-insensitive loss function* does not penalize the predictor as long as it stays within $\pm\epsilon$ around the known function values.

[8] Note that after the switch to kernels, $x \in \mathcal{X}$ need not be an element of a vector space.

2.4.4 Evaluation

We retrospectively evaluated the performance of SVM classification and regression with ISOAK. Data sets, machine learning algorithms, statistical validation procedures, and performance measures were chosen to allow comparison with the literature, in particular with the study by Fröhlich et al. (2005a).

Data sets

We used the `bbb` data set and the `drug-nondrug`, `cobra:ache`, `cobra:cox2`, `cobra:dhfr`, `cobra:fxa`, `cobra:ppar`, `cobra:thrombin` data sets to assess ISOAK performance on a public data set and subsets of a high-quality pharmacological data set, respectively.

We used the predictive toxicology challenge subsets `ptcfm`, `ptcmm`, `ptcfr`, and `ptcmr`, as well as the `bbb` data set to compare ISOAK to a related graph-based approach from the literature, the optimal assignment kernel (Fröhlich et al., 2005a, 2006).

Algorithms

We employed a soft margin, C-parameter variant of SVMs for binary classification, and a C-parameter variant of SVMs with ϵ-insensitive loss function for regression. In both cases, a modified version of the SVM^{light} package (Joachims, 1999) was used.

Statistical validation

Performance estimation was done using 10 runs of stratified 10-fold cross-validation.

Model selection

The SVM parameter C was optimized on the training folds of each cross-validation run using a uniform grid search in log parameter space, $C \in \left\{ 2^k \;\middle|\; k \in \mathbb{N}, -10 \leq k \leq 11 \right\}$. For SVM regression, we set $\epsilon = 3\sigma\sqrt{\ln n / n}$, as proposed by Cherkassky and Ma (2004). Kernel parameters were optimized by grid search (Table 2.6).

Table 2.6 Kernel parameter grid search.

Parameter	Kernel	Values
α	ISOAK	$\frac{1}{100}, \frac{1}{8}, \frac{2}{8}, \frac{3}{8}, \frac{4}{8}, \frac{5}{8}, \frac{6}{8}, \frac{7}{8}, \frac{99}{100}$
d	polynomial	$1, 2, 3, 4, 5$
γ	Gaussian	$\left\{ 2^k \;\middle|\; k \in \mathbb{N}, -10 \leq k \leq 3 \right\}$

Performance measures

We used the average (over all runs and cross-validation folds) percentage of correctly classified samples, and the correlation coefficient (Matthews, 1975; Baldi et al., 2000)

$$r = \frac{tp \cdot tn - fp \cdot fn}{\sqrt{(tp + fn)(tp + fp)(tn + fp)(tn + fn)}}, \tag{2.89}$$

where tp, tn, fp, fn are the numbers of true positives, true negatives, false positives and false negatives. For regression, we used the squared correlation r^2.

Table 2.7 Retrospective performance of ISOAK, baseline kernel / descriptor combinations, and the optimal assignment kernel (Fröhlich et al., 2005a, 2006). For each data set, the parametrization of each method with best averaged cross-validated performance is shown. Numbers are mean ± standard deviation, best performance per data set in bold face. For the echarge vertex comparison on the `cobra:fxa` data set, $\gamma = 0.1622$; on the `ptcmm` data set, $\gamma = 0.1313$. r = correlation coefficient, pc = percentage of correctly classified samples, poly = polynomial kernel, rbf = radial basis function kernel, gc = Ghose-Crippen descriptor, cats2d = CATS2D descriptor; OA = optimal assignment kernel, OARG = optimal assignment kernel on reduced graphs; parametrizations of our similarity measure are listed in Table 2.4.

(a) Performance of ISOAK and baseline kernel / descriptor combinations.

	Baseline kernels / descriptors			ISOAK		
Data set	Parameters	r	pc	Parameters	r	pc
`drug-nondrug`	rbf/gc, $C = 97$, $\gamma = 2^{-6}$	0.745 ± 0.04	87.2 ± 2.2	dppp/dbond, $C = 2$, $\alpha = 0.875$	$\mathbf{0.777 \pm 0.04}$	$\mathbf{88.9 \pm 2.0}$
`cobra:ache`	rbf/gc, $C = 6$, $\gamma = 2^{-6}$	0.874 ± 0.13	93.2 ± 7.1	delem/none, $C = 1$, $\alpha = 0.875$	$\mathbf{0.926 \pm 0.09}$	$\mathbf{96.0 \pm 5.1}$
`cobra:cox2`	poly/gc, $C = 9$, $d = 3$	$\mathbf{0.861 \pm 0.09}$	$\mathbf{92.9 \pm 4.7}$	dppp/dbond, $C = 2$, $\alpha = 0.875$	0.858 ± 0.09	92.4 ± 4.7
`cobra:dhfr`	rbf/cats2d, $C = 1$, $\gamma = 2^{-2}$	0.983 ± 0.05	99.1 ± 2.6	none/none, $C = 1$, $\alpha = 0.875$	$\mathbf{0.994 \pm 0.03}$	$\mathbf{99.7 \pm 1.6}$
`cobra:fxa`	poly/cats2d, $C = 2$, $d = 5$	0.945 ± 0.05	97.2 ± 2.5	echarge/none, $C = 3$, $\alpha = 0.875$	$\mathbf{0.973 \pm 0.03}$	$\mathbf{98.6 \pm 1.6}$
`cobra:ppar`	rbf/cats2d, $C = 3$, $\gamma = 2^{-2}$	0.822 ± 0.12	90.7 ± 6.5	dppp/none, $C = 3$, $\alpha = 0.625$	$\mathbf{0.989 \pm 0.09}$	$\mathbf{95.2 \pm 4.6}$
`cobra:thrombin`	poly/cats2d, $C = 3$, $d = 4$	0.891 ± 0.07	94.4 ± 3.7	dppp/dbond, $C = 2$, $\alpha = 0.875$	$\mathbf{0.930 \pm 0.06}$	$\mathbf{96.3 \pm 2.9}$

(b) Performance of baseline kernel / descriptor combinations, ISOAK, and other optimal assignment kernels.

		Baseline kernels / descriptors		Optimal assignment		ISOAK	
Dataset	Perf.	Parameters	Performance	Parameters	Performance	Parameters	Performance
`ptcfm`	pc	rbf/gc, $C = 4$, $\gamma = 2^{-4}$	64.1 ± 4.5	OARG	64.0 ± 3.3	delem/dbond, $\alpha = 0.875$	$\mathbf{71.1 \pm 5.9}$
`ptcmm`	pc	rbf/gc, $C = 1$, $\gamma = 2^{-3}$	66.4 ± 3.5	OARG	67.8 ± 2.3	echarge/dbond, $\alpha = 0.125$	$\mathbf{72.8 \pm 5.2}$
`ptcfr`	pc	poly/gc, $C = 110$, $d = 1$	68.4 ± 3.5	OA	66.9 ± 1.6	delem/none, $\alpha = 0.875$	$\mathbf{72.4 \pm 5.5}$
`ptcmr`	pc	poly/cats2d, $C = 67$, $d = 1$	64.9 ± 6.2	OA	63.3 ± 2.4	delem/dbond, $\alpha = 0.75$	$\mathbf{69.4 \pm 6.4}$
`bbb`	r^2	rbf/cats2d, $C = 0.1$, $\gamma = 1$	0.598 ± 0.20	OARG	$\mathbf{0.603 \pm 0.06}$	delem/dbond, $\alpha = 0.625$	0.58 ± 0.178

2.4.5 Results

Table 2.7 shows, for each data set, the best performing parametrization of each method. On 10 out of 12 data sets, ISOAK outperforms standard kernel / descriptor combinations as well as the optimal assignment kernel. On the two remaining data sets, our method performs about as good as its competitors. In Table 2.7a, the correlation coefficients of the baseline methods were within a standard deviation of ISOAK results, except on the `cobra:ppar` data set. In Table 2.7b, performance differences were within one standard deviation for two data sets, `ptcmr` and `bbb`, and above that for the remaining three data sets `ptcfm`, `ptcmm`, and `ptcfr`. The results for the latter were consistent with those of other studies (Byvatov et al., 2003; Müller et al., 2005).

2.5 Conclusions

Graph kernels provide a direct way to compare the topological structure of two compounds without resorting to vectorial descriptors. In this chapter, we gave an overview of graph kernels, and, introduced and retrospectively evaluated the iterative similarity optimal assignment graph kernel.

2.5.1 Summary

Graph kernels are formal similarity measures defined directly on graphs such as the annotated molecular structure graph. They correspond to inner products, and are suitable for kernel-based machine learning approaches to virtual screening. Three major types of graph kernels have been proposed in the literature so far, based on random walks, subgraphs, and optimal vertex assignments. By combining the latter with an iterative graph similarity scheme, we develop the iterative graph similarity optimal assignment kernel. We give an iterative algorithm for its computation, prove the convergence of the algorithm and the uniqueness of the solution, and provide an upper bound on the required number of iterations necessary to achieve a desired precision. In a retrospective virtual screening study using several pharmaceutical and toxicological data sets, our kernel consistently improved performance over chemical descriptors and other optimal assignment type graph kernels.

2.5.2 Graph kernels and virtual screening

Graph kernels (Section 2.2), i. e., positive definite measures of graph similarity for use with kernel-based machine learning, are a recent[9] and active[10] area of research.

Relevance

Graph kernels were retrospectively evaluated for use in virtual screening, ADME / toxicity prediction, and various bioinformatics tasks, with considerable success (Subsection 2.2.7). Prospective virtual screening studies using graph kernels are still rare; in the end, it is such studies that will prove whether graph kernels are indeed useful for ligand-based virtual screening. The prospective study conducted in Chapter 4 for novel inhibitors of the peroxisome proliferator-activated receptor γ using the ISOAK graph kernel developed in this chapter provides first positive hints in this direction.

[9] Starting perhaps with convolution kernels on structured data (Haussler, 1999), with early contributions by Tsuda et al. (2002) and Gärtner et al. (2003).

[10] See, e. g., Borgwardt (2007); Mahé and Vert (2009); Demco (2009); Vishwanathan et al. (2010).

Adaptation to molecular graphs

Graph kernels are a compromise between expressivity and computational complexity, i. e., they trade in separation capability for runtime efficiency (p. 64). Molecular graphs have specific characteristics (Subsection 2.1.3), such as small size and bounded vertex degree. Graph kernels designed for molecular graphs can exploit these properties, attaining higher expressivity than general-purpose graph kernels at acceptable computational cost. Consequently, efforts are being made to develop graph kernels specialized to molecular graphs (Ralaivola et al., 2005; Fröhlich et al., 2006; Rupp et al., 2007; Smalter et al., 2008; Demco, 2009). This work contributes to such efforts.

2.5.3 Iterative similarity optimal assignment kernel

The ISOAK graph kernel (Section 2.3) was designed to take advantage of the characteristics of molecular structure graphs. We discuss some of its properties, the obtained results, and compare it with other graph kernels.

Positive definiteness

It is currently not known whether ISOAK is positive definite (p. 71), but empirical evidence suggests that this is the case for $\alpha \to \infty$. Although positive definiteness is desirable for use with kernel-based machine learning because it allows globally optimal solutions, useful indefinite kernels exist, e. g., the sigmoidal kernel $k(\boldsymbol{x}, \boldsymbol{x}') = \tanh(\kappa \langle \boldsymbol{x}, \boldsymbol{x}' \rangle - \vartheta)$, with $\kappa, \vartheta > 0$ (Schölkopf, 1997). See Haasdonk (2005) for SVMs with indefinite kernels.

Parameter settings

We do not provide a single set of default parameter values because a good choice of parameters depends on the problem at hand (Table 2.7), in particular the choice of vertex and edge kernel. In the absence of any other information, a value of $\alpha = \frac{7}{8}$ seems a reasonable choice from practical experience.

Results

From Table 2.7, ISOAK seems to perform as good as CATS2D and Ghose-Crippen fragment descriptors, with reduced variance, and slightly better than the optimal assignment kernels of Fröhlich et al. (2005a, 2006), at the cost of higher variance. In Chapter 4, we apply ISOAK together with vectorial descriptors, leading to significantly improved results. This suggests that graph kernels might complement traditional chemical descriptors.

Comparison with other graph kernels

ISOAK combines global and local graph similarity in a unique way, setting it apart from other graph kernels.

Random walk kernels (Subsection 2.2.2) implicitly represent molecular graphs by weighted sequences, whereas subgraphs are often believed to be more relevant for chemical problems. The random walk model requires a choice of start, transition and termination probabilities, which on the one hand increases the number of free parameters, but on the other hand provides the possibility to adapt the kernel to problem-specific requirements, e. g., up-weighting of local reactivity centers.

Tree and cyclic pattern graph kernels (Subsections 2.2.3 and 2.2.4) rely on predefined subgraphs. These may be relevant for an application, but do not have to be so in general. Tree kernels have runtime super-exponential in the maximum vertex degree; even for the low degrees of molecular graphs, this constant is quite high, i. e., $7^{2.7} \approx 10^{11.8}$.

The optimal assignment kernels of Fröhlich et al. (2005a, 2006) are most similar to ISOAK, with the difference lying in the definition of pairwise vertex similarity: Where the former employ a fixed topological distance, ISOAK uses a recursive definition that is conceptually more related to the equilibrium states of Markov chains.

ISOAK shares one advantage with optimal assignment kernels in that it provides additional, interpretable information in the form of the pairwise vertex similarity matrix and the computed optimal assignment (Figures 2.10, 2.11).

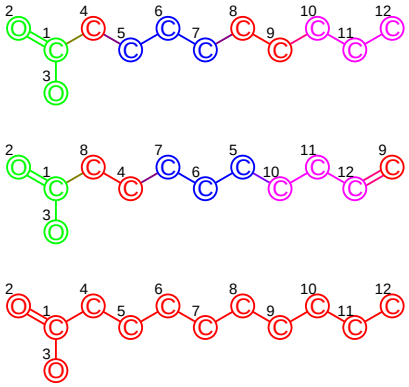
2.5.4 Outlook

We propose ideas for further research on ISOAK, as well as graph kernels and graph models for ligand-based virtual screening.

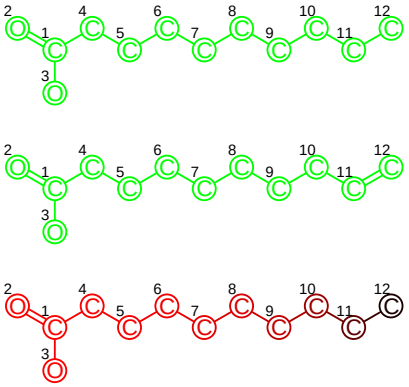
Iterative similarity optimal assignment kernel

We propose ideas for further development of ISOAK:

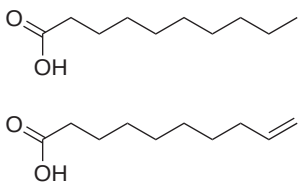
- *Vertex and edge annotation:* Other vertex and bond annotations exist, and might be useful for ligand-based virtual screening with ISOAK, e. g., quantum chemical atom and bond properties, and, E-state indices (Kier and Hall, 1990, 1999).
- *Completeness of ISOAK:* In the limit of maximum recursive similarity $\alpha \to 1$ and infinite precision $\epsilon \to 0$, every vertex influences the similarity of all other vertices, and ISOAK might be complete, i. e., able to separate all non-isomorphic graphs (Figure 2.13). Note that this is in accordance with the complexity results of Gärtner et al. (2003, p. 64) due to the increase in runtime (p. 83).
- *Neighborhood contiguity:* A disadvantage of ISOAK seems to be that vertices are assigned individually, i. e., no attention is paid to preserve neighborhoods in the assignment (Figure 2.13a). Consider matching the structure graph of benzene, where all vertices and edges are identical, against itself. The similarity matrix is the all-ones matrix, and all assignments are optimal. However, assignments that map neighbors of vertices to neighbors of the assigned vertices are preferable. In non-isomorphic graphs, preserving neighborhood contiguity might well be worth small losses in assignment strength. The optimal assignment algorithm (Algorithm 2.2) could be modified by including neighborhood contiguity in the optimality criterion.
- *Multiple alignment:* ISOAK can be used to rigidly align two compounds by finding a transformation matrix (translation and rotation) that minimizes the root mean squared error of the differences in the three-dimensional coordinates of assigned vertices; flexible alignments can be achieved by incorporating conformational flexibility. To align multiple compounds, the final assignment in Algorithm 2.1 needs to be optimal with regard to multiple similarity matrices, requiring the assignment to consider the similarity matrices between all pairs of graphs simultaneously. Alternatively, a stochastic approach in the spirit of simulated annealing (Salamon et al., 2002) might be possible, where the pairwise similarities between vertices of different graphs play the role of an attracting force, and conformational stress provides a repulsive force.



(a) ISOAK assignments for $\alpha = 10^{-4}$ and $\epsilon = 10^{-1}$; overall normalized similarity is 1. Since there is essentially no recursive similarity, pairwise vertex similarity reduces to element type identity and carbon atoms are assigned arbitrarily to each other.



(b) ISOAK assignments for $\alpha = \frac{9}{10}$ and $\epsilon = 10^{-1}$; overall normalized similarity is 0.78. The high impact of recursive similarity assures that each vertex is unique with respect to its neighborhood, as the compounds are not symmetric, and vertices are assigned correctly, with the mismatch in the $\omega - 1$ bond causing a decline of similarity towards the carbon terminus.



(c) Decenoic acid (top) and 9-decenoic acid (caproleic acid, bottom).

Figure 2.13 Completeness and neighborhood contiguity of ISOAK. Shown are assignments of decenoic and caproleic acid (c) for α close to 0 (a) and α close to 1 (b). Computations are with Dirac kernels on element and bond type annotations. In (a) and (b), colors indicate assigned contiguous neighborhoods in decenoic acid (top) and caproleic acid (middle), with red indicating lone vertices. Assignment strength (bottom) is shown color-coded from red (highest similarity) to black (lowest similarity).

Graph kernels

The basic idea of graph kernels for ligand-based virtual screening is to exploit properties of molecular structure graphs (Subsection 2.1.3) to increase expressivity while retaining efficient computability. We propose research ideas for graph kernels on molecular graphs:

- *Efficacy of complete graph kernels:* Complete graph kernels (p. 64) are thought to be desirable due to maximum expressivity, but are hard to compute (Ramon and Gärtner, 2003). Molecular graphs have bounded degree (p. 63); for such graphs, the graph isomorphism problem is in P (Luks, 1982), rendering the complete binary isomorphism graph kernel $k(G, G') = 1_{\{G \text{ isomorphic to } G'\}}$ efficiently computable. While this kernel is complete, it is unlikely to be useful. We propose to investigate the conditions under which a complete graph kernel is useful for ligand-based virtual screening.
- *Efficiency considerations:* The worst-case runtime of an algorithm is of obvious practical importance, but is not the only relevant measure of efficiency. Graph kernels efficient in the average-case, with empirical underlying input distributions, or, randomized graph kernels might provide valuable alternatives. An example of the latter is given by Shervashidze et al. (2009) who introduce graphlets (small subgraph sampling) to compare large graphs.

- *Characteristic subgraphs kernel:* Consider the feature space indexed by all subgraphs. Computing the inner product in this space is NP-hard (Gärtner et al., 2003) due to the large number of subgraphs. However, not all subgraphs will be equally relevant for a given data set. Similar to the determination of scoring matrices for sequence alignments (Eddy, 2004), one could enumerate all subgraphs (up to a given size) in a data set and a reference background data set, compute their log-odds scores, and retain only those subgraphs which are frequent in the training set, but rare in the background set. Although the initial computational requirements are high, computation of the background and training data set subgraphs has to be done only once, while restricting the feature space to the characteristic (or relevant) subgraphs allows efficient computation of the inner product, and might also retain efficacy in cheminformatics learning tasks. Another advantage of this approach is the interpretability of the discovered subgraphs; it also provides a measure of similarity between data sets.

Graph models

We propose an idea for generative use of graph models in ligand-based virtual screening:

- *Generative models:* Instead of using graph models to compare existing molecular graphs, one could use them as generative models to create new structure graphs, e. g., to suggest new compounds in de novo design. One approach, similar to modeling RNA secondary structure with stochastic context-free grammars (Nebel, 2004; Dowell and Eddy, 2004; Metzler and Nebel, 2008), is to create a stochastic grammar generating molecular graphs, learn its probabilities from a set of ligands, and use it to generate new compounds with similar structure and thus properties. Ideally, the grammar would use fragments as terminal symbols and known chemical reactions as rules to ensure synthetic feasibility.

References

Antti Airola, Sampo Pyysalo, Jari Björne, Tapio Pahikkala, Filip Ginter, Tapio Salakoski. *A graph kernel for protein-protein interaction extraction.* In Dina Demner-Fushman, Sophia Ananiadou, Bretonnel Cohen, John Pestian, Jun'ichi Tsujii, Bonnie Webber (editors), *Proceedings of the ACL-08 / HLT Workshop on Current Trends in Biomedical Natural Language Processing (BioNLP 2008), Columbus, Ohio, USA, June 19*, 1–9. Association for Computational Linguistics, 2008.

Gerald Alexanderson. *Euler and Königsberg's bridges: A historical view.* Bulletin of the American Mathematical Society, 43(4): 567–573, 2006.

Pierre Baldi, Søren Brunak, Yves Chauvin, Claus Andersen, Henrik Nielsen. *Assessing the accuracy of prediction algorithms for classification: An overview.* Bioinformatics, 16(5): 412–424, 2000.

Asa Ben-Hur, Cheng Soon Ong, Sören Sonnenburg, Bernhard Schölkopf, Gunnar Rätsch. *Support vector machines and kernels for computational biology.* PLoS Computational Biology, 4(10), 2008.

Kristin Bennett, Colin Campbell. *Support vector machines: Hype or hallelujah?* SIGKDD Explorations, 2(2): 1–13, 2000.

Vincent Blondel, Anahi Gajardo, Maureen Heymans, Paul Van Dooren. *A measure of similarity between graph vertices: Applications to synonym extraction and web searching.* SIAM Review, 46(4): 647–666, 2004.

Hans Bodlaender. *A tourist guide through treewidth.* Acta Cybernetica, 11(1–2): 1–21, 1993.

Karsten Borgwardt. *Graph kernels.* Ph.D. thesis, Faculty for Mathematics, Informatics and Statistics, Ludwig-Maximilians-University Munich, Germany, 2007.

Karsten Borgwardt, Arthur Gretton, Malte Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, Alex Smola. *Integrating structured biological data by kernel maximum mean discrepancy.* Bioinformatics, 22(14): e49–e57, 2006.

Karsten Borgwardt, Hans-Peter Kriegel. *Shortest-path kernels on graphs.* In *Proceedings of the 5th IEEE International Conference on Data Min-*

ing (ICDM 2005), Houston, Texas, USA, November 27–30, 74–81. IEEE Computer Society, 2005.

Karsten Borgwardt, Hans-Peter Kriegel, Vishy Vishwanathan, Nicol Schraudolph. *Graph kernels for disease outcome prediction from protein-protein interaction networks.* In Russ Altman, Keith Dunker, Lawrence Hunter, Tiffany Murray, Teri Klein (editors), *Proceedings of the 12th Pacific Symposium on Biocomputing (PSB 2007), Maui, Hawaii, USA, January 3–7*, 4–15, 2007.

Karsten Borgwardt, Cheng Soon Ong, Stefan Schönauer, Vishy Vishwanathan, Alexander Smola, Hans-Peter Kriegel. *Protein function prediction via graph kernels.* In Hosagrahar Visvesvaraya Jagadish, David States, Burkhard Rost (editors), *Proceedings of the 13th International Conference on Intelligent Systems for Molecular Biology (ISMB 2005), Detroit, USA, June 25–29*, volume 21 (Supplement 1) of *Bioinformatics*, i47–i56. Oxford, 2005.

Bernhard Boser, Isabelle Guyon, Vladimir Vapnik. *A training algorithm for optimal margin classifiers.* In *Proceedings of the 5th Annual ACM Conference on Computational Learning Theory (COLT 1992), Pittsburgh, Pennsylvania, USA, July 27–29, 1992*, 144–152. Association for Computing Machinery, 1992.

François Bourgeois, Jean-Claude Lassalle. *An extension of the Munkres algorithm for the assignment problem to rectangular matrices.* Communications of the ACM, 14(12): 802–804, 1971.

Evgeny Byvatov, Uli Fechner, Jens Sadowski, Gisbert Schneider. *Comparison of support vector machine and artificial neural network systems for drug / nondrug classification.* Journal of Chemical Information and Computer Sciences, 43(6): 1882–1889, 2003.

Rich Caruana, Thorsten Joachims. *KDD cup*, 2004. `http://kodiak.cs.cornell.edu/kddcup/`.

Romeo Cecchelli, Vincent Berezowski, Stefan Lundquist, Maxime Culot, Mila Renftel, Marie-Pierre Dehouck, Laurence Fenart. *Modelling of the blood-brain barrier in drug discovery and development.* Nature Reviews Drug Discovery, 6(8): 650–661, 2007.

Alessio Ceroni, Fabrizio Costa, Paolo Frasconi. *Classification of small molecules by two- and three-dimensional decomposition kernels.* Bioinformatics, 23(16): 2038–2045, 2007.

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

Vladimir Cherkassky, Yunqian Ma. *Practical selection of SVM parameters and noise estimation for SVM regression.* Neural Networks, 17(1): 113–126, 2004.

Michael Collins, Nigel Duffy. *Convolution kernels for natural language.* In Thomas Dietterich, Suzanna Becker, Zoubin Ghahramani (editors), *Advances in Neural Information Processing Systems 14 (NIPS 2001), Vancouver, British Columbia, Canada, December 8–3*, 625–632. MIT Press, 2002.

Donatello Conte, Pasquale Foggia, Carlo Sansone, Mario Vento. *Thirty years of graph matching in pattern recognition.* International Journal of Pattern Recognition and Artificial Intelligence, 18(3): 265–298, 2004.

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. *Introduction to Algorithms.* MIT Press, Cambridge, second edition, 2001.

Corinna Cortes, Vladimir Vapnik. *Support-vector networks.* Machine Learning, 20(3): 273–297, 1995.

Nello Cristianini, John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods.* Cambridge University Press, Cambridge, 2000.

Anthony Demco. *Graph kernel extensions and experiments with application to molecule classification, lead hopping and multiple targets.* Ph.D. thesis, School of Electronics and Computer Science, University of Southampton, England, 2009.

Persi Diaconis. *The Markov chain Monte Carlo revolution.* Bulletin of the American Mathematical Society, 46(2): 179–205, 2009.

Reinhard Diestel. *Graph Theory.* Springer, New York, third edition, 2005.

Robin Dowell, Sean Eddy. *Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction.* BMC Bioinformatics, 5(71), 2004.

Sean Eddy. *Where did the BLOSUM62 alignment score matrix come from?* Nature Biotechnology, 22(8): 1035–1036, 2004.

Robert Edwards. *Drug delivery via the blood-brain barrier.* Nature Neuroscience, 4(3): 221–222, 2001.

Thomas Engel. *Representation of chemical compounds.* In *Chemoinformatics*, 15–167. Wiley-VCH, Weinheim, 2003.

Leonhard Euler. *Solutio problematis ad geometriam situs pertinentis.* Commentarii Academiae Scientiarum Imperialis Petropolitanae, 8: 128–140, 1736.

Committee for Medicinal Products for Human Use, European Medicines Agency. *Guideline on similar biological medicinal products, CHMP / 437 / 04*, 2005.

Jörg Flum, Martin Grohe. *The parameterized complexity of counting problems.* SIAM Journal on Computing, 33(4): 892–922, 2004.

Holger Fröhlich, Jörg Wegner, Florian Sieker, Andreas Zell. *Optimal assignment kernels for attributed molecular graphs.* In Luc de Raedt, Stefan Wrobel (editors), *Proceedings of the 22nd International Conference on Machine Learning (ICML 2005), Bonn, Germany, August 7–11*, 225–232. Omnipress, Madison, WI, USA, 2005a.

Holger Fröhlich, Jörg Wegner, Florian Sieker, Andreas Zell. *Kernel functions for attributed molecular graphs - a new similarity-based approach to ADME prediction in classification and regression.* QSAR & Combinatorial Science, 25(4): 317–326, 2006.

Holger Fröhlich, Jörg Wegner, Andreas Zell. *Assignment kernels for chemical compounds.* In *Proceedings of the 2005 International Joint Conference on Neural Networks (IJCNN 2005), Montréal, Canada, July 31–August 4*, 913–918. IEEE Computer Society, 2005b.

Thomas Gärtner. *A survey of kernels for structured data.* ACM SIG Knowledge Discovery and Data Mining Explorations Newsletter, 5(1): 49–58, 2003.

Thomas Gärtner. *Kernels for Structured Data.* Number 72 in Series in Machine Perception and Artificial Intelligence. World Scientific Publishing, 2009.

Thomas Gärtner, Peter Flach, Stefan Wrobel. *On graph kernels: Hardness results and efficient alternatives.* In Bernhard Schölkopf, Manfred Warmuth (editors), *Learning Theory and Kernel Machines: Proceedings of the 16th Annual Conference on Learning Theory and 7th Kernel Workshop (COLT / Kernel 2003), Washington DC, USA, August 24-27*, volume 2777 of *Lecture Notes in Computer Science*, 129–143. Springer, 2003.

Thomas Gärtner, John Lloyd, Peter Flach. *Kernels and distances for structured data.* Machine Learning, 57(3): 205–232, 2004.

Johann Gasteiger, Mario Marsili. *Iterative partial equalization of orbital electronegativity — a rapid access to atomic charges.* Tetrahedron, 36(22): 3219–3228, 1980.

Petra Gleiss, Peter Stadler. *Relevant cycles in biopolymers and random graphs.* In *Proceedings of the 4th Slovene International Conference in Graph Theory, Lake Bled, Slovenia, June 28–July–2*, 1999.

Andrzej Granas, James Dugundji. *Fixed point theory.* Springer, New York, 2003.

Derek Greene, Pádraik Cunningham. *Practical solutions to the problem of diagonal dominance in kernel document clustering.* In William Cohen, Andrew Moore (editors), *Proceedings of the 23rd International Conference on Machine Learning (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25–29*, 377–384. Association for Computing Machinery, 2006.

Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, Alexander Smola. *A kernel method for the two-sample-problem.* In Bernhard Schölkopf, John Platt, Thomas Hofmann (editors), *Advances in Neural Information Processing Systems 19 (NIPS 2006), Vancouver, Canada, December 4–December 7*, 513–520. MIT Press, 2006.

Dan Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology.* Cambridge University Press, Cambridge, 1997.

Bernard Haasdonk. *Feature space interpretation of SVMs with indefinite kernels.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(4): 482–492, 2005.

Trevor Hambley. *Metal-based therapeutics.* Science, 318(5855): 1392–1393, 2007.

David Haussler. *Convolution kernels on discrete structures. Technical Report UCSC-CRL-99-10*, Department of Computer Science, University of California at Santa Cruz, California, USA, 1999.

Stephen Heller, Alan McNaught. *The IUPAC international chemical identifier (InChI).* Chemistry International, 31(1), 2009.

Christoph Helma, Stefan Kramer. *A survey of the predictive toxicology challenge 2000–2001.* Bioinformatics, 19(10): 1179–1182, 2003.

Andreas Hermann, Matthias Lein, Peter Schwerdtfeger. *The search for the species with the highest coordination number.* Angewandte Chemie International Edition, 46(14): 2444–2447, 2007.

Maureen Heymans, Ambuj Singh. *Deriving phylogenetic trees from the similarity analysis of metabolic pathways.* Bioinformatics, 19(Suppl. 1): i138–i146, 2003.

Roger Horn, Charles Johnson. *Topics in Matrix Analysis.* Cambridge University Press, Cambridge, 1991.

Tamás Horváth. *Cyclic pattern kernels revisited.* In Jaime Carbonell, Jörg Siekmann (editors), *Proceedings of the 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2005), Hanoi, Vietnam, May 18–20*, volume 3518 of *Lecture Notes in Computer Science*, 791–801. Springer, 2005.

Tamás Horváth, Thomas Gärtner, Stefan Wrobel. *Cyclic pattern kernels for predictive graph mining.* In Ronny Kohavi, Johannes Gehrke, William DuMouchel, Joydeep Ghosh (editors), *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2004), Seattle, Washington, USA, August 22–25*, 158–167. ACM Press, 2004.

T.J. Hou, Xiaojie Xu. *ADME evaluation in drug discovery. 3. Modeling blood-brain barrier partitioning using simple molecular descriptors.* Journal of Chemical Information and Computer Sciences, 43(6): 2137–2152, 2003.

Ovidiu Ivanciuc. *Applications of support vector machines in chemistry.* In Kenny Lipkowitz, Tom Cundari (editors), *Reviews in Computational Chemistry*, volume 23, chapter 6, 291–400. Wiley, Hoboken, 2007.

Brijnesh Jain, Peter Geibel, Fritz Wysotzki. *SVM learning with the Schur-Hadamard inner product for graphs.* Neurocomputing, 64: 93–105, 2005.

Tony Jebara, Risi Kondor, Andrew Howard. *Probability product kernels.* Journal of Machine Learning Research, 5: 819–844, 2004.

Glen Jeh, Jennifer Widom. *SimRank: A measure of structural-context similarity.* In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2002), Edmonton, Canada, July 23–26*, 538–543. ACM Press, 2002.

Thorsten Joachims. *Making large-scale SVM learning practical.* In Bernhard Schölkopf, Christopher Burges, Alexander Smola (editors), *Advances in Kernel Methods: Support Vector Learning*, chapter 11, 169–184. MIT Press, Cambridge, 1999.

Thorsten Joachims. *Learning to Classify Text using Support Vector Machines.* Kluwer/Springer, 2002.

David Johnson. *The NP-completeness column.* ACM Transactions on Algorithms, 1(1): 160–176, 2005.

Jaz Kandola, John Shawe-Taylor, Nello Cristianini. *Learning semantic similarity.* In Suzanna Becker, Sebastian Thrun, Klaus Obermayer (editors), *Advances in Neural Information Processing Systems 15 (NIPS 2002), Cambridge, Massachusetts, USA, December 10–12*, 657–664. MIT Press, 2003.

William Karush. *Minima of functions of several variables with inequalities as side constraints.* Master's thesis, Department of Mathematics, University of Chicago, Illinois, USA, 1939.

Hisashi Kashima, Koji Tsuda, Akihiri Inokuchi. *Kernels for graphs.* In Bernhard Schölkopf, Koji Tsuda, Jean-Philippe Vert (editors), *Kernel Methods in Computational Biology*, 155–170. MIT Press, Cambridge, 2004.

Lemont Kier, Lowell Hall. *An electrotopological-state index for atoms in molecules.* Pharmaceutical Research, 7(8): 801–807, 1990.

Lemont Kier, Lowell Hall. *Molecular Structure Description: The Electrotopological State.* Academic Press, New York, 1999.

George Kimeldorf, Grace Wahba. *Some results on Tchebycheffian spline functions.* Journal of Mathematical Analysis and Applications, 33(1): 82–95, 1971.

Jon Kleinberg. *Authoritative sources in a hyperlinked environment.* Journal of the Association for Computing Machinery, 46(5): 604–632, 1999.

Johannes Köbler, Uwe Schöning, Jacobo Torán. *The Graph Isomorphism Problem: Its Structural Complexity.* Springer, New York, 1993.

Risi Kondor, Jean-Philippe Vert. *Diffusion kernels.* In Bernhard Schölkopf, Koji Tsuda, Jean-Philippe Vert (editors), *Kernel Methods in Computational Biology*, 171–191. MIT Press, Cambridge, 2004.

Harold Kuhn. *The Hungarian method for the assignment problem.* Bulletin of the American Mathematical Society, 61: 557–558, 1955.

Harold Kuhn. *Variants of the Hungarian method for assignment problems.* Naval Research Logistics Quarterly, 3: 253–258, 1956.

Harold Kuhn, Albert Tucker. *Nonlinear programming.* In *Proceedings of the 2nd Berkeley Symposium on Mathematical Statistics and Probabilistics*, 481–492. University of California Press, 1951.

Paul Leeson, Brian Springthorpe. *The influence of drug-like concepts on decision-making in medicinal chemistry.* Nature Reviews Drug Discovery, 6(11): 881–890, 2007.

Elizabeth Leicht, Petter Holme, Mark Newman. *Vertex similarity in networks.* Physical Review E, 73(2): 026 120, 2006.

Vladimir Levenshtein. *Binary codes capable of correcting deletions, insertions, and reversals.* Soviet Physics — Doklady, 10(8): 707–710, 1966.

Christopher Lipinski, Franco Lombardo, Beryl Dominy, Paul Feeney. *Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings.* Advanced Drug Delivery Reviews, 23(1–3): 3–25, 1997.

Eugene Luks. *Isomorphism of graphs of bounded valence can be tested in polynomial time.* Journal of Computer and System Sciences, 25(1): 42–65, 1982.

Pierre Mahé, Nobuhisa Ueda, Tatsuya Akutsu, Jean-Luc Perret, Jean-Philippe Vert. *Extensions of marginalized graph kernels.* In Carla Brodley (editor), *Proceedings of the 21st International Conference on Machine Learning (ICML 2004), Banff, Alberta, Canada, July 4–8*, 552–559. Omnipress, Madison, WI, USA, 2004.

Pierre Mahé, Jean-Philippe Vert. *Graph kernels based on tree patterns for molecules.* Machine Learning, 75(1): 3–35, 2009.

Brian Matthews. *Comparison of the predicted and observed secondary structure of T4 phage lysozyme.* Biochimica et Biophysica Acta, 405(2): 442–451, 1975.

Donald McQuarrie. *Quantum Chemistry.* University Science Books, Sausalito, 2007.

Sergey Melnik, Héctor García-Molina, Erhard Rahm. *Similarity flooding: A versatile graph matching algorithm and its application to schema*

matching. In *Proceedings of the 18th International Conference on Data Engineering (ICDE 2002), San Jose, California, USA, February 26–March 1*, 117–128. IEEE Computer Society, 2002.

Sauro Menchetti, Fabrizio Costa, Paolo Frasconi. *Weighted decomposition kernels*. In Luc de Raedt, Stefan Wrobel (editors), *Proceedings of the 22nd International Conference on Machine Learning (ICML 2005), Bonn, Germany, August 7–11*, 585–592. Omnipress, Madison, WI, USA, 2005.

Dirk Metzler, Markus Nebel. *Predicting RNA secondary structures with pseudoknots by MCMC sampling*. Journal of Mathematical Biology, 56(1–2): 161–181, 2008.

Carl Meyer. *Matrix Analysis and Applied Linear Algebra*. Society for Industrial and Applied Mathematics, Philadelphia, 2001.

Klaus-Robert Müller, Gunnar Rätsch, Sören Sonnenburg, Sebastian Mika, Michael Grimm, Nikolaus Heinrich. *Classifying 'drug-likeness' with kernel-based learning methods*. Journal of Chemical Information and Modeling, 45(2): 249–253, 2005.

James Munkres. *Algorithms for the assignment and transportation problems*. Journal of the Society for Industrial and Applied Mathematics, 5(1): 32–38, 1957.

Markus Nebel. *Identifying good predictions of RNA secondary structure*. In Russ Altman, Keith Dunker, Lawrence Hunter, Tiffany Jung, Teri Klein (editors), *Proceedings of the 9th Pacific Symposium on Biocomputing (PSB 2004), Hawaii, USA, January 6–10*, 423–434. World Scientific, 2004.

Michel Neuhaus, Horst Bunke. *An error-tolerant approximate matching algorithm for attributed planar graphs and its application to fingerprint classification*. In Ana Fred, Terry Caelli, Robert Duin, Aurélio Campilho, Dick de Ridder (editors), *Proceedings of the 10th Joint IAPR International Workshops on Structural, Syntactic, and Statistical Pattern Recognition (SSPR 2004, SPR 2004), Lisbon, Portugal, August 18–20*, volume 3138 of *Lecture Notes in Computer Science*, 180–189. Springer, 2004.

Michel Neuhaus, Horst Bunke. *Edit distance-based kernel functions for structural pattern classification*. Pattern Recognition, 39(10): 1852–1863, 2006.

Takao Nishizeki, Norishige Chiba. *Planar Graphs: Theory and Algorithms*. North-Holland, Amsterdam, 1988.

Cheng Soon Ong, Alexander Smola, Robert Williamson. *Learning the kernel with hyperkernels*. Journal of Machine Learning Research, 6(7): 1043–1071, 2005.

Morris Plotkin. *Mathematical basis of of ring-finding algorithms in CIDS*. Journal of Chemical Documentation, 11(1): 60–63, 1971.

Ewgenij Proschak, Matthias Rupp, Swetlana Derksen, Gisbert Schneider. *Shapelets: Possibilities and limitations of shape-based virtual screening*. Journal of Computational Chemistry, 29(1): 108–114, 2008.

Ewgenij Proschak, Jörg Wegner, Andreas Schüller, Gisbert Schneider, Uli Fechner. *Molecular query language (MQL) — a context-free grammar for substructure matching*. Journal of Chemical Information and Modeling, 47(2): 295–301, 2007.

Liva Ralaivola, Sanjay Swamidass, Hiroto Saigo, Pierre Baldi. *Graph kernels for chemical informatics*. Neural Networks, 18(8): 1093–1110, 2005.

Jan Ramon, Thomas Gärtner. *Expressivity versus efficiency of graph kernels*. In Luc de Raedt, Takashi Washio (editors), *Proceedings of the 1st International Workshop on Mining Graphs, Trees and Sequences (MGTS 2003), Cavtat-Dubrovnik, Croatia, September 22–23*, 65–74, 2003.

John Raymond, Peter Willett. *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): 59–71, 2002.

Neil Robertson, Paul Seymour. *Graph minors. II. Algorithmic aspects of tree-width*. Journal of Algorithms, 7(3): 309–322, 1986.

Simon Roger, Ashraf Mikhail. *Biosimilars: Opportunity or cause for concern?* Journal of Pharmacy & Pharmaceutical Sciences, 10(3): 405–410, 2007.

Dennis Rouvray. *The origins of chemical graph theory*. In *Chemical Graph Theory. Introduction and Fundamentals*, 1–41. Taylor & Francis, London, 1991.

Christoph Rücker, Markus Meringer. *How many organic compounds are graph-theoretically nonplanar?* MATCH Communications in Mathematical and in Computer Chemistry, 45: 153–172, 2002.

Matthias Rupp, Ewgenij Proschak, Gisbert Schneider. *Kernel approach to molecular similarity based on iterative graph similarity*. Journal of Chemical Information and Modeling, 47(6): 2280–2286, 2007.

Peter Salamon, Paolo Sibani, Richard Frost. *Facts, Conjectures and Improvements for Simulated Annealing*. SIAM, Philadelphia, 2002.

Petra Schneider, Gisbert Schneider. *Collection of bioactive reference compounds for focused library design*. QSAR & Combinatorial Science, 22(7): 713–718, 2003.

Bernhard Schölkopf. *Support vector learning*. Ph.D. thesis, Technical University of Berlin, Germany, 1997.

Bernhard Schölkopf, Alexander Smola. *Learning with Kernels*. MIT Press, Cambridge, 2002.

David Searls. *The language of genes*. Nature, 420(6912): 211–217, 2002.

Robert Sedgewick. *Permutation generation methods.* ACM Computing Surveys, 9(2): 137–164, 1977.

John Shawe-Taylor, Nello Cristianini. *Kernel methods for pattern analysis.* Cambridge University Press, New York, first edition, 2004.

Nino Shervashidze, Vishy Vishwanathan, Tobias Petri, Kurt Mehlhorn, Karsten Borgwardt. *Efficient graphlet kernels for large graph comparison.* In David van Dyk, Max Welling (editors), *Proceedings of the 12th International Workshop on Artificial Intelligence and Statistics (AISTATS 2009), Clearwater Beach, Florida, USA, April 16–18*, 488–495, 2009.

Hidetoshi Shimodaira. *Improving predictive inference under covariate shift by weighting the log-likelihood function.* Journal of Statistical Planning and Inference, 90(2): 227–244, 2000.

Kilho Shin, Tetsuji Kuboyama. *A generalization of Haussler's convolution kernel — mapping kernel.* In Andrew McCallum, Sam Roweis (editors), *Proceedings of the 25th International Conference on Machine Learning (ICML 2008), Helsinki, Finland, July 5–9*, 944–951. Omnipress, 2008.

Aaron Smalter, Juan Huan, Gerald Lushington. *GPM: A graph pattern matching kernel with diffusion for chemical compound classification.* In *Proceedings of the 8th IEEE International Conference on Bioinformatics and Bioengineering (BIBE 2008), Athens, Greece, October 8–10.* IEEE Computer Society, 2008.

Alex Smola, Bernhard Schölkopf. *A tutorial on support vector regression.* Statistics and Computing, 14(3): 199–222, 2004.

Alexander Smola, Risi Kondor. *Kernels and regularization on graphs.* In Bernhard Schölkopf, Manfred Warmuth (editors), *Learning Theory and Kernel Machines: Proceedings of the 16th Annual Conference on Learning Theory and 7th Kernel Workshop (COLT / Kernel 2003), Washington DC, USA, August 24-27*, volume 2777 of *Lecture Notes in Computer Science*, 144–158. Springer, 2003.

Ingo Steinwart, Andreas Christmann. *Support Vector Machines.* Springer, New York, 2008.

Masashi Sugiyama, Matthias Krauledat, Klaus-Robert Müller. *Covariate shift adaptation by importance weighted cross validation.* Journal of Machine Learning Research, 8(5): 985–1005, 2007a.

Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul von Bünau, Motoaki Kawanabe. *Direct importance estimation with model selection and its application to covariate shift adaptation.* In John Platt, Daphne Koller, Yoram Singer, Sam Roweis (editors), *Advances in Neural Information Processing Systems 20 (NIPS 2007), Vancouver, Canada, December 3–6*, 1433–1440. MIT Press, 2007b.

James Joseph Sylvester. *Chemistry and algebra.* Nature, 17(432): 284, 1878.

Symyx. *CTFile Formats.* Symyx Technologies, `www.mdl.com`, 2007.

Hannu Toivonen, Ashwin Srinivasan, Ross King, Stefan Kramer, Christoph Helma. *Statistical evaluation of the predictive toxicology challenge 2000–2001.* Bioinformatics, 19(10): 1183–1193, 2003.

Koji Tsuda, Taishin Kin, Kiyoshi Asai. *Marginalized kernels for biological sequences.* Bioinformatics, 18(7): S268–S275, 2002.

Esko Ukkonen. *On-line construction of suffix trees.* Algorithmica, 14(3): 249–260, 1995.

Vladimir Vapnik. *The Nature of Statistical Learning Theory.* Springer, New York, first edition, 1995.

Jean-Philippe Vert. *The optimal assignment kernel is not positive definite. Technical Report HAL-00218278*, Centre for Computational Biology, Mines ParisTech, Paris, France, 2008.

Vishy Vishwanathan, Nicol Schraudolph, Risi Kondor, Karsten Borgwardt. *Graph kernels.* Journal of Machine Learning Research, 11(4): 1201–1242, 2010. Preprint at `arxiv.org/abs/0807.0093`.

Vishy Vishwanathan, Alexander Smola. *Fast kernels for string and tree matching.* In Suzanna Becker, Sebastan Thrun, Klaus Obermayer (editors), *Proceedings of the 2002 Neural Information Processing Systems Conference (NIPS 2002)*, volume 15 of *Advances in neural information processing systems.* MIT Press, 2003.

David Weininger. *SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules.* Journal of Chemical Information and Computer Sciences, 28(1): 31–36, 1988.

David Wishart, Craig Knox, An Chi Guo, Dean Cheng, Savita Shrivastava, Dan Tzur, Bijaya Gautam, Murtaza Hassanali. *Drugbank: A knowledgebase for drugs, drug actions and drug targets.* Nucleic Acids Research, 36(database issue): D222–D229, 2008.

Yong-jin Xu, Mark Johnson. *Algorithm for naming molecular equivalence classes represented by labeled pseudographs.* Journal of Chemical Information and Modeling, 41(1): 181–185, 2001.

Yong-jin Xu, Mark Johnson. *Using molecular equivalence numbers to visually explore structural features that distinguish chemical libraries.* Journal of Chemical Information and Modeling, 42(4): 912–926, 2002.

Bianca Zadrozny. *Learning and evaluating classifiers under sample selection bias.* In Carla Brodley (editor), *Proceedings of the 21st International Conference on Machine Learning (ICML 2004), Banff, Alberta, Canada, July 4–8.* Omnipress, Madison, WI, USA, 2004.

Laura Zager, George Verghese. *Graph similarity scoring and matching.* Applied Mathematics Letters, 21(1): 86–94, 2008.

Mohammed Zaki. *Efficiently mining frequent trees in a forest: Algorithms and applications.* IEEE Transactions on Knowledge and Data Engineering, 17(8): 1021–1035, 2005.

Time looking at your data is always well spent.

Witten and Frank (2005)

Chapter 3

Dimensionality reduction and novelty detection

Spectral dimensionality reduction methods like principal components analysis are tools for determining the intrinsic dimensionality and structure of data, and for their visualization. In this chapter, we demonstrate how graph kernels can improve chemical data visualization using kernel versions of spectral dimensionality reduction methods, and, provide proof of principle for novelty detection by kernel principal components analysis. The latter opens up a new way to address the negative samples problem, i. e., virtual screening using only experimentally confirmed positive samples.

3.1 Introduction

Classification algorithms like support vector machines (Subsection 2.4.3) are discriminative models in the sense that they discriminate between two or more classes of samples. Like Gaussian process regression (Subsection 4.2.4), they are supervised techniques that exploit given structural information in the form of labels. In contrast, dimensionality reduction methods are descriptive in nature, and, as unsupervised techniques, aim to discover structure in data.

The intrinsic dimensionality of chemical data sets is, as a rule, considerably lower than the dimensionality of the containing chemical descriptor space (p. 35). Dimensionality reduction methods aim at the identification of the internal structure of such data, enabling visualization (by reducing to two or three dimensions), feature selection (by determining the most relevant descriptors, e. g., sparse principal components analysis), chemical interpretation (by clustering), and novelty detection (by measuring how well new samples fit into the established low-dimensional model of the training data). In the following, we demonstrate how kernel-based versions of established spectral dimensionality reduction algorithms, more recent algorithms, and graph kernels can improve results for these tasks, and how the projection error of kernel principal component analysis can be used for novelty detection.

3.2 Spectral dimensionality reduction

Spectral dimensionality reduction methods rely on the spectrum of a data set, e. g., the eigenvalues of a matrix constructed from the training samples. We provide details for the perhaps best-known algorithm, (linear) principal component analysis, together with its kernel variant, and briefly mention other spectrum-based methods.

3.2.1 Principal component analysis

Principal component analysis (PCA) is a linear dimensionality reduction technique dating back to Pearson (1901) and made popular by Hotelling (1933a,b). It determines orthogonal directions (uncorrelated variables) of maximum variance in a data set, based on an eigendecomposition of the empirical covariance matrix. These directions are called *principal components* (PC). In the following, we briefly describe PCA; for further information, see the book by Jolliffe (2004). Since we are interested in statistical inference, we restrict ourselves to the sample PCA scenario (as opposed to the population PCA scenario, where covariances are known a priori).

Computation

Consider an unsupervised learning setting (p. 27) with i. i. d. vectorial training data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \subset \mathbb{R}^d$. PCA determines a new coordinate system based on the variance-covariance structure of the training data. If the latter occupy a linear subspace of lower dimensionality than the embedding space $\mathbb{R}^d$, the first PCs can be used to approximate this subspace. Whether this approach is appropriate or not depends on the structure of the data (Figure 3.1).

The *covariance* $\operatorname{covar}(A, B) = \mathrm{E}\big((A - \mathrm{E}(A))(B - \mathrm{E}(B))\big)$ of two random variables A and B measures how much A and B vary together.[1] In the case of centered data, where $\mathrm{E}(A) = \mathrm{E}(B) = 0$, the covariance reduces to $\mathrm{E}(AB)$. Let $\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i - \frac{1}{n}\sum_{j=1}^n \boldsymbol{x}_j$ denote the centered training data. Let $\tilde{\boldsymbol{X}} = (\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_n)^T \in \mathbb{R}^{n\times d}$. We call

$$\widetilde{\boldsymbol{C}} = \left(\frac{1}{n}\sum_{k=1}^n \tilde{\boldsymbol{x}}_{k,i}\,\tilde{\boldsymbol{x}}_{k,j}\right)_{i,j=1,\ldots,d} = \frac{1}{n}\sum_{k=1}^n \tilde{\boldsymbol{x}}_k\,\tilde{\boldsymbol{x}}_k^T = \frac{1}{n}\tilde{\boldsymbol{X}}^T\tilde{\boldsymbol{X}} \tag{3.1}$$

the *empirical covariance matrix* of the centered samples.[2] It is symmetric (due to symmetry of covariance) and positive semi-definite.[3] Let $\lambda_1, \ldots, \lambda_d \in \mathbb{R}_{\geq 0}$ and $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_d \in \mathbb{R}^d$ denote the non-negative eigenvalues (in descending order) and corresponding orthonormal eigenvectors of $\widetilde{\boldsymbol{C}}$. Substituting Equation 3.1 into $\widetilde{\boldsymbol{C}}\boldsymbol{v} = \lambda\boldsymbol{v}$, we get

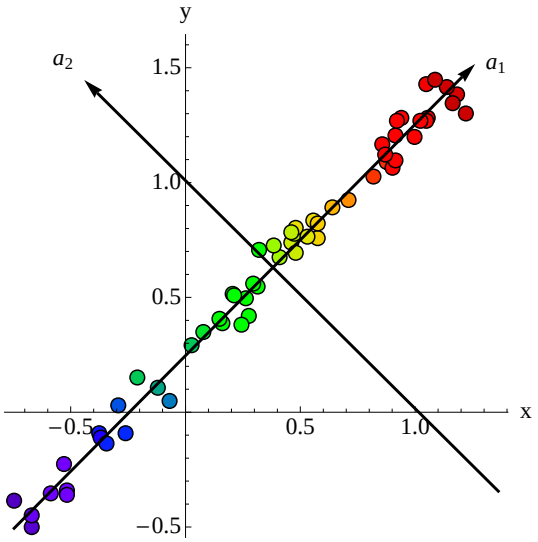
$$\frac{1}{n}\sum_{k=1}^n \langle\tilde{\boldsymbol{x}}_k, \boldsymbol{v}\rangle\,\tilde{\boldsymbol{x}}_k = \lambda\boldsymbol{v}, \tag{3.2}$$

[1] If $\operatorname{covar}(A, B)$ is positive (negative), A and B tend to increase (decrease) together. Independence of A and B implies $\operatorname{covar}(A, B) = 0$, but not vice versa. An example for the latter is $A \in \{1, 2, 3\}$ with probability $\frac{1}{3}$ each, and $B = 1_{\{A=2\}}$.

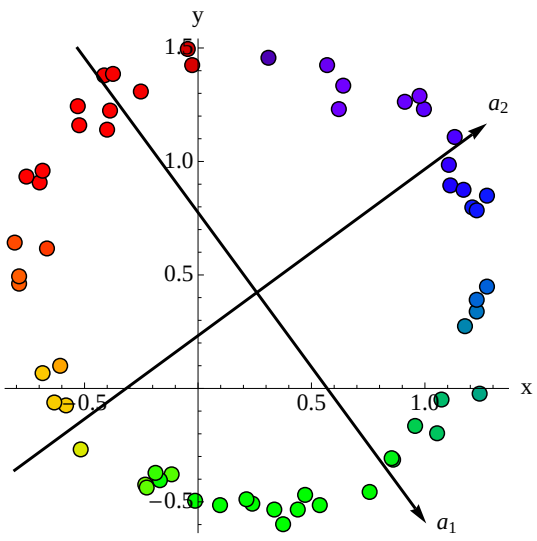
[2] For $\boldsymbol{x}_i$ with non-zero means, $\operatorname{covar}(A, B) = \mathrm{E}(AB) - \mathrm{E}(A)\,\mathrm{E}(B)$. Equation 3.1 then becomes

$$\boldsymbol{C} = \left(\frac{1}{n}\sum_{k=1}^n \boldsymbol{x}_{ki}\,\boldsymbol{x}_{kj} - \frac{1}{n^2}\sum_{k,l=1}^n \boldsymbol{x}_{ki}\,\boldsymbol{x}_{lj}\right)_{i,j=1,\ldots,d} = \frac{1}{n}\boldsymbol{X}^T\boldsymbol{X} - \frac{1}{n^2}\boldsymbol{X}^T\boldsymbol{1}_{n\times n}\boldsymbol{X} = \frac{1}{n}\boldsymbol{X}^T\Big(\boldsymbol{I}_{n\times n} - \frac{1}{n}\boldsymbol{1}_{n\times n}\Big)\boldsymbol{X}.$$

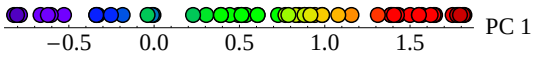
[3] By definition, $\widetilde{\boldsymbol{C}}$ is positive semi-definite iff $\forall\boldsymbol{v} \neq \boldsymbol{0} : \boldsymbol{v}^T\widetilde{\boldsymbol{C}}\boldsymbol{v} \geq 0 \Leftrightarrow \forall\boldsymbol{v} \neq \boldsymbol{0} : \frac{1}{n}\langle\tilde{\boldsymbol{X}}\boldsymbol{v}, \tilde{\boldsymbol{X}}\boldsymbol{v}\rangle \geq 0$.



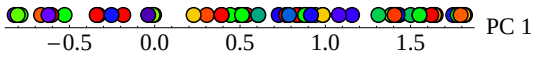
(a) Linear data set `pca1`. $n = 60$ samples of $\bar{\boldsymbol{x}} + (x, x) + (n_1, n_2)$ with $\bar{\boldsymbol{x}} = (\frac{1}{4}, \frac{1}{2})$, $x \sim \mathcal{U}(-1, 1)$ and $n_1, n_2 \sim \mathcal{N}(0, \frac{1}{20})$. The eigenvalues of $\boldsymbol{C} = \left(\begin{smallmatrix} 0.312 & 0.312 \\ 0.312 & 0.317 \end{smallmatrix}\right)$ are $\lambda_1 = 0.627$ and $\lambda_2 = 0.002$; the corresponding PCA coordinate system axes (black arrows) are unique up to sign. The first PC carries 99.6% of the total variance. Here, linear PCA successfully captures the structure of the data and allows dimensionality reduction.



(b) Non-linear circular data set `pca2`. $n = 60$ samples of $\bar{\boldsymbol{x}} + (\sin x, \cos x) + (n_1, n_2)$ with $\bar{\boldsymbol{x}} = (\frac{1}{4}, \frac{1}{2})$, $x \sim \mathcal{U}(0, 2\pi)$ and $n_1, n_2 \sim \mathcal{N}(0, \frac{1}{20})$. The eigenvalues of $\boldsymbol{C} = \left(\begin{smallmatrix} 0.510 & -0.009 \\ -0.009 & 0.516 \end{smallmatrix}\right)$ are $\lambda_1 = 0.522$ and $\lambda_2 = 0.504$. The similar eigenvalues indicate that the PCA coordinate system axes (black arrows) are arbitrary and mainly determined by noise. Here, linear PCA fails to capture the structure of the data.



(c) Projection of `pca1` data on the first PC.



(d) Projection of `pca2` data onto the first PC.

Figure 3.1 Two synthetic data sets for PCA. Data set `pca1` is suitable for linear PCA, whereas capturing the structure of data set `pca2` requires a non-linear form of PCA (Subsection 3.2.2).

Since the left hand side is a projection onto the subspace spanned by $\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_n$, all eigenvectors $\boldsymbol{v}$ with $\lambda \neq 0$ (and only these) lie in this subspace.

The projection of a sample $\tilde{\boldsymbol{x}}_i$ onto the k-th sample principal component $\langle \boldsymbol{v}_k, \cdot \rangle$ is $\langle \boldsymbol{v}_k, \tilde{\boldsymbol{x}}_i \rangle$. Let $\boldsymbol{V} \in \mathbb{R}^{q \times d}$ denote the matrix with rows $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_q$. The projection of $\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_n$ onto the subspace spanned by the first q PCs is given by the rows of $\tilde{\boldsymbol{X}}\boldsymbol{V}^T$. These vectors use PC coordinates; the global coordinates of these points are given by $\langle \boldsymbol{v}_k, \cdot \rangle \, \boldsymbol{v}_k$ and $\tilde{\boldsymbol{X}}\boldsymbol{V}^T\boldsymbol{V}$, respectively.[4]

The *projection error* (also *reconstruction error*) $p_q(\tilde{\boldsymbol{x}}')$ measures the Euclidean distance between a sample $\tilde{\boldsymbol{x}}'$ and its representation using the first q PCs,

$$p_q(\tilde{\boldsymbol{x}}') = \left\| \tilde{\boldsymbol{x}}' - \sum_{k=1}^{q} \langle \boldsymbol{v}_k, \tilde{\boldsymbol{x}}' \rangle \, \boldsymbol{v}_k \right\| = \sqrt{\|\tilde{\boldsymbol{x}}'\|^2 - \left\| \sum_{k=1}^{q} \langle \boldsymbol{v}_k, \tilde{\boldsymbol{x}}' \rangle \, \boldsymbol{v}_k \right\|^2}. \tag{3.3}$$

[4] See Meyer (2001) for details on orthogonal projection. Note that $\left(\tilde{\boldsymbol{X}}\boldsymbol{V}^T\boldsymbol{V}\right)^T = \boldsymbol{V}^T\boldsymbol{V}\tilde{\boldsymbol{X}}^T$.

The second equality follows from Pythagoras theorem.[5] For several test samples (centered with respect to the training samples) given as rows of $\tilde{\boldsymbol{X}}' \in \mathbb{R}^{m \times d}$,

$$p_q(\tilde{\boldsymbol{X}}') = \sqrt{\operatorname{diag}\left(\tilde{\boldsymbol{X}}'(\boldsymbol{I}_{d\times d} - \boldsymbol{V}^T\boldsymbol{V})\tilde{\boldsymbol{X}}'^T\right)}. \tag{3.4}$$

Figure 3.2 illustrates projection coordinates and reconstruction error. Algorithm 3.1 summarizes the computations.

Properties

It can be shown (Jolliffe, 2004) that $\boldsymbol{v}_1$ maximizes the sample variance of $\langle \boldsymbol{v}_1, \tilde{\boldsymbol{x}}_i \rangle$ subject to the constraint $\|\boldsymbol{v}_1\| = 1$, that $\boldsymbol{v}_2$ maximizes the sample variance of $\langle \boldsymbol{v}_2, \tilde{\boldsymbol{x}}_i \rangle$ subject to $\|\boldsymbol{v}_2\| = 1$ and the additional constraint of zero correlation between $\langle \boldsymbol{v}_1, \tilde{\boldsymbol{x}}_i \rangle$ and $\langle \boldsymbol{v}_2, \tilde{\boldsymbol{x}}_j \rangle$, and so on. In general, $\boldsymbol{v}_k$ maximizes the variance of the training data projected onto it under the constraints of unit length and orthogonality to the previous $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{k-1}$. The sample PCs have the following properties (Jolliffe, 2004; Schölkopf and Smola, 2002):

- The first q PCs carry more variance than any other q orthogonal directions. Conversely, the last q PCs carry less variance than any other q orthogonal directions.
- Sample representations using the first q PCs minimize the squared reconstruction error with regard to all other sets of q directions.
- The covariance matrix can be written as $\widetilde{\boldsymbol{C}} = \sum_{k=1}^{d} \lambda_k \boldsymbol{v}_k \boldsymbol{v}_k^T$ (*spectral decomposition*).
- The sample variance[6] of the PCs is given by the eigenvalues, $\operatorname{var}\big(\langle \boldsymbol{v}_k, \tilde{\boldsymbol{x}}_i \rangle\big) = \lambda_k$.

Further properties can be established under assumptions about the distribution of the samples. For pairwise different eigenvalues, the subspace defined by the PCs is unique up to sign; for equal eigenvalues, the subspace is unique up to sign and order (Figure 3.1b). See Subsection 3.2.6 on how to choose the number of eigenvalues q. Section 3.3 provides an in-depth example of using PCA to learn the concept of fatty acids.

In summary, PCA is a linear dimensionality reduction method based on an eigendecomposition of the empirical covariance matrix of the data. Eigenvectors correspond to the principal axes of the maximum variance subspace, whereas eigenvalues correspond to the projected variance of the input data along the eigenvectors. PCA is deterministic, parameter-free (given q), and globally optimal, but limited to second-order statistics.

3.2.2 Kernel principal component analysis

PCA can be expressed in terms of inner products (Equation 3.2); therefore, the kernel trick (p. 28) can be applied, resulting in implicit PCA in feature space, or *kernel PCA*. Since kernel PCA is linear PCA in feature space, it retains the properties of the latter. Although other non-linear generalizations of PCA such as principal curves (Hastie and Stuetzle, 1989) exist, kernel PCA has become widely popular since its introduction by Schölkopf et al. (1998). It is prototypical in the sense that many other spectral dimensionality reduction algorithms can be reduced to it.

[5] See Figure 3.2. Note that the length of the projected sample is the same for PC and global projection coordinates, i. e., $\|\tilde{\boldsymbol{x}}'^T\boldsymbol{V}^T\| = \|\tilde{\boldsymbol{x}}'^T\boldsymbol{V}^T\boldsymbol{V}\|$. This is due to the orthonormality of the PCs, and global coordinates being confined to PCA subspace. Technically, it follows from $\boldsymbol{V}\boldsymbol{V}^T = \boldsymbol{I}_{q\times q}$ and $\langle \boldsymbol{M}\boldsymbol{x}, \boldsymbol{y} \rangle = \langle \boldsymbol{x}, \boldsymbol{M}^T\boldsymbol{y} \rangle$ for a matrix $\boldsymbol{M}$ and vectors $\boldsymbol{x}$, $\boldsymbol{y}$ of compatible dimensions.

[6] Referring to the biased estimator $\operatorname{var}\big(\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}\big) = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i - \boldsymbol{\mu})^2$ with $\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i$.

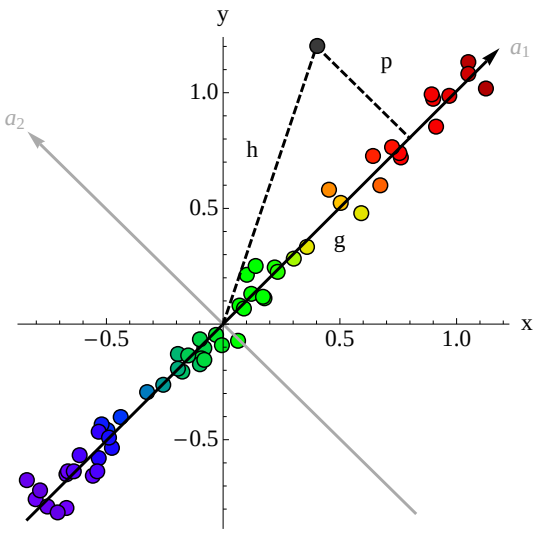


Figure 3.2 Projection coordinates and reconstruction error. When projected onto the first PC $\boldsymbol{v}_1 = (0.7046, 0.7096)$, the sample $\tilde{\boldsymbol{x}} = (0.4, 1.2)$ (black dot) has PC coordinate $\langle \boldsymbol{v}_1, \tilde{\boldsymbol{x}} \rangle = 1.334$ and global coordinates $\langle \boldsymbol{v}_1, \tilde{\boldsymbol{x}} \rangle \boldsymbol{v}_1 = (0.7986, 0.8042)$. Its projection error is $\|\tilde{\boldsymbol{x}} - (0.7986, 0.8042)\| = 0.5617$. The projection error can also be computed using Pythagoras theorem as $p = \sqrt{h^2 - g^2}$.

Algorithm 3.1 Principal component analysis training, projection, and projection error. Samples do not have to be centered.

(a) Principal component analysis training.

Input: sample matrix $\boldsymbol{X} \in \mathbb{R}^{n \times d}$, number of PCs $q \in \mathbb{R}$
Output: eigenvalues $\lambda_1, \ldots, \lambda_q$, eigenvector matrix $\boldsymbol{V} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_q)^T \in \mathbb{R}^{q \times d}$

1 Compute the empirical covariance matrix $\boldsymbol{C} \leftarrow \frac{1}{n} \boldsymbol{X}^T \left(\boldsymbol{I}_{n \times n} - \frac{1}{n} \boldsymbol{1}_{n \times n} \right) \boldsymbol{X}$.
2 Compute eigenvalues $\lambda_1, \ldots, \lambda_q$ and corresponding eigenvectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_q$ of $\boldsymbol{C}$.
3 Normalize eigenvectors $\boldsymbol{v}_i$ by scaling with $\|\boldsymbol{v}_i\|^{-1}$.
4 Return $\lambda_1, \ldots, \lambda_q$ and $\boldsymbol{V} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_q)^T$.

(b) Principal component analysis projection.

Input: test sample matrix $\boldsymbol{X}' \in \mathbb{R}^{m \times d}$, eigenvector matrix $\boldsymbol{V} \in \mathbb{R}^{q \times d}$
Output: matrix of projected samples in PC or global coordinates as row vectors

1 For principal component coordinates, compute $\boldsymbol{X}\boldsymbol{V}^T$.
For global coordinates, compute $\boldsymbol{X}\boldsymbol{V}^T\boldsymbol{V}$.

(c) Principal component analysis projection error.

Input: test sample matrix $\boldsymbol{X}' \in \mathbb{R}^{m \times d}$, eigenvector matrix $\boldsymbol{V} \in \mathbb{R}^{q \times d}$
Output: vector of projection errors

1 Compute $\sqrt{\operatorname{diag}\left(\boldsymbol{X}'(\boldsymbol{I}_{d \times d} - \boldsymbol{V}^T\boldsymbol{V})\boldsymbol{X}'^T \right)}$.

Computation

Let $\tilde{\Phi} : \mathcal{X} \to \mathcal{H}$ denote a map from input space to feature space with $\sum_{i=1}^{n} \tilde{\Phi}(x_i) = 0$, i. e., we assume that samples are centered in feature space.[7] There, Equation 3.2 becomes

$$\frac{1}{n} \sum_{k=1}^{n} \big\langle \tilde{\Phi}(x_k), \boldsymbol{v} \big\rangle \tilde{\Phi}(x_k) = \lambda \boldsymbol{v}. \tag{3.5}$$

Since eigenvectors $\boldsymbol{v}$ with $\lambda \neq 0$ lie in the span of $\tilde{\Phi}(x_1), \ldots, \tilde{\Phi}(x_n)$, we can replace $\widetilde{\boldsymbol{C}}\boldsymbol{v} = \lambda \boldsymbol{v}$ with

$$\big\langle \widetilde{\boldsymbol{C}}\boldsymbol{v}, \tilde{\Phi}(x_k) \big\rangle = \lambda \big\langle \boldsymbol{v}, \tilde{\Phi}(x_k) \big\rangle \quad \text{for all } k,\ 1 \leq k \leq n. \tag{3.6}$$

We express the eigenvectors in terms of the $\tilde{\Phi}(x_k)$ (*dual eigenvector representation*),

$$\boldsymbol{v} = \sum_{k=1}^{n} \boldsymbol{\alpha}_k \tilde{\Phi}(x_k), \tag{3.7}$$

where $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_n)^T \in \mathbb{R}^n$. Combining Equations 3.1, 3.6 and 3.7 yields

$$\begin{aligned}
&\left\langle \left(\frac{1}{n} \sum_{j=1}^{n} \tilde{\Phi}(x_j) \tilde{\Phi}(x_j)^T \right) \left(\sum_{i=1}^{n} \boldsymbol{\alpha}_i \tilde{\Phi}(x_i) \right), \tilde{\Phi}(x_k) \right\rangle = \lambda \left\langle \sum_{i=1}^{n} \boldsymbol{\alpha}_i \tilde{\Phi}(x_i), \tilde{\Phi}(x_k) \right\rangle \quad \text{for all } k \\
\iff\ & \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\alpha}_i \left\langle \sum_{j=1}^{n} \big\langle \tilde{\Phi}(x_j), \tilde{\Phi}(x_i) \big\rangle \tilde{\Phi}(x_j), \tilde{\Phi}(x_k) \right\rangle = \lambda \sum_{i=1}^{n} \boldsymbol{\alpha}_i \big\langle \tilde{\Phi}(x_i), \tilde{\Phi}(x_k) \big\rangle \quad \text{for all } k \\
\iff\ & \sum_{i=1}^{n} \boldsymbol{\alpha}_i \sum_{j=1}^{n} \big\langle \tilde{\Phi}(x_j), \tilde{\Phi}(x_i) \big\rangle \big\langle \tilde{\Phi}(x_j), \tilde{\Phi}(x_k) \big\rangle = n\lambda \sum_{i=1}^{n} \boldsymbol{\alpha}_i \big\langle \tilde{\Phi}(x_i), \tilde{\Phi}(x_k) \big\rangle \quad \text{for all } k \\
\iff\ & \tilde{\boldsymbol{K}}^2 \boldsymbol{\alpha} = n\lambda \tilde{\boldsymbol{K}} \boldsymbol{\alpha},
\end{aligned} \tag{3.8}$$

where $\tilde{\boldsymbol{K}} = \Big(\big\langle \tilde{\Phi}(x_i), \tilde{\Phi}(x_j) \big\rangle \Big)_{i,j=1,\ldots,n}$ is the kernel matrix corresponding to $\tilde{\Phi}$. The following Lemma shows that for our purpose, it is sufficient to instead solve

$$\tilde{\boldsymbol{K}} \boldsymbol{\alpha} = n\lambda \boldsymbol{\alpha}. \tag{3.9}$$

Lemma (Schölkopf et al., 1999)**.** *Solving $\tilde{\boldsymbol{K}}\boldsymbol{\alpha} = n\lambda\boldsymbol{\alpha}$ and $\tilde{\boldsymbol{K}}^2\boldsymbol{\alpha} = n\lambda\tilde{\boldsymbol{K}}\boldsymbol{\alpha}$ yields the same solutions of $\widetilde{\boldsymbol{C}}\boldsymbol{v} = \lambda\boldsymbol{v}$ in feature space with respect to* $\operatorname{span}\big(\tilde{\Phi}(x_1), \ldots, \tilde{\Phi}(x_n)\big)$.

Proof. Let $\mu_1, \ldots, \mu_n \in \mathbb{R}_{\geq 0}$ and $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n \in \mathbb{R}^n$ denote the eigenvalues (in descending order) and corresponding orthonormal eigenvectors of $\tilde{\boldsymbol{K}}$. Assume that λ, $\boldsymbol{\alpha}$ satisfy Equation 3.8. Writing $\boldsymbol{\alpha}$ in terms of the eigenvector basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ of $\tilde{\boldsymbol{K}}$ as $\boldsymbol{\alpha} = \sum_{i=1}^{n} \nu_i \boldsymbol{b}_i$ yields

$$\begin{aligned}
\tilde{\boldsymbol{K}}^2 \sum_{i=1}^{n} \nu_i \boldsymbol{b}_i = n\lambda \tilde{\boldsymbol{K}} \sum_{i=1}^{n} \nu_i \boldsymbol{b}_i \quad &\iff \quad \sum_{i=1}^{n} \nu_i \mu_i^2 \boldsymbol{b}_i = n\lambda \sum_{i=1}^{n} \nu_i \mu_i \boldsymbol{b}_i \\
\iff \quad \nu_i \mu_i^2 = n\lambda \nu_i \mu_i \quad \text{for all } i \quad &\iff \quad \mu_i = n\lambda \vee \mu_i = 0 \vee \nu_i = 0 \quad \text{for all } i,
\end{aligned} \tag{3.10}$$

where the second line follows from the orthogonality of the $\boldsymbol{b}_i$. Applying the same reasoning to Equation 3.9 gives

$$\begin{aligned}
\tilde{\boldsymbol{K}} \sum_{i=1}^{n} \nu_i \boldsymbol{b}_i = n\lambda \sum_{i=1}^{n} \nu_i \boldsymbol{b}_i \quad &\iff \quad \sum_{i=1}^{n} \nu_i \mu_i \boldsymbol{b}_i = n\lambda \sum_{i=1}^{n} \nu_i \boldsymbol{b}_i \\
\iff \quad \nu_i \mu_i = n\lambda \nu_i \quad \text{for all } i \quad &\iff \quad \mu_i = n\lambda \vee \nu_i = 0 \quad \text{for all } i.
\end{aligned} \tag{3.11}$$

[7] See the following page for a way to deal with non-centered samples.

All solutions of Equation 3.11 satisfy Equation 3.10. Conversely, solutions satisfying (3.10) but not (3.11) have some $\mu_i = 0$, and therefore differ only by multiples of eigenvectors $\boldsymbol{b}_i$ of $\tilde{\boldsymbol{K}}$ with eigenvalue 0. While such solutions lead to different coefficients $\boldsymbol{\alpha}$, they do not lead to different solution eigenvectors $\boldsymbol{v}$ with respect to $\operatorname{span}\big(\tilde{\Phi}(x_1), \ldots, \tilde{\Phi}(x_n)\big)$ since the solution vectors corresponding to such $\boldsymbol{b}_i$ are orthogonal to all $\tilde{\Phi}(x_k)$,

$$\Big\langle \tilde{\Phi}(x_k), \sum_{j=1}^{n} \boldsymbol{b}_{i,j} \tilde{\Phi}(x_j) \Big\rangle = \sum_{j=1}^{n} \boldsymbol{b}_{i,j} \big\langle \tilde{\Phi}(x_k), \tilde{\Phi}(x_j) \big\rangle = \big(\tilde{\boldsymbol{K}} \boldsymbol{b}_i\big)_k = 0. \qquad \square$$

We derive a normalization condition on $\boldsymbol{\alpha}$ by normalizing in feature space:

$$\begin{aligned} \|\boldsymbol{v}\| = 1 \Leftrightarrow \langle \boldsymbol{v}, \boldsymbol{v} \rangle = 1 \Leftrightarrow \sum_{i,j=1}^{n} \boldsymbol{\alpha}_i \boldsymbol{\alpha}_j \big\langle \tilde{\Phi}(x_i), \tilde{\Phi}(x_j) \big\rangle = 1 \\ \Leftrightarrow \big\langle \boldsymbol{\alpha}, \tilde{\boldsymbol{K}} \boldsymbol{\alpha} \big\rangle = 1 \Leftrightarrow n\lambda \langle \boldsymbol{\alpha}, \boldsymbol{\alpha} \rangle = 1 \Leftrightarrow \|\boldsymbol{\alpha}\| = \frac{1}{\sqrt{n\lambda}}. \end{aligned} \tag{3.12}$$

The projection of a test sample x' onto the k-th *kernel principal component* $\big\langle \boldsymbol{v}_k, \tilde{\Phi}(\cdot) \big\rangle$ is $\sum_{i=1}^{n} \boldsymbol{\alpha}_{k,i} \big\langle \tilde{\Phi}(x_i), \tilde{\Phi}(x') \big\rangle$. Global coordinate representations are not meaningful in a kernel PCA context, since they require explicit representations of feature space vectors.

Non-centered kernels

To deal with the restriction of samples being centered in feature space, we express $\tilde{\boldsymbol{K}}$ in terms of the original kernel matrix $\boldsymbol{K}$:

$$\begin{aligned} \tilde{\boldsymbol{K}} &= \Big(\big\langle \tilde{\Phi}(x_i), \tilde{\Phi}(x_j) \big\rangle \Big)_{i,j=1,\ldots,n} \\ &= \left(\Big\langle \Phi(x_i) - \frac{1}{n} \sum_{k=1}^{n} \Phi(x_k), \Phi(x_j) - \frac{1}{n} \sum_{k=1}^{n} \Phi(x_k) \Big\rangle \right)_{i,j=1,\ldots,n} \\ &= \left(\big\langle \Phi(x_i), \Phi(x_j) \big\rangle - \frac{1}{n} \sum_{k=1}^{n} \big\langle \Phi(x_i), \Phi(x_k) \big\rangle \right. \\ &\qquad \left. - \frac{1}{n} \sum_{k=1}^{n} \big\langle \Phi(x_j), \Phi(x_k) \big\rangle + \frac{1}{n^2} \sum_{l,m=1}^{n} \big\langle \Phi(x_l), \Phi(x_m) \big\rangle \right)_{i,j=1,\ldots,n} \\ &= \boldsymbol{K} - \frac{1}{n} \boldsymbol{K} \mathbf{1}_{n \times n} - \frac{1}{n} \mathbf{1}_{n \times n} \boldsymbol{K} + \frac{1}{n^2} \mathbf{1}_{n \times n} \boldsymbol{K} \mathbf{1}_{n \times n} \\ &= (\boldsymbol{I}_{n \times n} - \frac{1}{n} \mathbf{1}_{n \times n}) \boldsymbol{K} (\boldsymbol{I}_{n \times n} - \frac{1}{n} \mathbf{1}_{n \times n}). \end{aligned} \tag{3.13}$$

Equation 3.13 is sometimes known as the double centering equation. Analogously, the projection $\big\langle \boldsymbol{v}_k, \tilde{\Phi}(x') \big\rangle$ can be expressed in terms of the original kernel k as

$$\begin{aligned} \big\langle \boldsymbol{v}_k, \tilde{\Phi}(x') \big\rangle &= \sum_{i=1}^{n} \boldsymbol{\alpha}_{k,i} \big\langle \tilde{\Phi}(x_i), \tilde{\Phi}(x') \big\rangle \\ &= \sum_{i=1}^{n} \boldsymbol{\alpha}_{k,i} \Big\langle \Phi(x_i) - \frac{1}{n} \sum_{j=1}^{n} \Phi(x_j), \Phi(x') - \frac{1}{n} \sum_{j=1}^{n} \Phi(x_j) \Big\rangle \\ &= \sum_{i=1}^{n} \boldsymbol{\alpha}_{k,i} \left(k(x_i, x') - \frac{1}{n} \sum_{j=1}^{n} k(x_i, x_j) - \frac{1}{n} \sum_{j=1}^{n} k(x_j, x') + \frac{1}{n^2} \sum_{j,l=1}^{n} k(x_j, x_l) \right), \end{aligned} \tag{3.14}$$

where $\boldsymbol{\alpha}_{k,i} \in \mathbb{R}$ are coefficients of the k-th non-zero eigenvalue solution to Equation 3.9.

Algorithm 3.2 Kernel principal component analysis and projection of test samples.

(a) Kernel principal component analysis training.

Input: kernel matrix $\boldsymbol{K} \in \mathbb{R}^{n \times n}$, number of PCs $q \in \mathbb{N}$, $1 \leq q \leq \min\{\dim(\mathcal{H}), n\}$
Output: eigenvalues $n\lambda_1, \ldots, n\lambda_q$, coefficient matrix $\boldsymbol{A} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_q)^T \in \mathbb{R}^{q \times n}$

1 Compute centered kernel matrix $\tilde{\boldsymbol{K}} \leftarrow (\boldsymbol{I}_{n\times n} - \frac{1}{n}\mathbf{1}_{n\times n})\boldsymbol{K}(\boldsymbol{I}_{n\times n} - \frac{1}{n}\mathbf{1}_{n\times n})$.
2 Compute non-zero eigenvalues $n\lambda_1, \ldots, n\lambda_q$ and eigenvectors $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_q$ of $\tilde{\boldsymbol{K}}$.
3 Normalize eigenvectors $\boldsymbol{\alpha}_i$ by scaling with $1/\sqrt{n\lambda_i \langle \boldsymbol{\alpha}_i, \boldsymbol{\alpha}_i \rangle}$.
4 Return $n\lambda_1, \ldots, n\lambda_q$ and $\boldsymbol{A} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_q)^T$.

(b) Kernel principal component analysis projection.

Input: kernel matrices $\boldsymbol{K} \in \mathbb{R}^{n \times n}$, $\boldsymbol{L} \in \mathbb{R}^{n \times m}$, coefficient matrix $\boldsymbol{A} \in \mathbb{R}^{q \times n}$
Output: matrix of projected samples (rows)

1 Compute $\left(\boldsymbol{L}^T - \frac{1}{n}\mathbf{1}_{m\times n}\boldsymbol{K}\right)\left(\boldsymbol{I}_{n\times n} - \frac{1}{n}\mathbf{1}_{n\times n}\right)\boldsymbol{A}^T$.

Let $\boldsymbol{A} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_q)^T \in \mathbb{R}^{q \times n}$ denote the coefficient matrix of the first q PCs and let $\boldsymbol{L} \in \mathbb{R}^{n \times m}$ with $\boldsymbol{L}_{i,j} = k(x_i, x'_j)$ denote the kernel matrix between training and test samples. The projection of the test samples $x'_1, \ldots, x'_m$ onto the first q PCs is then given by the rows of

$$\begin{aligned}\left(\boldsymbol{A}\boldsymbol{L} - \frac{1}{n}\boldsymbol{A}\boldsymbol{K}\mathbf{1}_{n\times m} - \frac{1}{n}\boldsymbol{A}\mathbf{1}_{n\times n}\boldsymbol{L} + \frac{1}{n^2}\boldsymbol{A}\mathbf{1}_{n\times n}\boldsymbol{K}\mathbf{1}_{n\times m}\right)^T \\ = \left(\boldsymbol{L}^T - \frac{1}{n}\mathbf{1}_{m\times n}\boldsymbol{K}\right)\left(\boldsymbol{I}_{n\times n} - \frac{1}{n}\mathbf{1}_{n\times n}\right)\boldsymbol{A}^T. \qquad (3.15)\end{aligned}$$

Algorithm 3.2 summarizes kernel PCA training and projection computations.

Remarks

We conclude with some remarks on kernel PCA (Schölkopf and Smola, 2002):

- Linear PCA can extract at most $\min\{\dim(\mathcal{X}), n\}$ PCs with non-zero eigenvalues, whereas kernel PCA can extract up to $\min\{\dim(\mathcal{H}), n\}$ such PCs.
- Both linear and kernel PCA allow the reconstruction of the training samples based on all PCs. Linear PCA, if successful, also allows good reconstruction of the samples using only the first PCs. For kernel PCA, however, a pre-image in $\mathcal{X}$ of the approximate reconstruction in $\mathcal{H}$ does not necessarily exist.
- Since centering the training data renders kernel PCA translation invariant, conditionally positive definite kernels can be used.

3.2.3 Isometric feature mapping

The isometric feature mapping (Isomap) algorithm (Tenenbaum et al., 2000) assumes that the data lie on a Riemannian manifold, i. e., in a locally Euclidean subspace. It first approximates geodesic distances (shortest paths along the manifold) between the training data and then uses classic multidimensional scaling (Cox and Cox, 2001; Borg and

Groenen, 2005) to find an embedding preserving these distances. It can be formulated as kernel PCA with a special kernel (Ham et al., 2004).

Computation

For geodesic distance approximation, a symmetric neighborhood graph $G = (V, E)$ is computed, e. g., using k-nearest-neighbors. Then, the all-pairs-shortest-paths problem (Cormen et al., 2001) is solved on G, e. g., with the Floyd-Warshall (Floyd, 1962) algorithm.[8] The length of a shortest path in G between two vertices is an approximation of the distance along the underlying manifold between the corresponding samples. Let $\boldsymbol{S} \in \mathbb{R}^{n \times n}$ denote the matrix of squared geodesic distances. Kernel PCA with

$$\boldsymbol{K} = -\frac{1}{2}\Big(\boldsymbol{I}_{n\times n} - \frac{1}{n}\mathbf{1}_{n\times n}\Big)\boldsymbol{S}\Big(\boldsymbol{I}_{n\times n} - \frac{1}{n}\mathbf{1}_{n\times n}\Big) \tag{3.16}$$

yields the Isomap solution up to a factor of $\sqrt{\lambda_i}$ (Ham et al., 2004). Note that this centers the kernel matrix (removing the need for this step in Algorithm 3.2). Algorithm 3.3 summarizes the procedure.

Positive definiteness

It depends on $\boldsymbol{S}$ whether the kernel matrix in Equation 3.16 is positive definite. If $\boldsymbol{S}$ contains Euclidean distances, e. g., if the geodesic distances are proportional to Euclidean distances in the parameter space of the manifold, $\boldsymbol{K}$ is positive definite. This is due to negative quadratic distance kernels $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = -\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|^{\beta}$ being conditionally positive definite for $0 \leq \beta \leq 2$ and Equation 3.16 being positive definite if and only if $-\boldsymbol{S}$ is conditionally positive definite (Schölkopf and Smola, 2002, pp. 49–51).

Out-of-sample extension

There are several ways to extend Isomap to samples not available during training. We follow Bengio et al. (2004a) by using the precomputed neighborhood graph for geodesic distance approximation and the generalized Isomap kernel

$$k(x, x') = -\frac{1}{2}\left(g^2(x, x') - \frac{1}{n}\sum_{i=1}^{n} g^2(x, x_i) - \frac{1}{n}\sum_{i=1}^{n} g^2(x_i, x') + \frac{1}{n^2}\sum_{i,j=1}^{n} g^2(x_i, x_j)\right), \tag{3.17}$$

where $g(x, x')$ denotes the geodesic distance between x and x'. This leads to

$$\boldsymbol{L} = -\frac{1}{2}\left(\boldsymbol{I}_{n\times n} - \frac{1}{n}\mathbf{1}_{n\times n}\right)\boldsymbol{S}'\left(\boldsymbol{I}_{m\times m} - \frac{1}{m}\mathbf{1}_{m\times m}\right), \tag{3.18}$$

$$\boldsymbol{M} = \frac{1}{n}\mathbf{1}_{1\times n}\boldsymbol{L} + \frac{1}{n^2}\mathbf{1}_{n\times n}K\mathbf{1}_{n\times n}, \tag{3.19}$$

where $\boldsymbol{S}' \in \mathbb{R}^{n\times m}$ denotes the matrix of squared geodesic distances between training and test samples. For details see Algorithm 3.3.

[8] The classic Floyd-Warshall algorithm runs in time $O\big(|V|^3\big)$. Faster algorithms are available, e. g., Chan (2008) with $O\big(|V|^3/\log|V|\big)$. For sparse graphs, $|V|$ applications of Dijkstra's single-source-shortest path algorithm (Dijkstra, 1959; Cormen et al., 2001), using a Fibonacci heap implementation, yields runtime $O\big(|V|^2\log|V| + |E|\,|V|\big) = O\big(|V|^2\log|V|\big)$ since $|E|$ is upper-bounded by $k\,|V|$ in k-nearest-neighbor graphs. This can be improved using more complicated techniques, e. g., Pettie and Ramachandran (2005) with $O\big(|E|\,|V|\log\alpha(|E|,|V|)\big)$, where $\alpha(\cdot,\cdot)$ is the inverse Ackermann function (a constant for all practical purposes). The advantage of the Floyd-Warshall algorithm lies in its simplicity.

Algorithm 3.3 Isometric feature mapping (Isomap). Training and projection of test samples are done using kernel PCA (Algorithm 3.2), with projections scaled by $\sqrt{\lambda_i}$. The necessary matrices $\boldsymbol{K}$, $\boldsymbol{L}$, and $\boldsymbol{M}$ are computed by this algorithm.

(a) Isomap training. If distance matrices are given, step 1 can be omitted. Note that due to the symmetry in the construction procedure, vertices of G can have degree greater than k. G may be disconnected, which can be tested after step 3.

Input: input kernel matrix $\tilde{\boldsymbol{K}} \in \mathbb{R}^{n \times n}$, number of nearest neighbors $k \in \mathbb{N}$
Output: Isomap kernel matrix $\boldsymbol{K} \in \mathbb{R}^{n \times n}$, eigenvalues $n\lambda_1, \ldots, n\lambda_q$, coefficient matrix $\boldsymbol{A} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_q)^T \in \mathbb{R}^{q \times n}$, squared geodesic distances $\boldsymbol{S} \in \mathbb{R}^{n \times n}$

1 Convert $\tilde{\boldsymbol{K}}$ to a Euclidean distance matrix $\boldsymbol{D} \leftarrow \sqrt{\operatorname{diag}(\boldsymbol{K})\mathbf{1}_n{}^T - 2\boldsymbol{K} + \mathbf{1}_n \operatorname{diag}(\boldsymbol{K})^T} \in \mathbb{R}^{n \times n}$ (p. 29).
2 Construct k-nearest neighbor graph $G = (V, E)$ by setting $V \leftarrow \{1, \ldots, n\}$ and for each $v_i \in V$ inserting an undirected edge to its k nearest neighbors as given by $\boldsymbol{D}$.
3 Compute matrix $\boldsymbol{S} \in \mathbb{R}^{n \times n}$ of squared shortest path lengths between all vertices, using, e. g., the Floyd-Warshall algorithm (Footnote 8 on p. 113).
4 Do kernel PCA (Algorithm 3.2) with $\boldsymbol{K} \leftarrow -\frac{1}{2}\boldsymbol{S}$.

To define G using ϵ-balls instead of k-nearest neighbors, replace step 2 by
2 Construct neighborhood graph $G = (V, E)$ by setting $V \leftarrow \{1, \ldots, n\}$ and for each $v_i \in V$ inserting an undirected edge to all neighbors v_j with $\boldsymbol{D}_{i,j} < \epsilon$.

(b) Isomap projection. In case of ties in step 2, shortest path length should be computed using all closest training samples.

Input: input kernel matrices $\tilde{\boldsymbol{K}} \in \mathbb{R}^{n \times n}$, $\tilde{\boldsymbol{L}} \in \mathbb{R}^{n \times m}$, and $\tilde{\boldsymbol{M}} \in \mathbb{R}^m$, Isomap kernel matrix $\boldsymbol{K} \in \mathbb{R}^{n \times n}$, coefficient matrix $\boldsymbol{A} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_q)^T \in \mathbb{R}^{q \times n}$, squared geodesic distances between training samples $\boldsymbol{S} \in \mathbb{R}^{n \times n}$
Output: matrix of projected samples (rows)

1 Convert $\tilde{\boldsymbol{K}}$, $\tilde{\boldsymbol{L}}$, and $\tilde{\boldsymbol{M}}$ to a Euclidean distance matrix $\boldsymbol{D} \leftarrow \sqrt{\operatorname{diag}(\boldsymbol{K})\mathbf{1}_m{}^T - 2\boldsymbol{L} + \mathbf{1}_n \boldsymbol{M}^T} \in \mathbb{R}^{n \times m}$ between training and test samples.
2 For each test sample $x'_{i'}$, $1 \le i' \le m$, determine the closest training sample x_i with $i \leftarrow \arg\min_{1 \le i \le n} \boldsymbol{D}_{ii'}$.
3 Compute squared shortest path length matrix $\boldsymbol{S'} \in \mathbb{R}^{n \times m}$, $\boldsymbol{S'}_{j,i'} \leftarrow (\sqrt{\boldsymbol{S}_{j,i}} + \boldsymbol{D}_{i,i'})^2$.
4 Set $\boldsymbol{L} \leftarrow -\frac{1}{2}\boldsymbol{S'}$.
5 Do kernel PCA projection (Algorithm 3.2) with $\boldsymbol{K}$, $\boldsymbol{L}$, and $\boldsymbol{A}$.
6 Scale i-th projection by $\sqrt{\lambda_i}$.

Landmark Isomap

To reduce computational load, the Isomap authors proposed (de Silva and Tenenbaum, 2003) to use only a (randomly selected) subset of the training samples, called *landmark* points, together with a formula for the embedding of the remaining samples. The latter has been shown (Bengio et al., 2004b) to be equivalent to the Nyström formula and therefore to the computation given by Algorithm 3.3.

3.2.4 Other spectral methods

Many other spectral dimensionality reduction methods exist. We give a brief and non-comprehensive overview.

Laplacian eigenmaps

Laplacian eigenmaps (Belkin and Niyogi, 2002) start with a neighborhood graph $G = (V, E)$ and use the graph Laplacian matrix

$$\big(\boldsymbol{D} - \boldsymbol{A}\big)_{i,j} = \begin{cases} |v_i| & \text{if } i = j \\ -1 & \text{if } i \neq j \wedge \{v_i, v_j\} \in E \\ 0 & \text{otherwise} \end{cases}, \tag{3.20}$$

with $\boldsymbol{D}$ the diagonal matrix of vertex degrees, and $\boldsymbol{A}$ the adjacency matrix (p. 61).

Under certain conditions, the graph Laplacian is an approximation of the Laplace-Beltrami operator, whose eigenfunctions are mappings that optimally preserve the locality of the data. Laplacian eigenmaps can be formulated as kernel PCA with the pseudo-inverse of the graph Laplacian as kernel (Ham et al., 2004).

Diffusion maps

In diffusion maps (Coifman and Lafon, 2006), a normalized kernel matrix derived from a neighborhood graph, e. g., the normalized graph Laplacian, is interpreted as defining a random walk on this graph. This correspondence relates geometric properties of the manifold to properties of the corresponding ergodic Markov chain, with structure given by the neighborhood graph and transition probabilities proportional to the edge weights given by the kernel. Depending on parametrization, the normalized graph Laplacian, the Fokker-Planck operator, and the Laplace-Beltrami operator can be retrieved.

Locally linear embedding

Locally linear embedding (LLE; Roweis and Saul, 2000) uses linear interpolation to locally approximate the underlying manifold. A neighborhood graph defines the size of these locally linear patches; the linear approximations are computed by solving a least squares problem. Locally linear embedding can be formulated as kernel PCA (Ham et al., 2004). It can be seen as an empirical version of Laplacian eigenmaps (Donoho and Grimes, 2003).

Hessian locally linear embedding

Hessian locally linear embedding (also Hessian eigenmaps; Donoho and Grimes, 2003) is a variant of locally linear embedding related to Laplacian eigenmaps. It replaces the Laplacian with an operator based on the Hesse matrix of second derivatives defined using local tangent spaces. For this algorithm, the manifold does not have to be convex.

Local tangent space alignment

Local tangent space alignment (Zhang and Zha, 2004; Wang, 2008), like Hessian locally linear embedding, starts from a neighborhood graph and constructs local linear approximations of the underlying manifold in the form of tangent spaces. These are aligned to obtain a global parametrization of the manifold.

Maximum variance unfolding

Maximum variance unfolding (MVU; also semidefinite embedding, SDE; Weinberger et al., 2004) uses kernel PCA with a data-dependent kernel obtained by solving the semidefinite programming problem $\max_{\boldsymbol{K}} \operatorname{Tr}(\boldsymbol{K})$ subject to

1 $\boldsymbol{K}$ positive semi-definite,

2 $\sum_{i,j=1}^{n} \boldsymbol{K}_{i,j} = 0$ (samples are centered in feature space),

3 $\boldsymbol{K}_{i,i} - 2\boldsymbol{K}_{i,j} + \boldsymbol{K}_{j,j} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2$ iff $\{v_i, v_j\} \in E$ or $\exists v \in V : \{v_i, v\}, \{v, v_j\} \in E$,

where $G = (V, E)$ is the underlying neighborhood graph. The last condition ensures local distance preservation; it can be relaxed by the introduction of slack variables (Weinberger and Saul, 2006). Similar to landmark Isomap, computation time can be reduced by approximating $\boldsymbol{K} \approx \boldsymbol{Q}\boldsymbol{K}'\boldsymbol{Q}^T$ as a product of the kernel matrix between landmark samples $\boldsymbol{K}'$ and a matrix $\boldsymbol{Q}$ obtained by solving a sparse system of linear equations (Weinberger et al., 2005). Unlike in the previously mentioned algorithms, projection of test samples is not immediately possible as it requires the kernel matrix between training and test samples.

Minimum volume embedding

Minimum volume embedding (Shaw and Jebara, 2007) is a variant of maximum variance unfolding that additionally optimizes the eigengap, i. e., the difference between the smallest retained and the largest discarded eigenvalue. The optimization problem then becomes

$$\max_{\boldsymbol{K}} \sum_{i=1}^{q} \lambda_i - \sum_{i=q+1}^{n} \lambda_i \tag{3.21}$$

with the same constraints on $\boldsymbol{K}$ as in maximum variance unfolding. A local optimum of Equation 3.21 guaranteed to improve over initial PCA or MVU solutions can be obtained by solving a sequence of semi-definite programs.

For further information on spectral dimensionality reduction, see Bengio et al. (2006); Saul et al. (2006); Lee and Verleysen (2007).

3.2.5 Projection error-based novelty detection

Novelty detection (p. 27) is the problem of deciding whether new samples come from the same distribution as the training data. It can be used to address the negative samples problem (p. 28), i. e., for ligand-based virtual screening using only positive samples.

Idea

Spectral dimensionality reduction methods that allow the projection of new samples onto the learned manifold can be used for novelty detection by utilizing the projection error (also reconstruction error). The latter measures the error introduced by representing a sample using the learned manifold, or, in other words, the distance between a sample and its projection onto the manifold. If the learned manifold successfully captures the structure of the training data, the projection error will be low for these and similar test data. For data that do not lie on the manifold, i. e., data different from the training data, it will be high. Conceptually, this approach is related to multivariate outlier detection, where samples that are ordinary in the original variables individually may still be outliers if they do not conform to the covariance structure of the training data.

Kernel PCA projection error

Many spectral dimensionality reduction methods are effectively variants of kernel PCA with different kernels; we therefore focus on the kernel PCA projection error (Hoffmann, 2007), or reconstruction error in feature space. Rewriting Equation 3.3 in feature space, we get

$$p_q\big(\tilde{\Phi}(x'_j)\big) = \sqrt{\big\|\tilde{\Phi}(x'_j)\big\|^2 - \Big\|\sum_{k=1}^{q}\big\langle \boldsymbol{v}_k, \tilde{\Phi}(x'_j)\big\rangle \boldsymbol{v}_k\Big\|^2}. \tag{3.22}$$

The first summand $\big\langle\tilde{\Phi}(x'_j), \tilde{\Phi}(x'_j)\big\rangle$ is an evaluation of the centered (with respect to the training samples) kernel $\tilde{k}$.[9] The second summand can be evaluated using Equation 3.14. For matrix notation, we use $\sum_{k=1}^{q}\big\langle \boldsymbol{v}_k, \tilde{\Phi}(x'_j)\big\rangle \boldsymbol{v}_k{}^T = \tilde{\Phi}(x'_j)^T \boldsymbol{V}^T\boldsymbol{V}$ and $\big\|\tilde{\Phi}(x'_j)^T\boldsymbol{V}^T\boldsymbol{V}\big\| = \big\|\tilde{\Phi}(x'_j)^T\boldsymbol{V}^T\big\|$ to get

$$p_q^2\big(\tilde{\Phi}(x'_j)\big) = \big\|\tilde{\Phi}(x'_j)\big\|^2 - \big\|\boldsymbol{V}\tilde{\Phi}(x'_j)\big\|^2 = \big\langle\tilde{\Phi}(x'_j), \tilde{\Phi}(x'_j)\big\rangle - \big\langle\boldsymbol{V}\tilde{\Phi}(x'_j), \boldsymbol{V}\tilde{\Phi}(x'_j)\big\rangle. \tag{3.23}$$

Writing the eigenvectors $\boldsymbol{v}$ in dual representation yields

$$\boldsymbol{V}\tilde{\Phi}(x'_j) = \Big(\sum_{k=1}^{n}\boldsymbol{\alpha}_{i,k}\,\tilde{\Phi}(x_k)^T\Big)_{i=1,\dots,q}\tilde{\Phi}(x'_j) = \Big(\sum_{k=1}^{n}\boldsymbol{\alpha}_{i,j}\big\langle\tilde{\Phi}(x_k), \tilde{\Phi}(x'_j)\big\rangle\Big)_{i=1,\dots,q}. \tag{3.24}$$

Inserting into Equation 3.23 and switching to matrix notation gives

$$p_q\big(\tilde{\boldsymbol{X}}\big) = \sqrt{\operatorname{diag}\big(\tilde{\boldsymbol{M}} - (\boldsymbol{A}\tilde{\boldsymbol{L}})^T\boldsymbol{A}\tilde{\boldsymbol{L}}\big)}, \tag{3.25}$$

where $\tilde{\boldsymbol{M}} = \big(\tilde{k}(x'_i, x'_j)\big)_{i,j} \in \mathbb{R}^{m\times m}$ is the kernel matrix of test samples (centered with respect to the training samples). This is the equivalent of Equation 3.4 in feature space. Algorithm 3.4 summarizes the procedure. Figures 3.3 and 3.4 show examples of projection error-based novelty detection.

Decision threshold

Using the projection error for novelty detection requires a threshold $\tau \in \mathbb{R}_{\geq 0}$: Test samples x' with projection error $p_q(x') \leq \tau$ are considered to belong to the training samples' distribution, whereas samples with $p_q(x') > \tau$ are considered novel.

The choice of τ controls the trade-off between false negatives and false positives, i. e., between sensitivity and specificity. It can be based on statistical considerations, e. g., the distribution of the projection error over the training samples. This approach allows statistical bounds on misclassification rates.

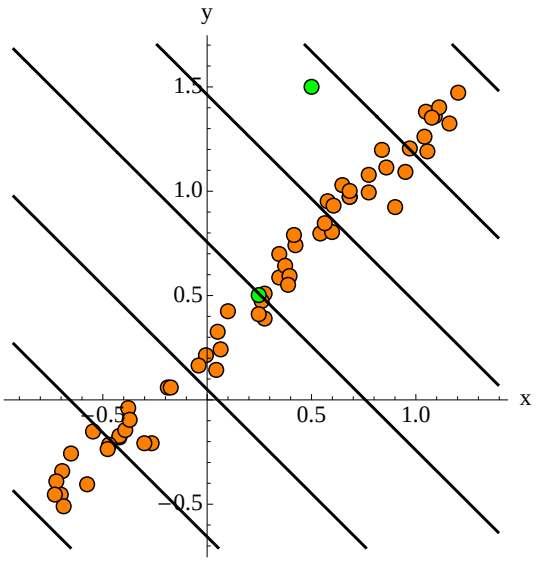
Choices of τ include the maximum projection error on the training samples and quantiles, e. g., the 75 % or 90 % quantiles. The maximum is a poor choice, as it is an extremal value and highly volatile. Figures 3.3 and 3.4 show projection error decision surfaces for the maximum and 75 % quantile in two example data sets, as well as box plots of the projection errors.

[9] A calculation similar to Equation 3.13 shows that

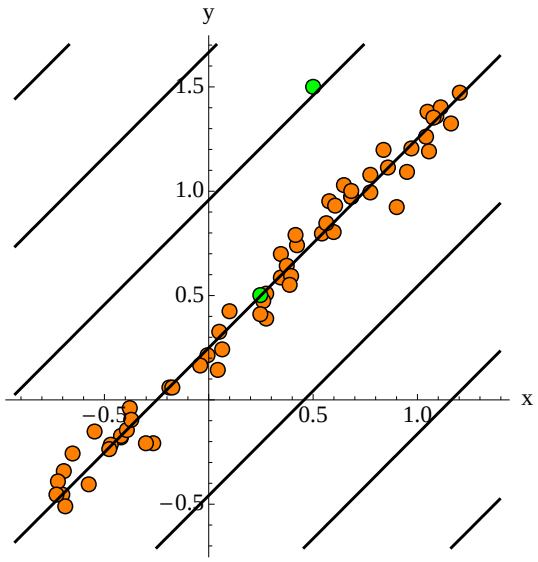
$$\big\langle\tilde{\Phi}(x'_i), \tilde{\Phi}(x'_j)\big\rangle = \big\langle\Phi(x'_i), \Phi(x'_j)\big\rangle - \frac{1}{n}\sum_{k=1}^{n}\big\langle\Phi(x'_i), \Phi(x_k)\big\rangle - \frac{1}{n}\sum_{k=1}^{n}\big\langle\Phi(x'_j), \Phi(x_k)\big\rangle + \frac{1}{n^2}\sum_{k,l=1}^{n}\big\langle\Phi(x_k), \Phi(x_l)\big\rangle,$$

or, in matrix notation,

$$\boldsymbol{M} - \frac{1}{n}\boldsymbol{L}^T\boldsymbol{1}_{n\times m} - \frac{1}{n}\boldsymbol{1}_{m\times n}\boldsymbol{L} + \frac{1}{n^2}\boldsymbol{1}_{m\times n}\boldsymbol{K}\boldsymbol{1}_{n\times m}.$$



(a) Isolines of first principal component.



(b) Isolines of second principal component.

Shown are isolines (lines of equal height; black lines) of both principal components; the isolines run orthogonal to the principal components themselves.

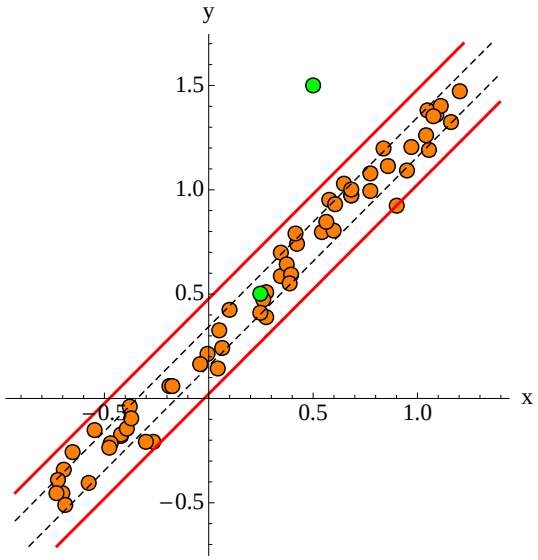
q	min	μ	σ	max
1	$0.356 \cdot 10^{-3}$	0.0445	0.0374	0.161
2	0	0	0	0

q	$Q_{0.25}$	$Q_{0.5}$	$Q_{0.75}$
1	0.0139	0.0330	0.0666
2	0	0	0

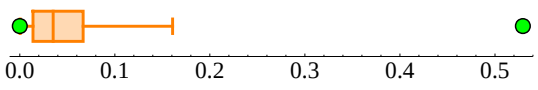
(c) Projection error statistics of training samples. q = number of principal components, min = minimum, μ = mean, σ = standard deviation, max = maximum, $Q_{0.25} = 25\,\%$ quantile, $Q_{0.5}$ = median, $Q_{0.75} = 75\,\%$ quantile.

q	$\boldsymbol{x}'_1$	$\boldsymbol{x}'_2$
1	$0.276 \cdot 10^{-3}$	0.529
2	0	0

(d) Projection error statistics of test samples.

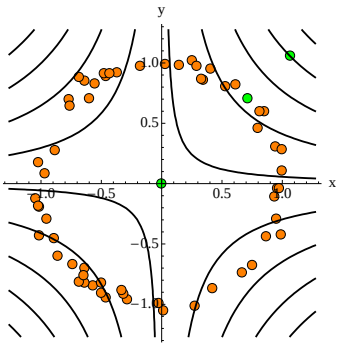


(e) Decision surface of novelty detection for thresholds 0.161 (max, solid red line) and 0.0666 ($Q_{0.75}$, dashed black line) with $q = 1$.

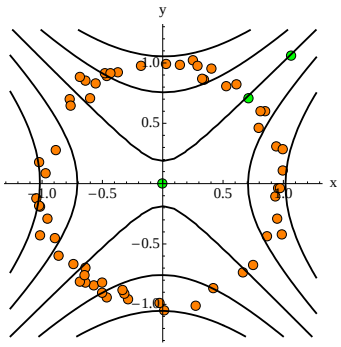


(f) Projection error box plot ($q = 1$).

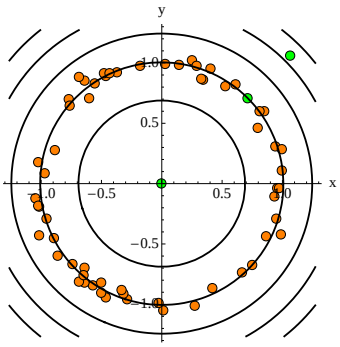
Figure 3.3 Projection error and novelty detection, linear example. 60 training samples (orange disks) generated as $\left(\bar{x} + (x, x) + (n_1, n_2)\right)^T$ with $\bar{x} = (\frac{1}{4}, \frac{1}{2})$, $x \sim \mathcal{U}(-1, 1)$ and $n_1, n_2 \sim \mathcal{N}(0, \frac{1}{20})$, and, two test samples (green disks) $\boldsymbol{x}'_1 = (\frac{1}{4}, \frac{1}{2})^T$ and $\boldsymbol{x}'_2 = (\frac{1}{2}, \frac{3}{2})^T$. The first eigenvalue carries 99.5 % of the total variance.



(a) Isolines of first principal component.



(b) Isolines of second principal component.



(c) Isolines of third principal component.

Shown are isolines (lines of equal height; black lines) of all three principal components; the isolines run orthogonal to the principal components themselves. The eigenvalues cover 0.52 %, 0.47 %, and 0.01 % of the total variance. The last eigenvector captures almost no variance, but encodes the constant-norm invariant of the training data.

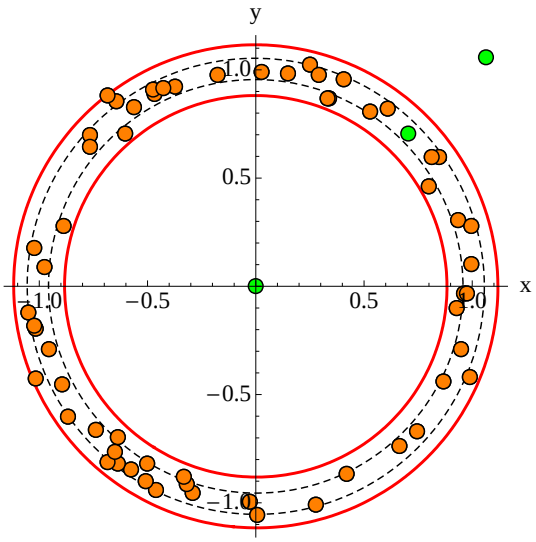
q	min	μ	σ	max
1	0.0429	0.443	0.219	0.800
2	$0.884 \cdot 10^{-3}$	0.0496	0.0378	0.166
3	0	0	0	0

q	$Q_{0.25}$	$Q_{0.5}$	$Q_{0.75}$
1	0.248	0.447	0.639
2	0.0171	0.0450	0.0705
3	0	0	0

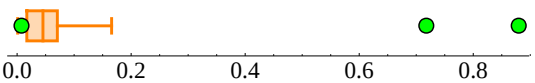
(d) Projection error statistics of training samples. q = number of principal components, min = minimum, μ = mean, σ = standard deviation, max = maximum, $Q_{0.25} = 25\,\%$ quantile, $Q_{0.5}$ = median, $Q_{0.75} = 75\,\%$ quantile.

q	$\boldsymbol{x}'_1$	$\boldsymbol{x}'_2$	$\boldsymbol{x}'_3$
1	0.717	0.0206	0.880
2	0.716	$0.697 \cdot 10^{-2}$	0.880
3	0	0	0

(e) Projection error statistics of test samples.



(f) Decision surface of novelty detection for thresholds 0.161 (max, solid red line) and 0.0666 ($Q_{0.75}$, dashed black line) with $q = 2$.



(g) Projection error box plot ($q = 2$).

Figure 3.4 Projection error and novelty detection, non-linear example. 60 training samples (orange disks) generated as $\big(\sin(x), \cos(x)\big)^T + (n_1, n_2)^T$ with $x \sim \mathcal{U}(-1, 1)$ and $n_1, n_2 \sim \mathcal{N}(0, \frac{1}{20})$, and, three test samples (green disks) $\boldsymbol{x}'_1 = (0, 0)^T$, $\boldsymbol{x}'_2 = \big(\sin(\pi/4), \cos(\pi/4)\big)^T$, and $\boldsymbol{x}'_3 = \frac{3}{2}\boldsymbol{x}'_2$.

Algorithm 3.4 Kernel principal component analysis projection error. The formula for centering a kernel matrix of training versus test samples in step 2 follows from a calculation similar to those of Equation 3.13 and Footnote 9.

Input: kernel matrices $\boldsymbol{K} \in \mathbb{R}^{n \times n}$, $\boldsymbol{L} \in \mathbb{R}^{n \times m}$, $\boldsymbol{M} \in \mathbb{R}^{m \times m}$, coefficient matrix $\boldsymbol{A} \in \mathbb{R}^{q \times n}$
Output: vector of projection errors

1 Center $\boldsymbol{M}$ with respect to the training samples,
$\tilde{\boldsymbol{M}} \leftarrow \boldsymbol{M} - \frac{1}{n}\boldsymbol{L}^T \mathbf{1}_{n \times m} - \frac{1}{n}\mathbf{1}_{m \times n}\boldsymbol{L} + \frac{1}{n^2}\mathbf{1}_{m \times n}\boldsymbol{K}\mathbf{1}_{n \times m}$.
2 Center $\boldsymbol{L}$ with respect to the training samples,
$\tilde{\boldsymbol{L}} \leftarrow (\boldsymbol{I}_{n \times n} - \frac{1}{n}\mathbf{1}_{n \times n})(\boldsymbol{L} - \frac{1}{n}\boldsymbol{K}\mathbf{1}_{n \times m})$.
3 Compute $\sqrt{\operatorname{diag}\left(\tilde{\boldsymbol{M}} - (\boldsymbol{A}\tilde{\boldsymbol{L}})^T \boldsymbol{A}\tilde{\boldsymbol{L}}\right)}$.

3.2.6 Choice of eigenvalues

PCA models come with a free parameter, the number $q \in \mathbb{N}$ of principal components to use. Aside from statistical validation techniques (Subsection 1.4.3) like cross-validation, several eigenspectrum-based heuristics for the choice of q exist. In the following, $p = \min\{\dim \mathcal{H}, n\}$ denotes the number of available kernel PCs. See Jackson (1993) for an empirical comparison and further heuristics.

Fraction of total variance

Choose the lowest number of PCs that still cover a given fraction x of the total variance,

$$q = \operatorname*{arg\,min}_{1 \leq q \leq p} \frac{\sum_{i=1}^{q} \lambda_i}{\sum_{i=1}^{p} \lambda_i} \geq x. \tag{3.26}$$

A common choice is $x = 0.9$.

Largest eigengap

An *eigengap* (also *spectral gap*) is the difference between two eigenvalues (that are neighbors when sorted in descending order), $|\lambda_i - \lambda_{i+1}|$. Choose the q with maximal eigengap,

$$q = \operatorname*{arg\,max}_{1 \leq q \leq p-1} |\lambda_q - \lambda_{q+1}|. \tag{3.27}$$

This criterion is theoretically motivated by the Davis-Kahan theorem (Stewart and Sun, 1990) and related to the stability of the PCs under perturbation (von Luxburg, 2007).

Kaiser-Guttman criterion

Guttman (1954) and Kaiser and Dickman (1959) suggest to retain all PCs with eigenvectors greater than unity,

$$q = \operatorname*{arg\,max}_{1 \leq q \leq p} \lambda_q > 1. \tag{3.28}$$

The motivation for this heuristic is that a PC is not of interest if it explains less variance than one of the original variables. Note that this reasoning assumes standardization of the input data in the space that PCA is carried out in.

Scree plot

The Scree criterion (Cattell, 1966) chooses q to be the number where the most prominent bend in the eigenspectrum plot occurs. This subjective (and manual) criterion can be made more precise by the use of the first derivative, at the risk of numerical problems. The motivation for this heuristic is that eigenvalues of (homoscedastic) random data tend to be of equal size, resulting in the flattened out terminal part of the eigenvalue plot.

In empirical tests of projection error-based novelty detection on several artificial data sets, none of the above criteria seemed suitable for the determination of q. This is exacerbated further by the fact that the decision threshold τ and the number of PCs q are not independent.

3.3 Learning fatty acids

We apply the ideas developed so far to the recognition of fatty acids as a simple but illustrative example, thereby providing proof of principle for their applicability to virtual screening using only positive samples. The advantage of this example is that the target concept is simple enough to be completely understood; this extends to observed phenomena like outliers and the obtained models.

3.3.1 Fatty acids

Fatty acids are the building blocks of lipids, and therefore play an important role in human health and biochemistry in general.

Definition

The international union of pure and applied chemistry (IUPAC) defines fatty acids as

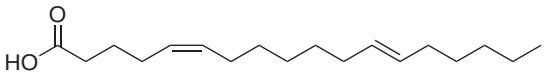
> aliphatic monocarboxylic acids derived from or contained in esterified form in an animal or vegetable fat, oil or wax. Natural fatty acids commonly have a chain of 4 to 28 carbons (usually unbranched and even-numbered), which may be saturated or unsaturated. By extension, the term is sometimes used to embrace all acyclic aliphatic carboxylic acids. (Moss et al., 1995)

There are over 1 000 known natural fatty acids, of which about 20–50 are of common concern; most of these are straight-chained with an even number of carbon atoms between 12 and 22. (Gunstone, 1996) Fatty acids are either *saturated*, having only single bonds in the hydrocarbon chain, or *unsaturated*, having at least one double bond between two carbon atoms. Such a double bond is either in *cis* or in *trans* isomeric form (Scheme 3.1).

For the purposes of this section, we define fatty acids as single-carbon-chain monocarboxylic acids. This excludes branched and cyclic variations, and includes methanoic acid (COOH), ethanoic acid (C-COOH), and, propanoic acid (CC-COOH). The resulting concept is simple enough to be expressed as a regular expression (Sipser, 2005):

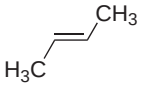
$$\{\mathrm{C}, \backslash\mathrm{C}{=}\mathrm{C}/\mathrm{C}, \backslash\mathrm{C}{=}\mathrm{C}\backslash\mathrm{C}\}^{*}\text{-}\mathrm{COOH},$$

where the star denotes zero or more selections from the bracketed terms and SMILES (Weininger, 1988) syntax has been used to denote cis (\C=C/C) and trans (\C=C\C) isomerism.



7 cis,trans-5,12-octadecadienoic acid

8 cis-2-butene



9 trans-2-butene

Scheme 3.1 cis,trans-5,12-octadecadienoic acid. In fatty acid nomenclature, carbon atoms are counted starting from and including the carboxylic group COOH carbon, with numbers specifying the first atom of a double bond. The name of Compound 7 consists of the isomeric forms of the double bonds (cis, trans), the positions of the double bonds (5, 12), the number of carbon atoms (octadeca) and the number of double bonds (dienoic), giving cis,trans-5,12-octadecadienoic acid. The position of the last double bond can be indicated by counting from the last carbon, named ω, as, e. g., ω-6. The butene isomers 8 and 9 exemplify cis and trans isomerism.

Data sets

The `fattyacids` training data set (Scheme 3.2) contains 85 fatty acids in four series of structurally similar compounds. In this respect, it mimics the composition of real-world virtual screening data sets. The `nonfattyacids` test data set (Scheme 3.3) contains 85 compounds, arranged in 11 series. Two series consist of compounds very different from fatty acids, while the other series consist of decoys structurally similar to fatty acids in different ways.

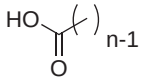
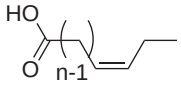
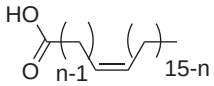
3.3.2 A linear model

The concept of fatty acids (one carboxylic group attached to the end of a straight carbon chain) is simple enough to be captured by a linear model, here PCA. The choice of molecular descriptors is of capital importance, as the model is limited to the information contained in the descriptors.

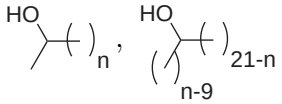
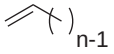
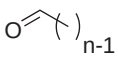
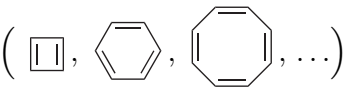
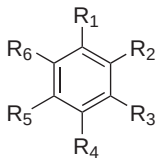
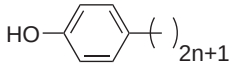
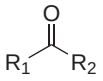
The model

For the linear model `linfa`, we used six simple topological descriptors (Table 3.1), standardized by subtraction of mean and division by standard deviation (on the `fatty-acids` training data set). Constant descriptors were only centered. The descriptors of the `nonfattyacids` test data set were standardized using means and standard deviations computed on the training data set. A PCA resulted in two non-zero eigenvalues $\lambda_1 = 2.98$ and $\lambda_2 = 0.98$, with corresponding eigenvectors $a_1 = (0.316, 0, 0.317, 0.052, 0.316, 0)$ and $a_2 = (0.071, 0, 0.001, 0.858, 0.070, 0)$.

The number of eigenvalues coincides with the two degrees of freedom in fatty acids, the length of the carbon chain, and the degree of its saturation. The first PC weights the descriptors d_c, d_b, d_i equally, puts little weight on d_d and ignores the other descriptors. This reflects the fact that in the carbon chain, the number of carbon atoms, the number of bonds and the diameter are proportional to each other. The small weight on d_d is due to the variance in saturation being lower than the variance in chain length in the training data. With the addition of highly unsaturated fatty acids to the training data, the weight on d_d would increase. The second PC puts weight only on the number of double bonds d_d, reflecting the degree of saturation of the carbon chain.

Id	Structure / name	Series
s_n $1 \leq n \leq 40$		saturated fatty acids
$\omega 3_n$ $1 \leq n \leq 10$		otherwise unsaturated cis isomers of ω-3 fatty acids
od_n $1 \leq n \leq 15$		cis-octadecenoic acids with moving double bond
v_n $1 \leq n \leq 20$	4-pentenoic acid (allylacetic acid) 9-decenoic acid (caproleic acid) 11-dodecenoic acid (11-lauroleic acid) 9-tetradecenoic acid (myristoleic acid) 9-hexadecenoic acid (palmitoleic acid) cis,trans-9,11-octadecadienoic acid (rumenic acid) 9,12-octadecadienoic acid (linoleic acid) 5,9,12-octadecatrienoic acid (pinolenic acid) 6,9,12-octadecatrienoic acid (γ-linolenic acid) cis,trans,cis-8,10,12-octadecatrienoic acid (jacaric acid) cis,trans,cis-9,11,13-octadecatrienoic acid (punicic acid) 9,12,15-octadecatrienoic acid (α-linolenic acid) 6,9,12,15-octadecatrienoic acid (stearidonic acid) 11-eicosenoic acid (gondoic acid) 5,8,11-eicosatrienoic acid (mead acid) 5,8,11,14-eicosatetraenoic acid (arachidonic acid) 5,8,11,14,17-eicosapentaenoic acid (timnodonic acid) 13-docosenoic acid (erucic acid) 7,10,13,16-docosatetraenoic acid (adrenic acid) 15-tetracosenoic acid (nervonic acid)	unsaturated fatty acids

Scheme 3.2 The `fattyacids` training data set, containing 85 fatty acids in four series of 40, 10, 15 and 20 molecules each. If not specified otherwise, all double bonds are in cis isomeric form. The unsaturated fatty acids of the v series occur naturally, except for 4-pentenoic acid.

Id	Structure / name	Series
za_n $1 \le n \le 5$	H_2O, O_2, CO_2, N_2, HCl	tiny molecules
zb_n $1 \le n \le 5$	acetylsalicylic acid, atorvastatin, celecoxib, rosiglitazone, sildenafil[a]	drugs
zc_n $1 \le n \le 10$		primary alcohols
zd_n $1 \le n \le 10$ $11 \le n \le 15$		secondary alcohols
ze_n $1 \le n \le 5$		alkenes
zf_n $1 \le n \le 5$		aldehydes
zg_n $1 \le n \le 10$	[2+2n]annulene ()	annulenes
zh_n $1 \le n \le 5$	 zh_1 R1: OH zh_2 R1: OH, R2: OC, R4: CC=C zh_3 R1: C(=O)O, R3–5: O zh_4 R1: O, R4: CCC(=O)C zh_5 R1: C=CC(=O)O	phenols
zi_n $1 \le n \le 5$		alkylphenols
zj_n $1 \le n \le 5$	 R1–2: C R1: C, R2: CC R1–2: benzene R1: O, R2: CC(=O)C R1: O, R2: CC(O)C	ketones

Continued on next page...

[a]International non-proprietary names (INN; World Health Organization, 1997); corresponding IUPAC names are 2-acetyloxybenzoic acid, (3R,5R)-7-[2-(4-fluorophenyl)-3-phenyl-4-(phenylcarbamoyl)-5-propan-2-ylpyrrol-1-yl]-3,5-dihydroxyheptanoic acid, 4-[5-(4-methylphenyl)-3-(trifluoromethyl)pyrazol-1-yl]benzenesulfonamide, 5-[[4-[2-(methyl-pyridin-2-ylamino)ethoxy]phenyl]methyl]-1,3-thiazolidine-2,4-dione, 5-[2-ethoxy-5-(4-methylpiperazin-1-yl)sulfonylphenyl]-1-methyl-3-propyl-4H-pyrazolo[5,4-e]pyrimidin-7-one.

Scheme 3.3 The `nonfattyacids` test data set, containing 85 non-fatty acids in 11 series of 5, 5, 10, 15, 5, 5, 10, 5, 5, 5 and 15 molecules each. The *za* and *zb* series contain molecules highly different from fatty acids, while the other series contain compounds structurally similar to fatty acids in different aspects. If not specified otherwise, all double bonds are in cis isomeric form.

...continued from previous page.

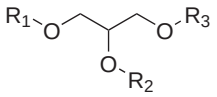
Id	Structure / name	Series
zk_n $1 \le n \le 15$		triglycerides
	zk_1 R1–3: H	
	zk_2 R1–3: C=O	
	zk_3 R1–3: s_{10}	
	zk_4 R1–3: s_{11}	
	zk_5 R1–3: s_{12}	
	zk_6 R1–3: s_{14}	
	zk_7 R1–3: s_{16}	
	zk_8 R1–3: s_{18}	
	zk_9 R1–3: 9-octadecenoic acid	
	zk_{10} R1–3: v_7	
	zk_{11} R1: s_{16}	
	R2: 9,12-heptadecanoic acid	
	R3: 9-hexadecanoic acid	
	zk_{12} R1: s_{18}	
	R2–3: v_5	
	zk_{13} R1: s_{16}	
	R2: v_5	
	R3: v_7	
	zk_{14} R1: v_5	
	R2: 9-octadecenoic acid	
	R3: 6,9-octadecenoic acid	
	zk_{15} R1: 9-hexadecenoic acid	
	R2: v_7	
	R3: v_{16}	

Table 3.1 Descriptors used for the `linfa` model. The first four descriptors have similar means, but higher standard deviations on the test data. Descriptors d_o and d_r are constant on the training data. μ = mean, σ = standard deviation.

		`fattyacids` data set				`nonfattyacids` data set			
Id	Description	min	max	μ	σ	min	max	μ	σ
d_c	number of carbon atoms	1	40	18.0	9.0	0	57	14.7	16.3
d_o	number of oxygen atoms	2	2	2.0	0.0	0	6	2.0	2.1
d_b	number of bonds	3	120	52.4	27.2	1	172	41.3	47.5
d_d	number of double bonds[a]	1	6	1.8	1.1	0	11	2.8	3.2
d_i	diameter of structure graph	2	40	18.1	8.9	0	41	10.9	11.3
d_r	number of rings	0	0	0.0	0.0	0	4	0.4	0.8

[a]For the purposes of this computation, aromatic bonds were counted as double bonds.

Visualization

Projection of training and test data onto the two PCs (Figure 3.5) shows the two types of variance in the training data (variance of carbohydrate chain length and variance in saturation). The projection coordinates alone, however, are not sufficient to separate fatty acids from non-fatty acids.

Novelty detection

The projection error allows almost perfect separation of training and test samples: With the exception of s_1 (methanoic acid), all training samples have projection error less than 0.0056, whereas the lowest projection error of the test samples is 0.0748 (Figure 3.6).

Outliers

Training sample s_1 (methanoic acid) has projection error 0.0853 and is more than nine standard deviations away from the mean training sample projection error (more than 60 standard deviations if mean and standard deviation are computed without s_1). The second largest projection error is 0.0056. The `fattyacids` data set therefore contains methanoic acid as the only outlier.

This is caused by methanoic acid being different from the other fatty acids with regard to the proportionality between the descriptors d_c, d_b and d_i that determine the first PC. Normally, a reduction in d_c and d_b is accompanied by a corresponding reduction in d_i; going from ethanoic acid (s_2) to methanoic acid (s_1), however, reduces d_c and d_b, but leaves d_i unchanged (the longest shortest path is now between the two oxygen atoms).

Setting the projection error threshold to the largest non-outlier projection error in the training data allows perfect recognition of fatty acids. Alternatively, training without the outlier results in a maximum projection error of 0.001 on the training data and a minimum projection error of 0.0794 on the test data, again allowing perfect recognition.

Invariants

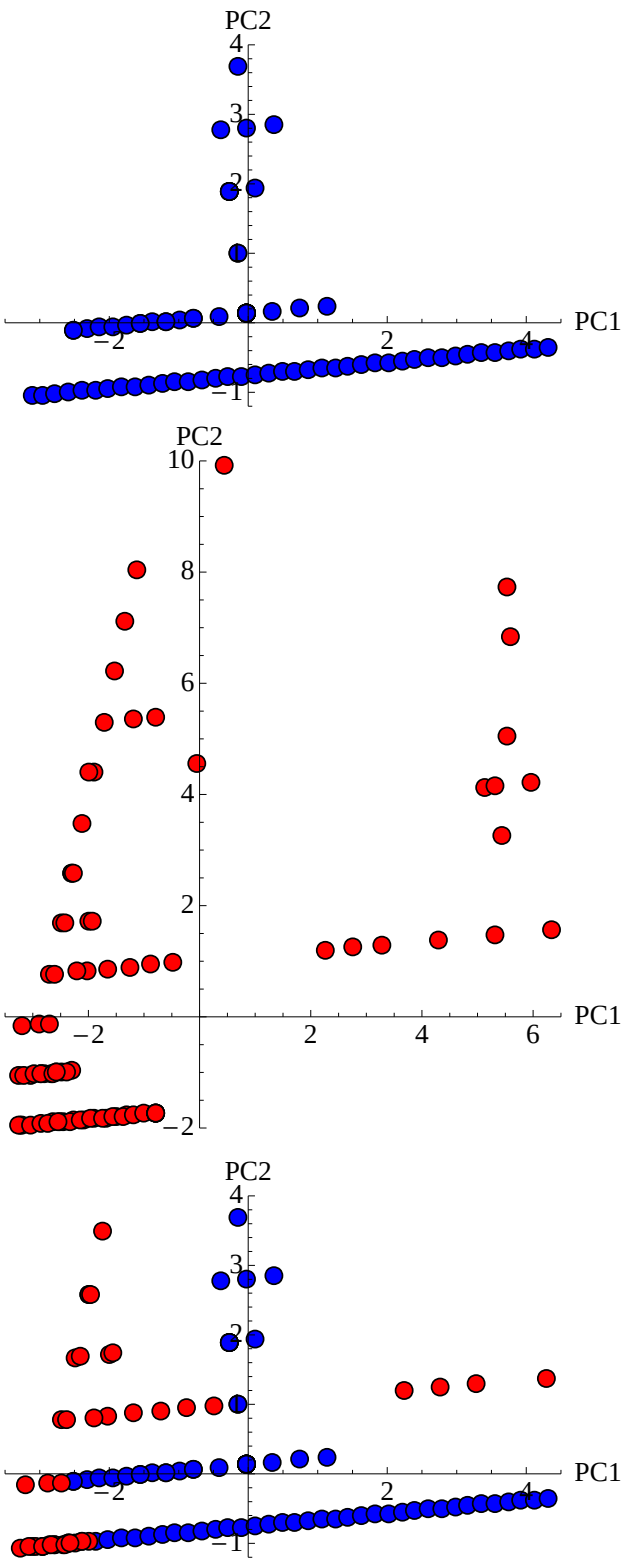
Descriptors constant on the training data can encode invariants of the target concept, e. g., in the `linfa` model the descriptors d_o and d_r encode the containment of two oxygen atoms and no ring structures, respectively. Such descriptors do not vary, and therefore do not contribute to the PCs. They do, however, increase the projection error because they are orthogonal to the PC subspace.[10] Indeed, the average projection error on the `nonfattyacids` test data set decreases by 80.91% from 1.97 down to 0.38 when excluding descriptors d_o and d_r.

Stability

10 runs of 10-fold stratified cross-validation[11] were used to test the stability of the `linfa` model with regard to the composition of the training set, resulting in a ROC AUC of 0.998 ± 0.004 (mean $\pm$ standard deviation). The second eigenvalue and the PCs showed comparably little variation, with first eigenvalue variation slightly elevated (Table 3.2).

[10] Assume that the p-th descriptor is constant, i. e., zero for centered data. Then by definition the p-th row and column of the covariance matrix $\boldsymbol{C}$ are zero. Let $\boldsymbol{e} = (0, \ldots, 0, 1, \ldots, 0)$ denote the vector with p-th entry 1 and let $\boldsymbol{v}$ denote an eigenvector of $\boldsymbol{C}$ with non-zero eigenvalue λ. Then $\langle \boldsymbol{e}, \boldsymbol{v} \rangle = \boldsymbol{v}_p = 0$ due to $(\boldsymbol{C}\boldsymbol{v})_p = (\lambda \boldsymbol{v})_p \Leftrightarrow \sum_{i=1}^{n} \boldsymbol{C}_{p,i} \boldsymbol{v}_i = \lambda \boldsymbol{v}_p \Leftrightarrow 0 = \lambda \boldsymbol{v}_p$.

[11] Novelty detection algorithms simply do not use the novel samples provided in the training folds.

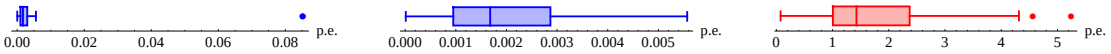


(a) Training data set `fattyacids`. Chain length increases along the x-axis, saturation decreases with the y-axis. The lowest row of points consists of saturated fatty acids (s_n series), the next row contains all fatty acids with one double bond, and so on, until the topmost row, which contains v_{17}, the only pentaenoic fatty acid in the data set. Note that fatty acids with equal chain length and number of double bonds, e. g., $\omega 3_6$ and v_2, are projected onto the same coordinates.

(b) Test data set `nonfattyacids`. The embedding follows the rules given in (a) for the training data, e. g., the zg series stretches from $(-2.8, -0.14)$ to $(-1.1, 8)$, staying on the left due to short overall carbon "chain" length and extending upward due to the large number of double bonds).

(c) Training and test data set. The projection coordinates alone are not enough to separate fatty acids from non-fatty acids. Some test data lie outside of the shown range.

Figure 3.5 Projection of data sets `fattyacids` ($n = 85$, blue disks) and `nonfattyacids` ($n = 85$, red disks) onto the two principal components of the `linfa` model.



(a) Training data set `fatty-acids`.

(b) Training data set `fatty-acids` without the outlier s_1.

(c) Test data set `nonfatty-acids`.

Figure 3.6 Projection error box plots of the `linfa` model. p.e. = projection error.

Table 3.2 Stability of the `linfa` model under 10 runs of 10-fold stratified cross-validation. Albeit lower in absolute terms, relative to their means the second eigenvalue λ_2 shows more variation than the first eigenvalue λ_1 (9 % versus 5 %). Variation of the second PC, albeit low, is about an order of magnitude higher than variation of the first PC. We attribute this to carbon chain length varying more strongly than saturation, and the latters smaller range of values.

Quantity	Mean ± standard deviation
ROC AUC	0.998 ± 0.004
λ_1	2.896 ± 0.138
λ_2	0.971 ± 0.084
a_1	$(0.572, 0, 0.575, -0.114, 0.573, 0) \pm (0.002, 0, 0.000, 0.018, 0.002, 0)$
a_2	$(0.093, 0, 0.010, \quad 0.991, 0.093, 0) \pm (0.010, 0, 0.011, 0.002, 0.011, 0)$

Noise

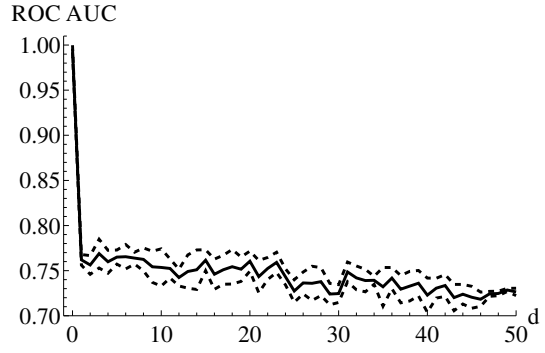
We investigated the influence of noise in the form of additional random dimensions (Figure 3.7). These represent descriptors not related to the target concept. The first added random descriptor leads to a sharp drop in performance, whereas further noise dimensions have little effect. The number of non-zero eigenvalues increases linearly with the number of noise dimensions, since each of these constitutes an independent (orthogonal) attribute of the data.

3.4 Conclusions

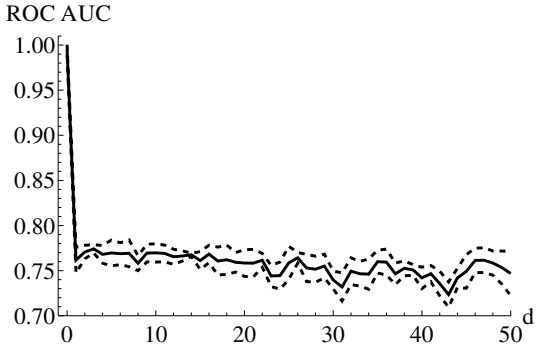
Novelty detection based on the projection error of spectral dimensionality reduction methods can be used for ligand-based virtual screening using only positive samples. In this chapter, we provided an overview of kernel principal component analysis, an algorithm underlying many spectral dimensionality reduction methods, and gave proof of principle of how the projection error can be used in a novelty detection approach to virtual screening.

3.4.1 Summary

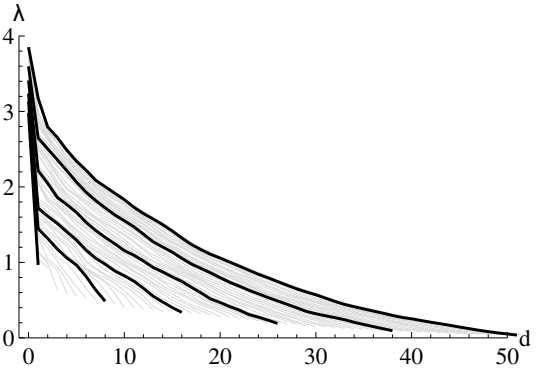
Chemical data sets often lie in subspaces or manifolds of lower dimensionality than the embedding chemical descriptor space. Dimensionality reduction methods allow the identification of these manifolds, effectively providing descriptive models of the data. For spectral dimensionality reduction methods based on kernel principal component analysis, the projection error provides a quantitative measure of how well new samples are described by such models. This can be used for novelty detection, i. e., the identification of compounds structurally dissimilar to the training samples, and thereby for ligand-based virtual screening using only known ligands. As proof of principle, we show how the concept of fatty acids can be learned using principal component analysis.



(a) Gaussian noise.



(b) Uniform noise.



(c) Eigenvalue spectra for Gaussian noise (gray area). The eigenspectra for 0, 7, 15, 25, 37 and 50 additional noise dimensions are highlighted (black lines).

Figure 3.7 Influence of added noise dimensions. Random noise descriptors with zero mean and unit standard deviation were added. Shown are mean (solid lines) ± standard deviation (dashed lines) of 10 runs of 10-fold cross-validation for each added noise dimension d, $0 \leq d \leq 50$. The eigenvalue spectra for uniform noise are similar to those of Gaussian noise.

3.4.2 Dimensionality reduction and novelty detection

We discuss aspects related to virtual screening based on novelty detection and dimensionality reduction.

Previous work

Dimensionality reduction methods, including kernel-based spectral dimensionality reduction methods, have been intensively researched. In contrast, the application of such methods to novelty detection in general, and ligand-based virtual screening using only positive samples in particular, have so far been scarcely investigated. Hoffmann (2007) explores the use of kernel principal component analysis with the Gaussian kernel for novelty detection on artificial data and an optical character recognition benchmark data set. Hristozov et al. (2007) apply self-organizing maps to novelty detection for virtual screening. To the best of our knowledge, projection error-based novelty detection has not been applied to ligand-based virtual screening before.

Assessment

Our proposed approach — ligand-based virtual screening via projection error-based novelty detection — is new, and, although backed up by a detailed example in this chapter, needs further investigation, in particular a prospective study.

The underlying principal component analysis is deterministic, well understood, optimal with regard to a least squares-criterion, and allows formulation of a kernel version. Projection error-based novelty detection is a straight-forward and natural extension, deterministic, interpretable, and applicable to all spectral dimensionality reduction methods that allow projection of new test samples onto the learned model.

Drawbacks and unresolved aspects include the potential susceptibility of principal component analysis to outliers in the data (robustness), and the introduction of two free parameters (the number of principal components q and the novelty detection cut-off τ). It is not clear how our approach performs compared to other methods, most notably the one-class support vector machine (Schölkopf et al., 2001) and its variants like the quarter-sphere support vector machine (Laskov et al., 2004).

3.4.3 Visualization

A classic application of dimensionality reduction methods is visualization of high-dimensional data. In anticipation of Chapter 4, we use kernel PCA with different kernels to visualize a data set of 176 agonists of the peroxisome proliferator-activated receptor (both target and data set are described in detail in Chapter 4, Subsection 4.1 and p. 149). Figures 3.8 and 3.9 show two-dimensional projections of this data set using linear PCA and kernel PCA with two chemical descriptors, as well as kernel PCA with the iterative similarity optimal assignment graph kernel of Chapter 2.

Linear PCA (Figures 3.8a, 3.8b) is not able to resolve the different chemical classes in the data set, the only exception being fatty acids, the class structurally most dissimilar to the other classes, and the related thiazolidinedione-fatty acid hybrids. We attribute this to the non-linearity of the relationships between the compound representations.

Kernel PCA with the Gaussian kernel[12] did not improve results. This was expected, as the Gaussian kernel on the one hand is a universal approximator (p. 30), and therefore suited to novelty detection, but on the other hand requires many dimensions, and is therefore not suited to visualization.

The iterative similarity optimal assignment kernel (Figure 3.9) clearly improves results, and the parametrization with atom pharmacophore types retrieves almost all chemical classes of the data set (when considering the tyrosine classes A, B, and C as one class). Details include the adjoint placement of thiazolidinediones and thiazolidinedione-fatty acid hybrids, where the hybrids are closer to the thiazolidinediones than to the fatty acids, closely reflecting the structural relationships of the classes.

In summary, visualization using kernel PCA in conjunction with the iterative similarity optimal assignment kernel projected most of the chemically motivated classes of the data set into separate clusters. This shows that for these data, our approach is able to reproduce human chemical classification in an unsupervised setting, i. e., without explicit guidance.

3.4.4 Outlook

The introduction of projection error-based novelty detection opens up a new and promising approach to ligand-based virtual screening. We propose ideas for future research:

- *Model selection for q and τ:* In this chapter, we chose the number q of model PCs and the novelty detection cut-off τ based on visual inspection and retrospective novelty

[12] The kernel width σ was optimized using a grid search and a cluster separation criterion together with visual inspection.

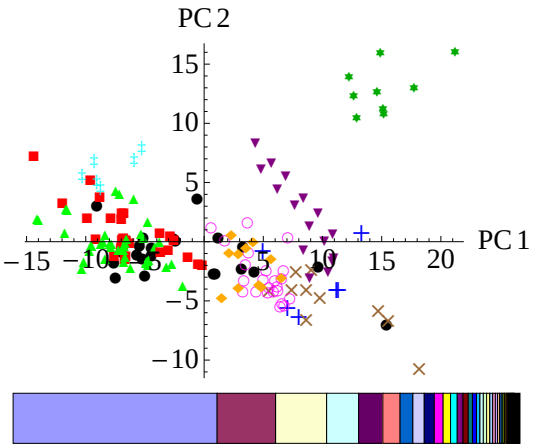
detection performance, which requires negative samples. This is clearly unsatisfactory in a prospective setting. We tested several commonly used criteria for the automatic choice of q (p. 120) without success. A number of model selection approaches in PCA and novelty detection have been proposed, e. g., the Bayesian approach by Hoyle (2008), or, the consistency-based approach by Tax and Müller (2004). Some of these might be suitable for this purpose.

- *Incorporation of training sample distribution along the PCs:* Consider Figure 3.3. The point $\left(\bar{\boldsymbol{x}} + 10^6(1,1)\right)^T$ has zero projection error, but is far away from all training samples. This is because the training samples cover only a finite fraction of the unbounded PCA model subspace. A possible solution is to incorporate into the novelty score the likelihood of the test samples with respect to the empirical distribution given by the projection of the training samples onto each model PC. Alternatively, one could conduct outlier tests on the same projections. In both cases, restriction to the PCs avoids the curse of dimensionality.

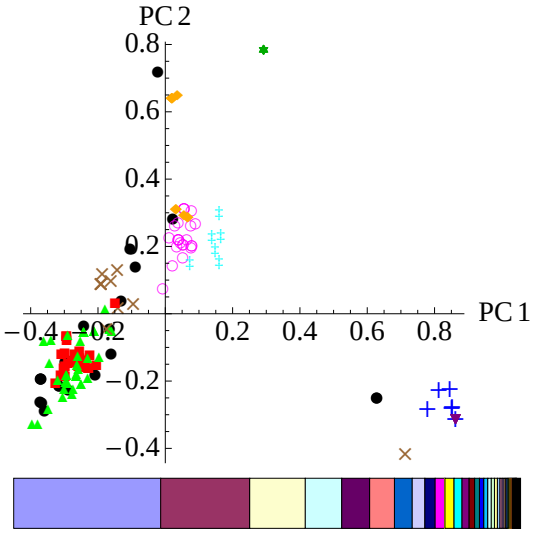
- *Robustness and sparsity:* Two drawbacks of PCA are its susceptibility to outliers, due to the large variance introduced by them, and the density of its solution. Robust (de la Torre and Black, 2003; Deng et al., 2006; Nguyen and De la Torre, 2008; Huang et al., 2009) and sparse (Tipping, 2001; Zou et al., 2006; d'Aspremont et al., 2007; Witten et al., 2009) variants of kernel PCA exist, and could benefit our approach.

- *Domain of applicability:* The idea of using the projection error to measure distance to the training sample manifold is not limited to novelty detection. In (kernel) partial least squares (PLS; also projection to latent structures; Wold et al., 2001), a combination of PCA (Subsection 3.2.1) and multiple linear regression (Rice, 2006), the projection error is a natural measure of the domain of applicability (p. 49).

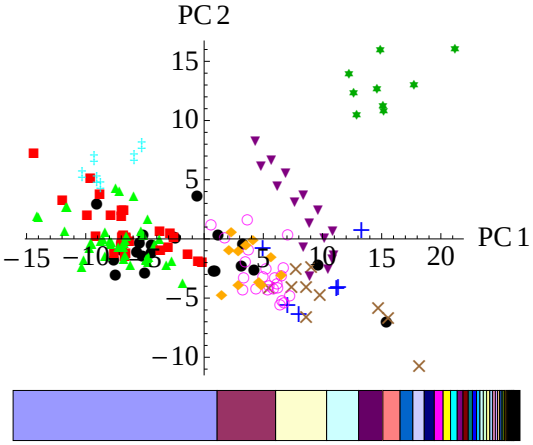
- *Projection of new samples for minimum volume embedding:* A first study (Lasitschka, 2009) indicates that minimum volume embedding (p. 116), a spectral dimensionality reduction method specialized to subspaces of low preset dimensionality, can improve visualization of chemical data sets even further. Unfortunately, it does not allow for the projection (error) of new samples (out-of-sample data), as it uses a data dependent kernel, i. e., it provides no explicit kernel function. Chin and Suter (2008) approximate the maximum variance unfolding (p. 116) kernel function by a series expansion of Gaussian basis functions, similar to the approach by Schwaighofer et al. (2004). Alternatively, one could solve the original optimization problem again for each test sample while keeping the already computed kernel values of the training data fixed.



(a) Linear PCA, MOE 2D descriptors. The first two PCs explain 51.8 % of the total variance; 15 PCs cover 90 % of it. While fatty acids (∗) and thiazolidinedione-fatty acid hybrids (▼) form distinguishable clusters, other classes are clustered, but not distinguishable from each other.

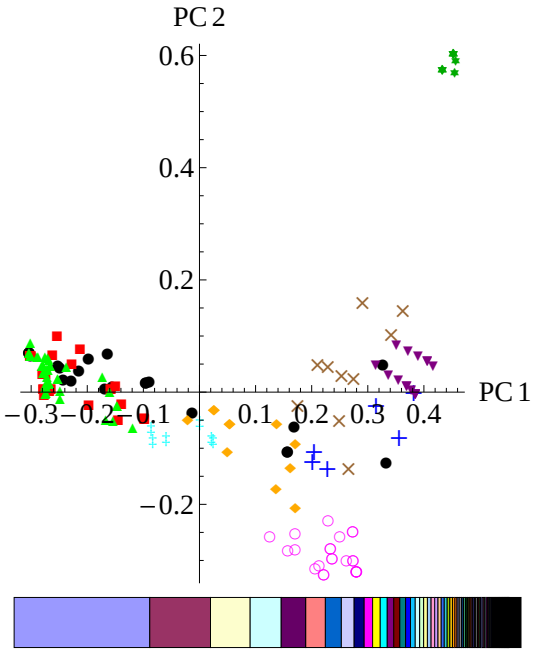


(b) Linear PCA, CATS2D descriptor. The first two PCs explain 46.6 % of the total variance; 14 PCs cover 90 % of it. Only fatty acids (∗, all projected into a single point) form a distinguishable cluster, while other classes form clusters, but are not distinguishable.

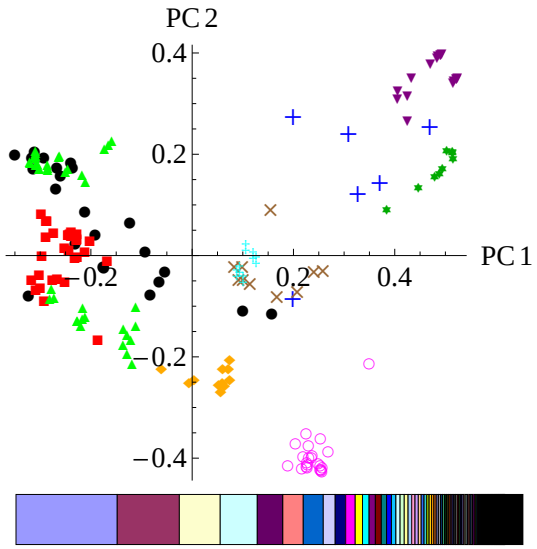


(c) Kernel PCA with Gaussian kernel (σ = 13) on MOE 2D descriptors. The first two PCs explain 34.6 % of the total variance; 39 PCs cover 90 % of it. Clustering behavior is marginally better than in (a). Parameter σ optimized using a grid search and the average receiver operating characteristic area under curve (p. 43) when ranking against each compound as performance criterion.

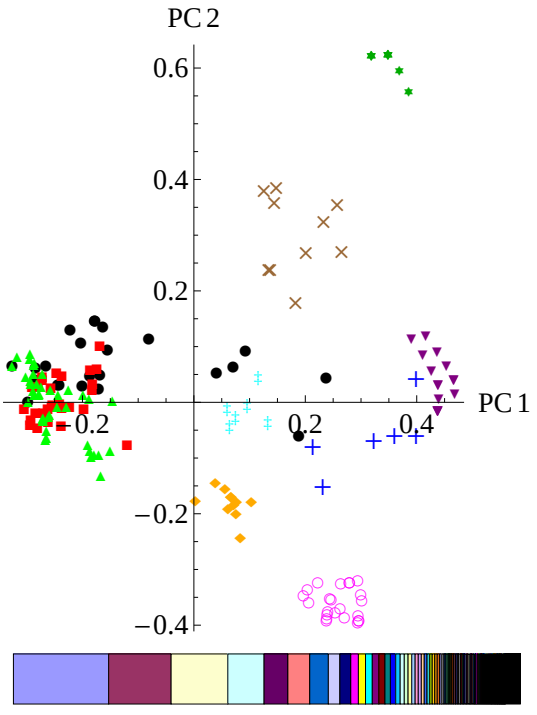
Figure 3.8 Principal component analysis visualization of the `ppar` data set. • = tyrosines A (1–23), ■ = tyrosines B (24–52, 176), ▲ = tyrosines C (53–94), + = thiazolidinediones (95–100), × = indoles (101–110), ○ = oxadiazoles (111-133), ∗ = fatty acids (134–142), ‡ = tertiary amides (143–148), ♦ = tyrosines N (149–159), ▼ = thiazolidinedione-fatty acid hybrids (160–175). Numbers in brackets refer to compound numbers in Rücker et al. (2006), whose classification is similar to, but different from the one in Scheme 4.7.



(a) Kernel PCA with ISOAK (no vertex kernel, no edge kernel). The first two PCs explain 38.7 % of the total variance; 37 PCs cover 90 % of it.



(b) Kernel PCA with ISOAK (Dirac kernel on element type as vertex annotation, Dirac kernel on bond type as edge annotation). The first two PCs explain 32.2 % of the total variance; 52 PCs cover 90 % of it.



(c) Kernel PCA with ISOAK (Dirac kernel on pharmacophore type as vertex annotation, no edge kernel). The first two PCs explain 31.1 % of the total variance; 48 PCs cover 90 % of it.

Figure 3.9 Kernel principal component analysis visualization of the `ppar` data set using the iterative graph similarity optimal assignment kernel. • = tyrosines A (1–23), ■ = tyrosines B (24–52, 176), ▲ = tyrosines C (53–94), + = thiazolidinediones (95–100), × = indoles (101–110), ○ = oxadiazoles (111-133), ∗ = fatty acids (134–142), ‡ = tertiary amides (143–148), ♦ = tyrosines N (149–159), ▼ = thiazolidinedione-fatty acid hybrids (160–175). Numbers in brackets refer to compound numbers in Rücker et al. (2006), whose classification is similar to, but different from the one in Scheme 4.7.

References

Mikhail Belkin, Partha Niyogi. *Laplacian eigenmaps for dimensionality reduction and data representation.* Neural Computation, 15(6): 1373–1396, 2002.

Yoshua Bengio, Olivier Delalleau, Nicolas Le Roux, Jean-François Paiement, Pascal Vincent, Marie Ouimet. *Learning eigenfunctions links spectral embedding and kernel PCA.* Neural Computation, 16(10): 2197–2219, 2004a.

Yoshua Bengio, Olivier Delalleau, Nicolas Le Roux, Jean-François Paiement, Pascal Vincent, Marie Ouimet. *Spectral dimensionality reduction.* In Isabelle Guyon, Steve Gunn, Masoud Nikravesh, Lotfi A. Zadeh (editors), *Feature Extraction. Foundations and Applications*, volume 207 of *Studies in Fuzziness and Soft Computing*, 519–550. Springer, 2006.

Yoshua Bengio, Jean-François Paiement, Pascal Vincent, Olivier Delalleau, Nicolas Le Roux, Marie Ouimet. *Out-of-sample extensions for LLE, Isomap, MDS, eigenmaps, and spectral clustering.* In Sebastian Thrun, Lawrence Saul, Bernhard Schölkopf (editors), *Advances in Neural Information Processing Systems 16 (NIPS 2003), Vancouver and Whistler, British Columbia, Canada, December 8–13*, 177–184. MIT Press, 2004b.

Ingwer Borg, Patrick Groenen. *Modern Multidimensional scaling.* Springer, New York, second edition, 2005.

Raymond Cattell. *The Scree test for the number of factors.* Multivariate Behavioral Research, 1(2): 245–276, 1966.

Timothy Chan. *All-pairs shortest paths with real weights in $O(n^3/\log n)$ time.* Algorithmica, 50(2): 236–243, 2008.

Tat-Jun Chin, David Suter. *Out-of-sample extrapolation of learned manifolds.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(9): 1547–1556, 2008.

Ronald Coifman, Stéphane Lafon. *Diffusion maps.* Applied and Computational Harmonic Analysis, 21(1): 5–30, 2006.

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. *Introduction to Algorithms.* MIT Press, Cambridge, second edition, 2001.

Trevor Cox, Michael Cox. *Multidimensional Scaling.* Chapman & Hall/CRC, Boca Raton, second edition, 2001.

Alexandre d'Aspremont, Laurent El Ghaoui, Michael Jordan, Gert Lanckriet. *A direct formulation for sparse PCA using semidefinite programming.* SIAM Review, 49(3): 434–448, 2007.

Fernando de la Torre, Michael Black. *A framework for robust subspace learning.* International Journal of Computer Vision, 54(1–3): 117–142, 2003.

Vin de Silva, Joshua Tenenbaum. *Global versus local methods in nonlinear dimensionality reduction.* In Suzanna Becker, Sebastian Thrun, Klaus Obermayer (editors), *Advances in Neural Information Processing Systems 15 (NIPS 2002), Cambridge, Massachusetts, USA, December 10–12*, 705–712. MIT Press, 2003.

Xinwei Deng, Ming Yuan, Agus Sudjianto. *A note on robust kernel principal component analysis.* In Joseph Verducci, Xiaotong Shen, John Lafferty (editors), *AMS-IMS-SIAM Joint Summer Research Conference on Machine and Statistical Learning: Prediction and Discovery, Snowbird, Utah, USA, June 25–29*, 21–33. American Mathematical Society, 2006.

Edsger Dijkstra. *A note on two problems in connexion with graphs.* Numerische Mathematik, 1(1): 269–271, 1959.

David Donoho, Carrie Grimes. *Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data.* Proceedings of the National Academy of Sciences of the United States of America, 100(10): 5591–5596, 2003.

Robert Floyd. *Algorithm 97 (shortest path).* Communications of the ACM, 5(6): 345, 1962.

Frank Gunstone. *Fatty Acid and Lipid Chemistry.* Springer, New York, 1996.

Louis Guttman. *Some necessary conditions for common-factor analysis.* Psychometrika, 19(2): 149–161, 1954.

Jihun Ham, Daniel Lee, Sebastian Mika, Bernhard Schölkopf. *A kernel view of the dimensionality reduction of manifolds.* In Carla Brodley (editor), *Proceedings of the 21st International Conference on Machine Learning (ICML 2004), Banff, Alberta, Canada, July 4–8*, 369–376. Omnipress, Madison, WI, USA, 2004.

Trevor Hastie, Werner Stuetzle. *Principal curves.* Journal of the American Statistical Association, 84(406): 502–516, 1989.

Heiko Hoffmann. *Kernel PCA for novelty detection.* Pattern Recognition, 40(3): 863–874, 2007.

Harold Hotelling. *Analysis of a complex of statistical variables into principal components.* Journal of Educational Psychology, 24(6): 417–441, 1933a.

Harold Hotelling. *Analysis of a complex of statistical variables into principal components.* Journal of Educational Psychology, 24(7): 498–520, 1933b.

David Hoyle. *Automatic PCA dimension selection for high dimensional data and small sample sizes.* Journal of Machine Learning Research, 9(12): 2733–2759, 2008.

Dimitar Hristozov, Tudor Oprea, Johann Gasteiger. *Ligand-based virtual screening by novelty detection with self-organizing maps.* Journal of Chemical Information and Modeling, 47(6): 2044–2062, 2007.

Su-Yun Huang, Yi-Ren Yeh, Shinto Eguchi. *Robust kernel principal component analysis.* Neural Computation, 21(11): 3179–3213, 2009.

Donald Jackson. *Stopping rules in principal components analysis: A comparison of heuristic and statistical approaches.* Ecology, 74(8): 2204–2214, 1993.

Ian Jolliffe. *Principal Component Analysis.* Springer, New York, second edition, 2004.

Henry Kaiser, Kern Dickman. *Analytic determination of common factors.* In *67th Annual Convention of the American Psychological Association, Cincinnati, Ohio, USA, September 3–9.* American Psychological Association, 1959.

Bärbel Lasitschka. *Spektrale Methoden zur Dimensionsreduktion von Bioaktivitätsdaten.* Master's thesis, Johann Wolfgang Goethe-University, Frankfurt am Main, Germany, 2009.

Pavel Laskov, Christin Schäfer, Igor Kotenko. *Intrusion detection in unlabeled data with quarter-sphere support vector machines.* In Ulrich Flegel, Michael Meier (editors), *Proceedings of the GI SIG SIDAR Workshop on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA 2004), Dortmund, Germany, July 6–7*, volume 46 of *Lecture Notes in Informatics*, 71–82. Gesellschaft für Informatik, 2004.

John Lee, Michel Verleysen. *Nonlinear Dimensionality Reduction.* Springer, New York, 2007.

Carl Meyer. *Matrix Analysis and Applied Linear Algebra.* Society for Industrial and Applied Mathematics, Philadelphia, 2001.

Gerry Moss, Peter Smith, Dirk Tavernier. *Glossary of class names of organic compounds and reactivity intermediates based on structure (IUPAC recommendations 1995).* Pure and Applied Chemistry, 67(8): 1307–1375, 1995.

Minh Hoai Nguyen, Fernando De la Torre. *Robust kernel principal component analysis.* In Daphne Koller, Dale Schuurmans, Yoshua Bengio, Léon Bottou (editors), *Advances in Neural Information Processing Systems 21 (NIPS 2008), Vancouver, Canada, December 8–11*, 1185–1192. MIT Press, 2008.

Karl Pearson. *On lines and planes of closest fit to systems of points in space.* Philosophical Magazine, 2(6): 559–572, 1901.

Seth Pettie, Vijaya Ramachandran. *A shortest path algorithm for real-weighted undirected graphs.* SIAM Journal on Computing, 34(6): 1398–1431, 2005.

John Rice. *Mathematical Statistics and Data Analysis.* Duxbury Press, Belmont, third edition, 2006.

Sam Roweis, Lawrence Saul. *Nonlinear dimensionality reduction by locally linear embedding.* Science, 290(5500): 2323–2326, 2000.

Christoph Rücker, Marco Scarsi, Markus Meringer. *2D QSAR of PPARγ agonist binding and transactivation.* Bioorganic & Medicinal Chemistry, 14(15): 5178–5195, 2006.

Lawrence Saul, Kilian Weinberger, Fei Sha, Jihun Ham, Daniel Lee. *Spectral methods for dimensionality reduction.* In Olivier Chapelle, Bernhard Schölkopf, Alexander Zien (editors), *Semi-Supervised Learning*, 293–308. MIT Press, Cambridge, 2006.

Bernhard Schölkopf, John Platt, John Shawe-Taylor, Alexander Smola, Robert Williamson. *Estimating the support of a high-dimensional distribution.* Neural Computation, 13(7): 1443–1471, 2001.

Bernhard Schölkopf, Alexander Smola. *Learning with Kernels.* MIT Press, Cambridge, 2002.

Bernhard Schölkopf, Alexander Smola, Klaus-Robert Müller. *Nonlinear component analysis as a kernel eigenvalue problem.* Neural Computation, 10(5): 1299–1319, 1998.

Bernhard Schölkopf, Alexander Smola, Klaus-Robert Müller. *Kernel principal component analysis.* In Bernhard Schölkopf, Christopher Burges, Alexander Smola (editors), *Advances in Kernel Methods — Support Vector Learning*, 327–352. MIT Press, Cambridge, 1999.

Anton Schwaighofer, Volker Tresp, Kai Yu. *Learning Gaussian process kernels via hierarchical Bayes.* In Lawrence Saul, Yair Weiss, Léon Bottou (editors), *Advances in Neural Information Processing Systems 17 (NIPS 2004), Vancouver and Whistler, British Columbia, Canada, December 13–18*, 1209–1216. MIT Press, 2004.

Blake Shaw, Tony Jebara. *Minimum volume embedding.* In Marina Meila, Xiaotong Shen (editors), *Proceedings of the 11th International Workshop on Artificial Intelligence and Statistics (AISTATS 2007), San Juan, Puerto Rico, March 19–22*, 2007.

Michael Sipser. *Introduction to the Theory of Computation.* Thomson, second edition, 2005.

Gilbert Stewart, Ji-Guang Sun. *Matrix Perturbation Theory.* Academic Press, New York, 1990.

David Tax, Klaus-Robert Müller. *A consistency-based model selection for one-class classification.* In *Proceedings of the 17th International Conference on Pattern Recognition (ICPR 2004), Cambridge, England, August 23–26*, volume 3, 363–366. IEEE Computer Society, 2004.

Josh Tenenbaum, Vin de Silva, John Langford. *A global geometric framework for nonlinear dimensionality reduction.* Science, 290(5500): 2319–2323, 2000.

Michael Tipping. *Sparse kernel principal component analysis.* In Todd Leen, Thomas Dietterich, Volker Tresp (editors), *Advances in Neural Information Processing Systems 13 (NIPS 2000), Denver, Colorado, USA, November 27–December 2.* MIT Press, 2001.

Ulrike von Luxburg. *A tutorial on spectral clustering.* Statistics and Computing, 17(4): 395–416, 2007.

Jing Wang. *Improve local tangent space alignment using various dimensional local coordinates.* Neurocomputing, 71(16–18): 3575–3581, 2008.

Kilian Weinberger, Benjamin Packer, Lawrence Saul. *Nonlinear dimensionality reduction by semidefinite programming and kernel matrix factorization.* In Zoubin Ghahramani, Robert Cowell (editors), *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics (AISTATS 2005), Barbados, January 6–8*, 2005.

Kilian Weinberger, Lawrence Saul. *Unsupervised learning of image manifolds by semidefinite programming.* International Journal of Computer Vision, 70(1): 77–90, 2006.

Kilian Weinberger, Fei Sha, Lawrence Saul. *Learning a kernel matrix for nonlinear dimensionality reduction.* In Carla Brodley (editor), *Proceedings of the 21st International Conference on Machine Learning (ICML 2004), Banff, Alberta, Canada, July 4–8*, 106. Omnipress, Madison, WI, USA, 2004.

David Weininger. *SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules.* Journal of Chemical Information and Computer Sciences, 28(1): 31–36, 1988.

Daniela Witten, Rob Tibshirani, Trevor Hastie. *A penalized matrix decomposition, with applications to sparse canonical correlation analysis and principal components.* Biostatistics, 10(3): 515–534, 2009.

Ian Witten, Eibe Frank. *Data Mining. Practical Machine Learning Tools and Techniques.* Elsevier, 2005.

Svante Wold, Michael Sjöström, Lennart Eriksson. *PLS-regression: A basic tool of chemometrics.* Chemometrics and Intelligent Laboratory Systems, 58(2): 109–130, 2001.

World Health Organization. *Guidelines on the use of international nonproprietary names (INNs) for pharmaceutical substances*, 1997. `http://www.who.int/medicines/services/inn`.

Zhenyue Zhang, Hongyuan Zha. *Principal manifolds and nonlinear dimensionality reduction via tangent space alignment.* SIAM Journal on Scientific Computing, 26(1): 331–338, 2004.

Hui Zou, Trevor Hastie, Robert Tibshirani. *Sparse principal component analysis.* Journal of Computational and Graphical Statistics, 15(2): 265–286, 2006.

It is not the estimate or the forecast that matters so much as the degree of confidence with the opinion.

Nassim Taleb (2005)

Chapter 4

Peroxisome proliferator-activated receptor

The peroxisome proliferator-activated receptor is a nuclear transcription factor involved in the regulation of lipid and glucose metabolism that plays a crucial role in the development of diseases like type 2 diabetes and dyslipidemia. We carry out a prospective virtual screening study for novel agonists of subtype γ of this receptor, based on Gaussian process regression using molecular descriptors and the iterative similarity optimal assignment graph kernel developed in Chapter 2. The most potent selective hit (EC_{50} $= 10 \pm 0.2\,\mu$M) is a derivative of truxillic acid, a substance that occurs naturally in plant cell walls. Our study delivered a novel agonist, de-orphanized a natural bioactive product, and, hints at the natural product origins of pharmacophore patterns in synthetic ligands.

4.1 Target

We describe the *peroxisome proliferator-activated receptor* (PPAR) as a drug target, with particular emphasis on the PPARγ subtype (Table 4.1). For further information on PPARs, see Willson et al. (2000); Mudaliar and Henry (2002); Michalik et al. (2006); for PPARγ in particular, see Rosen and Spiegelman (2001); Henke (2004).

4.1.1 Overview

PPAR research started when Issemann and Green (1990) cloned the first member of this receptor group.[1] Since then, PPARs and their ligands have been intensively investigated,[2] establishing the receptor as a validated drug target (Rau et al., 2006).

[1] The name PPAR is a misnomer, based upon the early identification of the receptor as a target for substances that cause a proliferation of liver peroxisomes in rodents. See p. 144 for regulatory functions.

[2] As of 2009-04-03, a search for publications using the keyword *PPAR* yielded 9 869 hits in PubMed (`www.ncbi.nlm.nih.gov/pubmed`), and 13 151 hits in Web of Science (`www.isiknowledge.com`).

Table 4.1 PPARγ summary, based on Michalik et al. (2006). HRE = hormone response element, Hs = *Homo sapiens*, Mm = *Mus musculus*, Rn = *Rattus norvegicus*, NR1C3 = nuclear receptor subfamily 1, group C, member 3 (see Table 4.2).

Property	Description
Receptor	
trivial name	Peroxisome proliferator-activated receptor γ
abbreviation	PPARγ
nomenclature	NR1C3
DNA binding	
structure	Hetero-dimer, RXR partner (physical, functional)
HRE core sequ.	AACTAGGNCA A AGGTCA (DR-1)
activated genes	FATP, acyl CoA-synthetase, aP2 adipocyte lipid-binding protein, Lpl, UCP-1, PEPCK, Apoa2 (all Mm)
co-repressors	NRIP1, SAF-B, TAZ, NCOR1, NCOR2, SMRT
co-activators	PGC-2, ARA-70, PGC-1α, PPARGC1B, CREBBP, p300, CITED2, ERAP140, PPARBP, PRMT-2, PIMT, NCOA1, NCOA2, NCOA3, NCOA6, SWI/SNF, PDIP
Tissue distribution	Adipose tissues, lymphoid tissues, colon, liver, heart (Hs, Mm, Rn)
Important isoforms	• *PPARγ1 (Hs, Mm):* encoded by 8 exons (2 of them γ1-specific) • *PPARγ2 (Hs, Mm, Rn):* 28 additional N-terminal amino acids • *PPARγ3 (Hs):* protein indistinguishable from PPARγ1, different promoter, only expressed in colon and adipose tissue
Human disease	• *Obesity, insulin resistance:* associated with a mutation in the ligand-independent activation domain of PPARγ2 • *Insulin resistance, type II diabetes mellitus, hypertension:* associated with a mutation of the ligand binding domain • *Syndrome X, metabolic syndrome:* associated with dominant-negative PPARγ mutations • *Atherosclerosis:* increased receptor expression in atherosclerotic lesions, macrophages, monocytic cell lines • *Colon cancer:* associated with loss-of-function mutations in PPARγ ligand binding domain • *Prostate cancer:* PPARγ expressed in human prostate adenocarcinomas and cell lines derived from human prostate tumors • *Thyroid tumors:* the PAX8-PPARγ fusion protein promotes differentiated follicular thyroid neoplasia

Classification

PPARs belong to the frequently targeted[3] superfamily of nuclear receptors (Table 4.2; Ottow and Weinmann, 2008), which act as transcription factors regulating gene expression. There are 3 PPAR subtypes, PPARα (NR1C1), PPARβ/δ (NR1C2),[4] and PPARγ (NR1C3). Each subtype is the product of a distinct gene. The human gene for PPARα is located on chromosome locus 22q12–q13.1 (Sher et al., 1993), PPARβ/δ on 6p21.1–p21.2 (Yoshikawa et al., 1996), and PPARγ on 3p25 (Beamer et al., 1997). Of PPARγ, three messenger RNA isoforms are known in humans, PPARγ1, PPARγ2, and PPARγ3, arising through different promoter usage and alternative splicing (Fajas et al., 1997). PPARγ1 and PPARγ3 encode the same protein, while the PPARγ2 protein has 28 additional amino acids at the N-terminus.

Mechanism of action

PPARs form hetero-dimers with the retinoid X receptor (RXR; NR2B)[5] and bind to PPAR-specific response elements (PPRE) within the promoter regions of their target genes. In their inactive (ligand-free) state, these hetero-dimers form complexes with nuclear receptor co-repressors, which prevent DNA transcription, e. g., via histone deacetylation (Mudaliar and Henry, 2002). After binding by an agonist, the induced conformational change into the active form of the protein causes the co-repressor complexes to dissociate and allows the recruitment of co-activators; the resulting complexes acetylize histones near the promoter, leading to gene transcription (Figure 4.1). This activation depends on the agonist to stabilize the active conformation[6] of the PPAR (Figure 4.3). Transcriptional activity is further modulated via phosphorylation of PPAR by various kinases (Burns and Vanden Heuvel, 2007). See Gampe et al. (2000) for structural details of hetero-dimerization and agonist binding.

PPREs have mainly been found in the control regions of genes related to lipid metabolism and transport (Duval et al., 2007), e. g., acyl-CoA synthetase (Schoonjans et al., 1995), and lipoprotein lipase (Schoonjans et al., 1996), as well as genes related to glucose neogenesis and metabolism, e. g., phosphoenolpyruvate carboxykinase (Tontonoz et al., 1995) and pyruvate dehydrogenase kinase (Degenhardt et al., 2007). See Heinäniemi et al. (2007) for an analysis and prediction of PPREs in the human genome.

From the three PPAR subtypes, PPARγ binds PPREs with the highest affinity, while PPARα and PPARβ/δ show similar but lower affinities. Juge-Aubry et al. (1997) propose to classify PPREs into strong (all 3 subtypes bind with equal affinity), intermediate (PPARγ binds with twice the affinity of PPARα and PPARβ/δ), and weak (only bound by PPARγ) elements. PPREs can also influence the preferred RXR subtype for dimerization.

[3] 13 % of the drugs approved by the United States food and drug administration are targeted at nuclear receptors (Overington et al., 2006).

[4] PPARβ was first discovered in *Xenopus laevis* by Dreyer et al. (1992). When subsequently identified in mice (Kliewer et al., 1994), rats, and humans, it was called PPARδ. Only later were the two recognized as orthologues.

[5] In contrast to other RXR hetero-dimers, only the ligand binding domain contributes to the dimerization, but not the DNA binding domain. This is due to the PPAR D-box having only 3 instead of the usual 5 amino acids between the cysteins of the first pair (Hsu et al., 1998). Note that PPAR/RXR hetero-dimers can be activated by both PPAR and RXR ligands (Kliewer et al., 1992).

[6] See Gani and Sylte (2008) for a molecular dynamics study of PPARγ stabilization and activation by two glitazones and docosahexenoic acid.

Table 4.2 Human nuclear receptors, based on Gronemeyer et al. (2004). For details on nuclear receptor nomenclature, see Nuclear Receptors Nomenclature Committee (1999); Germain et al. (2006). F = subfamily, G = group, # = member. The code for a subfamily x, group y, member z receptor is NRxyz, e. g., NR1C3 for PPARγ. COUP-TF = chicken ovalbumin upstream promoter-transcription factor, NGF = nerve growth factor, DSS-AHC = dosage-sensitive sex reversal-adrenal hypoplasia congenita.

F	G	#	Trivial name	Abbreviation
1	A	1,2	Thyroid hormone receptor	TRα, TRβ
	B	1,2,3	Retinoic acid receptor	RARα, RARβ, RARγ
	C	1,2,3	Peroxisome proliferator-activated receptor	PPARα, PPARβ/δ, PPARγ
	D	1	Reverse erbA	Rev-erbα, Rev-erbβ
	F	1,2,3	RAR-related orphan receptor	RORα, RORβ, RORγ
	H	3,2	Liver X receptor	LXRα, LXRβ
		4,5	Farnesoid X receptor	FXRα, FXRβ[a]
	I	1	Vitamin D receptor	VDR
		2	Pregnane X receptor	PXR
		3	Constitutive androstane receptor	CAR
2	A	1,2	Human nuclear factor 4	HNF4α, HNF4γ
	B	1,2,3	Retinoid X receptor	RXRα, RXRβ, RXRγ
	C	1,2	Testis receptor	TR2, TR4
	E	2	Tailless	TLL
		3	Photoreceptor-specific nuclear receptor	PNR
	F	1,2	COUP-TF	COUP-TFI, COUP-TFII
		6	ErbA2-related gene-2	EAR2
3	A	1,2	Oestrogen receptor	ERα, ERβ
	B	1,2,3	Oestrogen receptor-related receptor	ERRα, ERRβ, ERRγ
	C	1	Glucocorticoid receptor	GR
		2	Mineralocorticoid receptor	MR
		3	Progesterone receptor	PR
		4	Androgen receptor	AR
4	A	1	NGF-induced factor B	NGFIB
		2	Nur related factor 1	NURR1
		3	Neuron-derived orphan receptor 1	NOR1
5	A	1	Steroidogenic factor 1	SF1
		2	Liver receptor homologous protein 1	LRH1
6	A	1	Germ cell nuclear factor	GCNF
	B	1	DSS-AHC critical region chrom. gene 1[b]	DAX1
		2	Short hetero-dimeric partner	SHP

[a] FXRβ is a pseudo-gene in humans.
[b] DSS-AHC critical region on chromosome, gene 1.

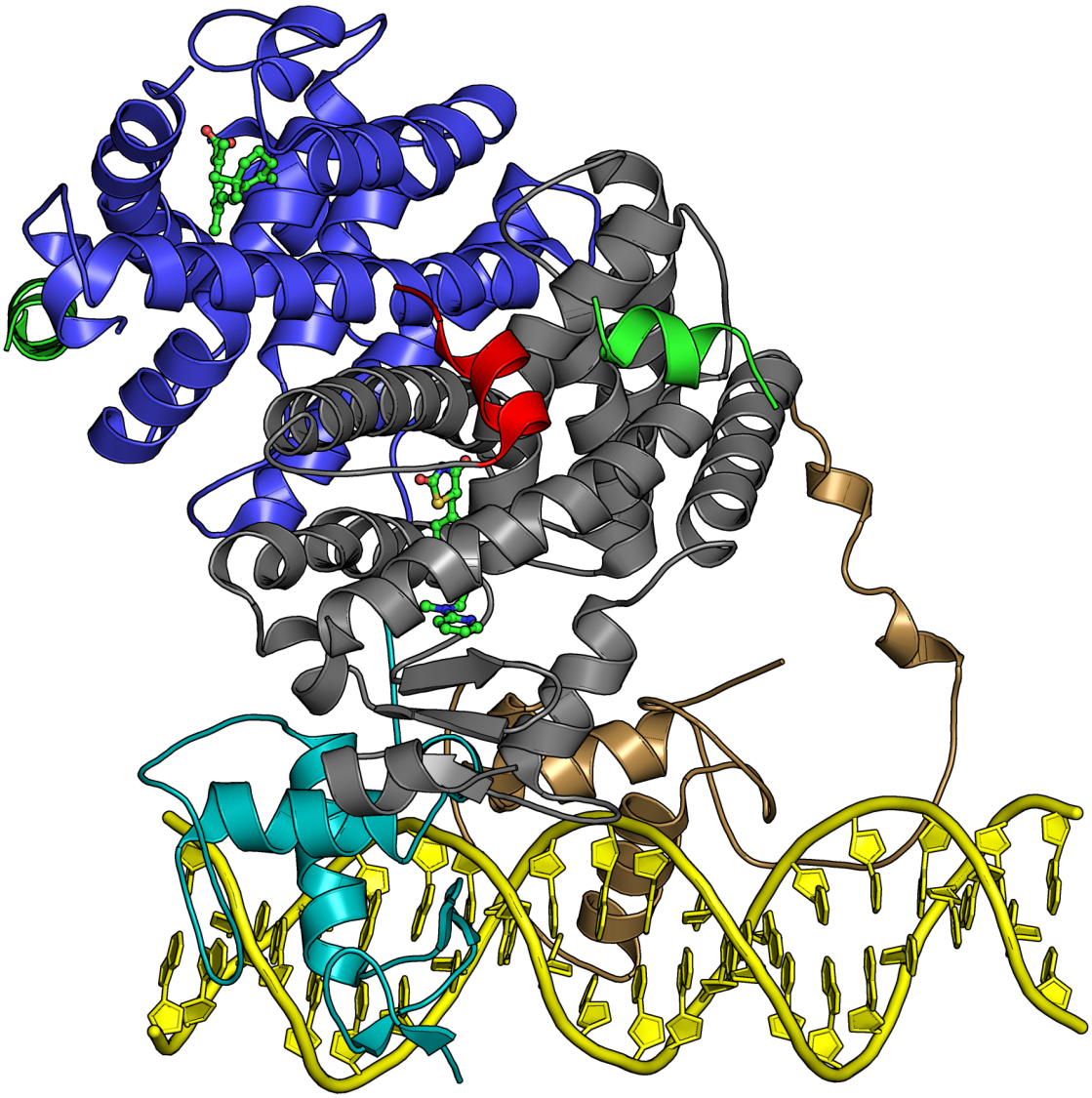


Figure 4.1 PPARγ-RXRα hetero-dimer in complex with DNA (PDBid 3dzy; Chandra et al., 2008). Shown are PPARγ (LBD in grey, DBD in brown, AF-2 (helix H12) in red) with the ligand rosiglitazone (ball and stick model) and RXRα (LBD in blue, DBD in cyan) with the ligand 9-cis-retinoic acid (ball and stick model), co-activator peptides (green), and DNA (yellow). PPAR = peroxisome proliferator-activated receptor, RXR = retinoid X receptor, PDBid = protein data bank identifier, DNA = deoxyribonucleic acid, LBD = ligand binding domain, DBD = DNA binding domain, AF-2 = activation function 2.

AF-1	transcriptional activation function 1
DBD	DNA-binding domain (~70 amino acids)
H	hinge region (highly variable)
LBD	ligand-binding domain (~250 amino acids)
AF-2	transcriptional activation function 2

Figure 4.2 Domain structure of human PPAR (Henke, 2004). The DBD is strongly conserved (e. g., 83 % sequence identity between PPARα and PPARγ1/PPARγ2), whereas the LBD is less conserved (e. g., 68 % sequence identity between PPARα and PPARγ1/PPARγ2).

Structure

Human PPARs consist of about 450 amino acids,[7] with significant variation in the residues of the ligand binding pocket (Willson et al., 2000). The structures of all human PPAR subtypes have been determined by X-ray crystallography,[8] and are similar to those of other nuclear receptors (Bourguet et al., 2000); they share the general domain structure (Figure 4.2) of steroid, retinoid, and thyroid hormone receptors. The first N-terminal domain is the poorly conserved ligand-independent transactivation function 1. It is followed by the strongly conserved DNA binding domain, consisting of two zinc finger motifs, a highly variable hinge region, and the ligand binding domain with the ligand-dependent transactivation function 2. The ligand binding domain consists of 13 α-helices and a four-stranded β-sheet (Figure 4.3a).

Binding pocket

The PPAR binding pockets (Figures 4.3b, 4.4) are large[9] compared to other nuclear receptors, mostly hydrophobic, and deeply buried, with PPARα having the largest and most hydrophobic binding pocket, followed by PPARγ and PPARβ/δ. Many ligands occupy only a small portion (~20 %) of the pockets (Gronemeyer et al., 2004), enabling PPAR to accommodate a variety of ligands.

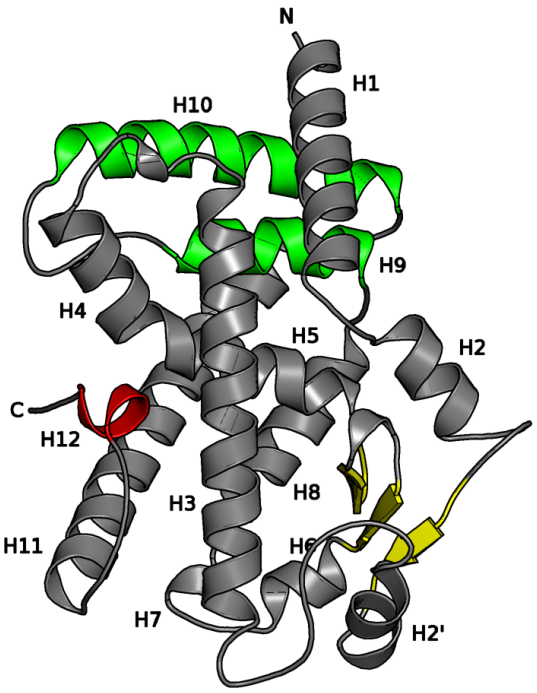
The binding pockets are defined by 35 residues, about 80 % of which are conserved across all PPAR subtypes. They consist of three parts, a polar part (left proximal pocket, arm I) including the AF-2 domain, a hydrophobic part (left distal pocket, arm II), and a structurally conserved part (right distal pocket, arm III) that has hydrophobic and hydrophilic residues (Pirard, 2003; Markt et al., 2007). The carboxyl group of endogenous fatty acid ligands interacts with the four residues Ser289, His323, His449, Tyr473 in the PPARγ[10] pocket (Figure 4.4), stabilizing the AF-2 helix (Zoete et al., 2007); their hydrophobic tail is buried in the left or right distal pocket.

[7] *h*PPARα 468 amino acids, *h*PPARβ/δ 441 amino acids, *h*PPARγ1 477 amino acids (Rau, 2007).

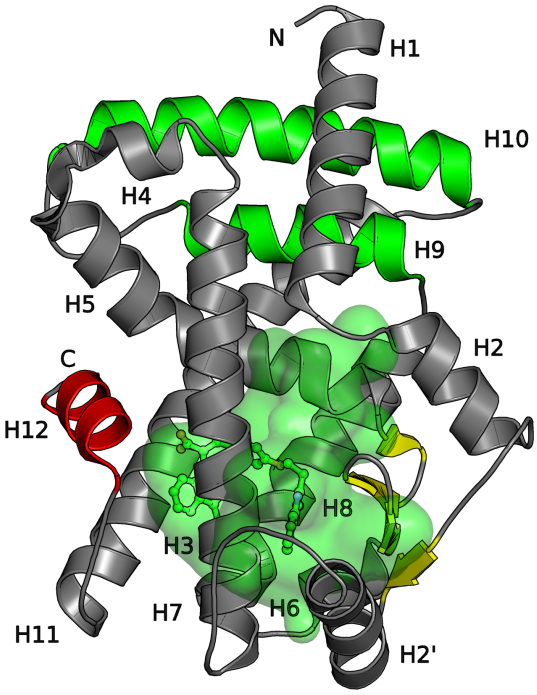
[8] Structures are accessible via the protein data bank (PDB; Berman et al., 2000); see Zoete et al. (2007) for a list of identifiers.

[9] For PPARγ, binding pocket volumes have been given as 1.4 nm^3 (1 400 Å^3; Zoete et al., 2007) and 1.6 nm^3 (1 600 Å^3; Gronemeyer et al., 2004). Other nuclear receptors typically have binding pocket volumes between 0.6 nm^3 (600 Å^3) and 1.1 nm^3 (1 100 Å^3; Itoh et al., 2008).

[10] Ser280, Tyr314, His440, Tyr464 in PPARα, Thr289, His323, His449, Tyr473 in PPARβ/δ.



(a) Ligand-free (apo) conformation (PDBid 3prg; Nolte et al., 1998) of the PPARγ LBD. Helices in gray, β-sheets in yellow, AF-2 (H12) in red, dimerization interface in green.



(b) Ligand-bound (holo) form with the agonist farglitazar (PDBid 1fm9; Gampe et al., 2000) in the binding pocket. Coloring as in (a). The binding pocket surface, calculated by PocketPicker (Weisel et al., 2007), is shown in transparent green. The ligand fills only a small part of the large pocket.

Figure 4.3 Structure of PPARγ in ligand-free (apo) and ligand-bound (holo) form. The subtle change in the conformation of helix 12 from (a) to (b) leads to a stabilized charge clamp for co-activator recruitment (Gampe et al., 2000). PDBid = protein data bank identifier, PPAR = peroxisome proliferator-activated receptor, LBD = ligand binding domain, AF-2 = activation function 2 (helix 12), H = α-helix, C = carboxyl-terminal end, N = amine-terminal end.

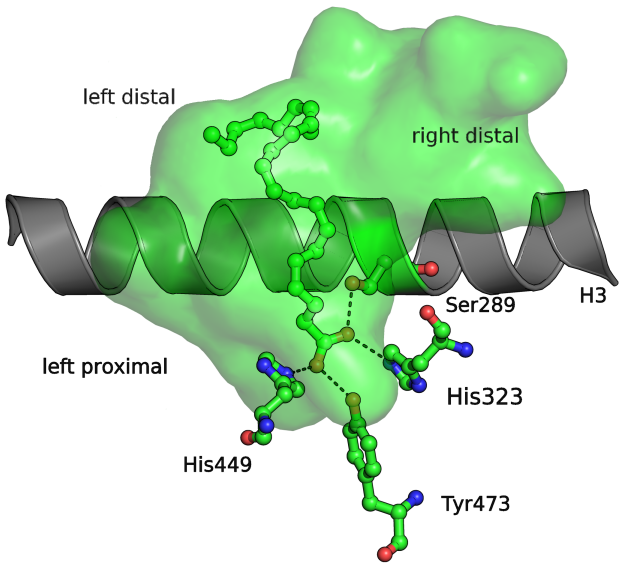


Figure 4.4 Binding pocket of PPARγ with the endogenous agonist docosahexaenoic acid (PDBid 2vv0; Itoh et al., 2008). Helix 3 is shown in gray, the binding pocket surface, calculated by PocketPicker (Weisel et al., 2007), in transparent green; dashed lines indicate amino acid interactions of the ligands acidic headgroup. PDBid = protein data bank identifier, PPAR = peroxisome proliferator-activated receptor, H = α-helix, His = histidine, Ser = serine, Tyr = tyrosine.

4.1.2 Relevance

PPARs play essential roles in the regulation of lipid and glucose metabolism, as well as cellular differentiation and development. They are related to a wide variety of diseases, and consequently became established and successful[11] pharmaceutical targets. The subtypes of PPAR are distinct in expression, physiological role, and involved diseases.

Expression in humans

PPARα is highly expressed in cells that have active fatty acid oxidation capacity, i. e., high mitochondrial and peroxisomal β-oxidation activity, including hepatocytes, cardiomyocytes, enterocytes, and renal proximal tubule cells (Desvergne and Wahli, 1999).

PPARβ/δ is expressed ubiquitously, often at higher levels than PPARα and PPARγ. Expression levels vary across tissues, with higher expression in placenta and large intestine (Desvergne and Wahli, 1999), as well as lipid-metabolizing tissues such as adipose tissue, small intestine, skeletal muscles, and cardiac muscle (Grimaldi, 2007). It has high expression rates in embryonic tissues, where it precedes expression of PPARα and PPARγ (Michalik et al., 2002).

PPARγ is expressed predominantly in adipose tissue and cells of the immune system. Of its isoforms, PPARγ1 has the broadest tissue expression, including adipose tissue, liver, and heart. PPARγ2 is expressed mostly in adipose tissue, and PPARγ3 was found in adipose tissue, macrophages, and colon epithelium. Additionally, PPARγ1 and PPARγ2 were also found at lower levels in skeletal muscle. In culture, PPARγ1 is expressed in B lymphocytes, myeolid cell lines, and primary bone marrow stromal cells (Desvergne and Wahli, 1999).

Physiological role

The primary function of PPARs in adult tissues is the regulation of lipid and glucose homeostasis by inducing key enzymes of the primary metabolism of fatty acids and glucose, mainly through expression of PPARα in the liver and PPARγ in adipose tissue. PPARs also play a role in inflammatory processes (Szanto and Nagy, 2008).

The main physiological role of PPARα is to react to elevated blood levels of fatty acids by initiating counter-regulatory signals. In particular, it increases the expression of apolipoproteins and lipoprotein lipases, which leads to an increased uptake of very low density lipoproteins by liver cells, and up-regulates fatty acid binding protein, acyl-CoA-oxidase and acyl-CoA-synthase, which are key enzymes for the intra-cellular binding and metabolization of free fatty acids. PPARα also represses the hepatic inflammatory response by down-regulating the expression of numerous pro-inflammatory genes, such as various acute-phase proteins (Gervois et al., 2004).

PPARβ/δ has functional roles in angiogenesis, skin homeostasis, and, in wound healing by governing keratinocyte differentiation (Burdick et al., 2006; Chong et al., 2009). It stimulates fatty acid oxidation and glucose uptake in adipose tissue, as well as in skeletal and cardiac muscle (Krämer et al., 2007); it also regulates hepatic production and catabolism of very low density lipoprotein. It can inhibit the other two PPAR subtypes.

PPARγ regulates adipogenesis depending on systemic lipid metabolism (Tontonoz et al., 1994). It increases lipid uptake from blood, similar to PPARα, as well as glucose uptake by induction of transporters such as glucose-transmembrane-transporters GLUT-2 and GLUT-4 (Im et al., 2005; Wu et al., 1998). Through this mechanism, activation of PPARγ leads to adipocyte growth and hyperplasia of adipose tissue (Miyazaki et al.,

[11] The 2005 global sales for the PPARγ agonists rosiglitazone and pioglitazone exceeded $5 \cdot 10^9$ US$.

Table 4.3 Human processes and diseases involving PPARs. Preliminary evidence exists for a role of PPARγ in amyotrophic lateral sclerosis (Kiaei, 2008), Huntington's disease (Quintanilla et al., 2008), and Parkinson's disease (Chaturvedi and Beal, 2008). See Willson et al. (2001) for a review of PPARγ in metabolic disease.

Disease / process	References
Atherosclerosis	Rosen and Spiegelman (2000)
Type 2 diabetes mellitus	Patsouris et al. (2004)
Dyslipidemia	Duval et al. (2007); Rau et al. (2008)
Obesity	Zhang et al. (2004)
Immune system	Schlezinger et al. (2004)
Inflammation	Delerive et al. (2001); Welch et al. (2003)
Inflammatory bowel diseases	Dubuquoy et al. (2006)
Multiple sclerosis	Drew et al. (2008)
Cancer	Tien et al. (2003)
Aging	Cheng and Mukherjee (2005); Fernandez (2004)
Cellular proliferation	Cuzzocrea et al. (2004)
Fertility	Komar (2005); Corton and Lapinskas (2005)
Alzheimer's disease	Kummer and Heneka (2008)
Ischemic stroke	Culman et al. (2007)

2002). PPARγ also regulates adipocyte mediators like tumor necrosis factor α, leptin, and plasminogen activator inhibitor-1.

Diseases

PPARs have been linked to several diseases associated with altered levels of fatty acids, most notably obesity, type 2 diabetes mellitus, atherosclerosis, and hypertension (Table 4.3). A connection to cancer has recently been proposed, based on the reasoning that carcinogenesis is the result of aberrant cell differentiation, in whose control PPARγ is involved. Although it seems to play no direct role in the causation of human carcinomas, PPARγ mutations and loss of expression have been associated with colon cancer, thyroid follicular carcinomas, and breast cancer (Sporn et al., 2001).

4.1.3 Ligands

Most PPAR ligands consist of an acidic headgroup, an aromatic core, a hydrophobic tail, and connecting linkers (Figure 4.5), mimicking in part the endogenous fatty acid ligands. Nuclear receptor ligands are more lipophilic and membrane-permeable compared to ligands of other receptors. PPARs accommodate a large variety of natural and synthetic compounds, many of which fill only a small part (~20 %) of the large binding pocket (~1.5nm^3, p. 142). For further information on PPARγ ligands, see Henke (2004).

Modulation

A nuclear receptor *ligand* is a compound that binds to the receptors C-terminal ligand binding domain; it is *selective* if there is a large enough (>100-fold) difference in affinity to other nuclear receptors. *Agonists* induce the active conformation of the receptor, with *full agonists* causing maximal activation and *partial agonists* producing weaker effects.

Antagonists induce conformations and receptor actions different from agonists, and can thereby oppose the latter. *Inverse agonists* are ligands that stabilize the inactive form of the receptor, preventing constitutive activity.

PPAR agonists induce and stabilize an AF-2 (helix 12) conformation (Figure 4.3b) that leads to the recruitment of co-activators and gene transcription. Partial PPAR agonists are unable to form some of the hydrogen bonds that full agonists employ (Figure 4.4), using hydrophobic interactions instead. This leads to the recruitment of different co-activators, and reduced or altered gene transcription (Markt et al., 2007). PPARs are thought to be constitutively active depending on the ratio of co-activators and co-repressors (Tudor et al., 2007).

The concepts of agonist and antagonist have been criticized as being overly simplistic with respect to nuclear receptors, as a ligands action may depend on the cellular context, such as the presence of co-repressants and co-activators (Sporn et al., 2001). The terms *modulator* and *selective nuclear receptor modulator* (SNuRM) have been suggested for ligands that induce tissue-selective agonist or antagonist activity.

Selectivity

Due to the size and hydrophobicity of their binding pockets (p. 142), PPARα and PPARγ seem more suited for fatty acids and hydroxylated fatty acids, respectively (Markt et al., 2007). Because of its smaller size, the PPARβ/δ binding pocket can not accommodate large hydrophobic tail groups. Subtype selectivity is often achieved via differences in the left proximal and distal subpockets (arms I and II).

Adverse effects

Currently marketed PPAR agonists display serious safety issues. Adverse reactions of specific PPARγ and dual PPARα/γ agonists include potential carcinogenicity in rodents, myopathy, rhabdomyolysis, increase in plasma creatinine and homocysteine, hemodilution, decrease in glomerular filtration rate, weight gain, fluid retention, peripheral edema, and congestive cardiac failure (Rubenstrunk et al., 2007). The toxicology of PPAR ligands is reviewed by Peraza et al. (2006).

Endogenous ligands

Native PPAR ligands (Scheme 4.1) are lipid metabolites such as oxidized fatty acids (Itoh et al., 2008), phospholipids, and, arachidonic acid derivatives, such as prostaglandins and prostacyclins (Nettles, 2008), most of which activate the receptor in the micromolar range. The physiological concentrations of these native ligands are often lower than those necessary for PPAR activation (up to three orders of magnitude, Schopfer et al., 2005). However, its versatile and large binding pocket allows PPAR to bind many ligands, making it a sensor not of a single ligand, but of a pool of ligands. Thus, it is the physiological concentration of the pool, e. g., total fatty acid levels, that is relevant (Itoh et al., 2008).

High affinity endogenous ligands are unsaturated nitrated fatty acid derivatives that activate PPARs in the nanomolar range (Baker et al., 2005). PPARγ in particular is activated by nitrolinoleic acid (Compound 12), docosahexaenoic acid (Figure 4.4; Zapata-Gonzalez et al., 2008), 5-oxo-eicosatetraenoic acid, 15-oxo-eicosatetraenoic acid, 15-oxo-eicosadecaenoic acid, 15-keto-prostaglandin $F_{2\alpha}$, and 15-keto-prostaglandin $F_{1\alpha}$ (Shiraki et al., 2005).

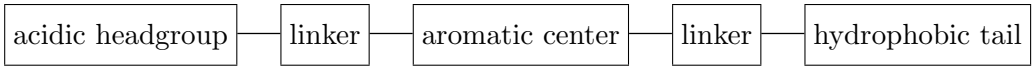
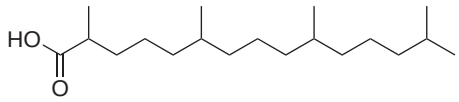
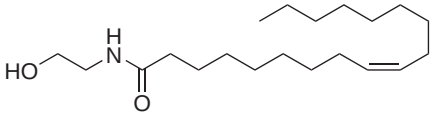


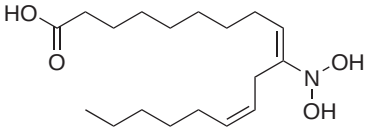
Figure 4.5 Simplified topology of many synthetic PPAR agonists (based on Kuhn et al., 2006). Linkers can be branched to access additional parts of the binding pocket.



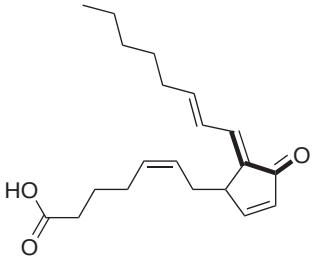
10 Pristanic acid (2,6,10,14-tetramethylpentadecanoic acid), a fatty acid derived (via external sources) from the phytol side chain of chlorophyll, with phytanic acid as intermediate (PPARα EC_{50} = 40 μM; Zomer et al., 2000).



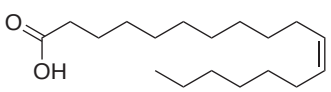
11 Oleoylethanolamide, a fatty acid ethanolamide that regulates satiety and body weight (PPARα EC_{50} = 0.12 μM, PPARβ/δ EC_{50} = 1.1 μM, inactive on PPARγ at 10 μM; Fu et al., 2003).



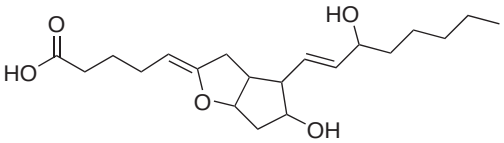
12 Nitrolinoleic acid (10- and 12-nitro-9,12-octadecadienoic acid), an anti-inflammatory cell signaling mediator generated by NO and fatty acid-dependent redox reactions (PPARγ EC_{50} = 0.05–0.62 μM, depending on regioisomer; Schopfer et al., 2005; Alexander et al., 2009).



13 15-deoxy-$\Delta^{12,14}$-prostaglandin J_2, an irreversible specific PPARγ agonist, covalently binds to Cys-285 of the LBD by a Michael addition reaction of its α,β-unsaturated ketone subgroup (shown in bold face; Forman et al., 1995; Shiraki et al., 2005).



14 Vaccenic acid (cis-11-octadecenoic acid) activates PPARβ/δ (Fyffe et al., 2006).

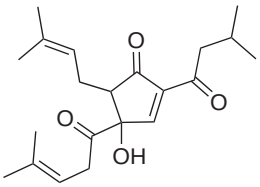


15 Prostacyclin (PGI_2) is a prostaglandin that activates PPARβ/δ (Gupta et al., 2000).

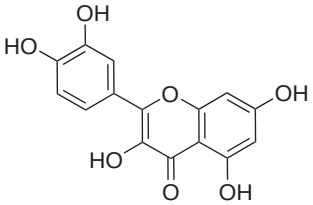
Scheme 4.1 Examples of endogenous PPAR ligands. PPAR = peroxisome proliferator-activated receptor, EC_{50} = half maximal effective concentration, LBD = ligand binding domain. Activities measured in different assays.

Natural product ligands

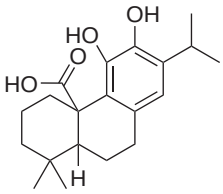
Substances produced by living organisms differ substantially from synthetic compounds, e. g., they have fewer aromatic atoms, more chiral centers, fewer nitrogen atoms, and more oxygen atoms (Henkel et al., 1999; Grabowski et al., 2008). This general observation also holds for PPAR ligands (Scheme 4.2). See Salam et al. (2008) for a virtual screening study using a natural product library.



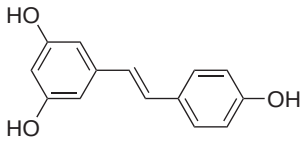
16 Isohumulone, an iso-α acid contained in hop (*Humulus lupulus* L.), gives beer its bitter flavor, and activates PPARα and PPARγ (Yajima et al., 2004; Shimura et al., 2005).



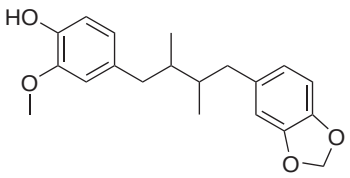
17 Quercetin is a flavonoid that occurs in fruits like mango and papaya, but also in vegetables and nuts. It shows vasodilator and antihypertensive effects (PPARα EC_{50} = 59.6 μM, PPARβ/δ EC_{50} = 76.9 μM, PPARγ EC_{50} = 56.3 μM; Wilkinson et al., 2008).



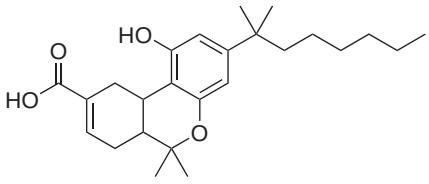
18 Carnosic acid, a phenolic diterpene of sage (*Salvia officinalis* L.), is known for its anti-oxidative and anti-microbial properties (PPARγ EC_{50} = 19.6 μM; Rau et al., 2006).



19 Resveratrol is a polyphenol occurring in grapes that regulates polyamine metabolism. It is thought to be cardioprotective and useful in cancer chemoprevention; it activates PPARγ (Ulrich et al., 2006).



20 Macelignan, a component of nutmeg (*Myristica fragrans* Houtt) with anti-diabetic properties, is a dual PPARα/γ agonist (PPARα EC_{50} = 5.4 μM, PPARγ EC_{50} = 4.2 μM; Han et al., 2008).



21 Ajulemic acid, a synthetic dimethylheptyl analog of tetrahydrocannabinol-11-oic acid, shows analgesic and anti-inflammatory, but not psychotropic activity (mPPARγ EC_{50} = 13 μM; inactive on hPPARα and hPPARβ/δ at 50 μM; Liu et al., 2003).

Scheme 4.2 Examples of natural compound PPAR ligands. PPAR = peroxisome proliferator-activated receptor, EC_{50} = half maximal effective concentration. Activities measured in different assays.

Synthetic ligands

Motivated by the various diseases linked to PPARs, a wide spectrum of synthetic ligands with different scaffolds, binding modes, activity profiles, side effects, and therapeutic uses has been developed over the last decades. After several late-stage failures of thiazolidinediones such as troglitazone (withdrawn from market), muraglitazar, and tesaglitazar (both abandoned after phase 3 clinical trials) due to severe adverse effects, research focus has shifted from full agonists to partial agonists, especially for PPARγ. Partial agonists, or, more generally, modulators of PPARs are thought to retain clinical efficacy without (some of) the adverse effects of full agonists (Cho et al., 2008).

All marketed selective PPARα agonists are fibrates, a class that has been in use since the 1960s (Hellman et al., 1963). Specific PPARβ/δ ligands are a recent development; consequently, there are no marketed selective PPARβ/δ drugs at the moment. PPARγ agonists are typically thiazolidinediones (Lehmann et al., 1995) and tyrosine analogs (Henke, 2004), although a variety of new scaffolds has been developed. Schemes 4.3, 4.4, and 4.5 show examples of PPARα, PPARβ/δ, and PPARγ ligands, respectively; Scheme 4.6 shows examples of PPAR pan agonists and modulators.

4.2 Retrospective evaluation

We used a published data set of PPARγ agonists to establish a regression model for PPARγ binding in preparation of the prospective virtual screening study in Section 4.3.

4.2.1 Data

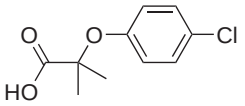
In a regression setting, the accuracy of the measured data labels — in our case, binding and activation of PPARγ — is important. As discussed before (p. 41), simply collecting compounds with associated measurements from the literature is not sufficient. For our study, we therefore selected a data set with homogeneous measurements from a single assay type.

Data set

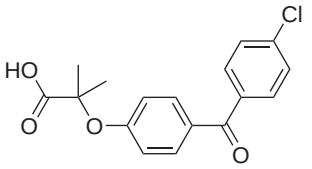
The public data set published by Rücker et al. (2006, Scheme 4.7) contains 176 compounds active on PPARγ. Of those, 144 compounds are annotated with the negative decadic logarithm of the PPARγ dissociation constant $\mathrm{p}K_i = -\log_{10} K_i$ as measured by a scintillation proximity assay (Nichols et al., 1998), 150 compounds are annotated with the negative decadic logarithm of the half maximal activation concentration $\mathrm{pEC}_{50} = -\log_{10} \mathrm{EC}_{50}$ as measured by a transient co-transfection assay (Lehmann et al., 1995), and 118 compounds are annotated with both values.

$\mathrm{p}K_i$ *versus* pEC_{50} *values*

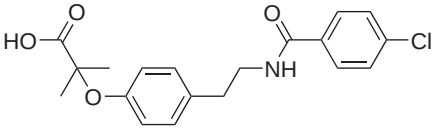
The $\mathrm{p}K_i$ values are measurements of association (binding) of compound to receptor, with higher values indicating stronger binding; the pEC_{50} values are measurements of receptor activation, with higher values indicating stronger activity. Binding and activity are related phenomena in the sense that binding is necessary but not sufficient for activation. The latter is usually determined in cell-based assays, which requires compounds to cross cell and nuclear membranes, to be soluble in the cytoplasm, to not be cytotoxic, and to not bind to other macromolecules in the cell in order to successfully activate a nuclear receptor. This relationship is reflected in the distribution of the $\mathrm{p}K_i$ and pEC_{50} values (Figure 4.6). In this work, we model binding constants ($\mathrm{p}K_i$ values), as they describe the more basic, and more predictable, phenomenon.



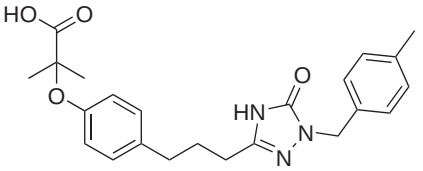
22 Clofibric acid, the active metabolite of clofibrate (hPPARα EC_{50} = 55 μM, hPPARγ EC_{50} $\approx$ 500 μM, inactive on hPPARβ/δ at 100 μM; Willson et al., 2000).



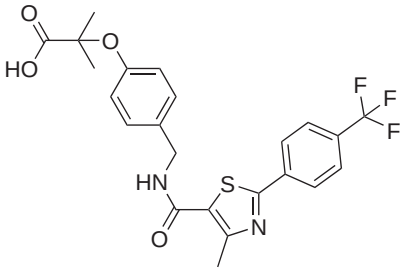
23 Fenofibric acid, the active metabolite of fenofibrate (hPPARα EC_{50} = 30 μM, hPPARγ EC_{50} $\approx$ 300 μM, inactive on hPPARβ/δ at 100 μM; Willson et al., 2000).



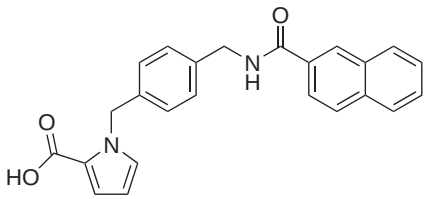
24 Bezafibrate (hPPARα EC_{50} = 50 μM, hPPARγ EC_{50} = 60 μM, hPPARβ/δ EC_{50} = 20 μM; Willson et al., 2000).



25 LY518674, a selective PPARα agonist (hPPARα EC_{50} = 46 nM, inactive on hPPARγ and hPPARβ/δ at 10 μM; Singh et al., 2005).

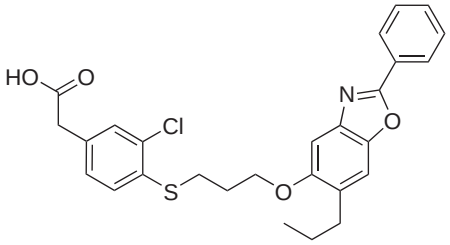


26 GW590735 (hPPARα EC_{50} = 4 nM, hPPARβ/δ EC_{50} = 3 μM, hPPARγ EC_{50} > 10 μM; Sierra et al., 2007).

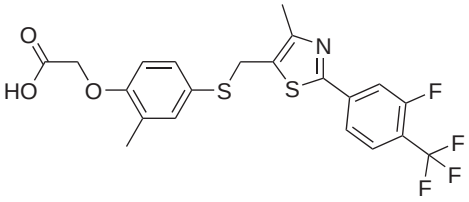


27 A variant of GW590735 obtained by fragment-based bioisosteric replacement of the acidic head group and the hydrophobic tail (PPARα EC_{50} = 0.51 μM, PPARγ EC_{50} = 0.63 μM; Proschak et al., 2008).

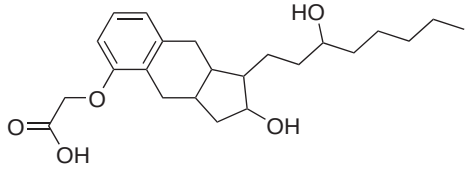
Scheme 4.3 Examples of PPARα agonists. The comparatively weak binders 22, 23, and 24 have been used since the 1960s to lower triglyceride levels (Hellman et al., 1963). The more recent compounds 25 and 26 are potent and highly selective against PPARγ. PPAR = peroxisome proliferator-activated receptor, hPPAR = human PPAR, EC_{50} = half maximal effective concentration. Activities measured in different assays.



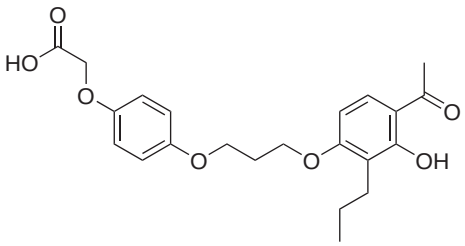
28 2-(3-chloro-4-(3-(2-phenyl-6-propyl-benzol[d]oxazol-5-yloxy) propylthio) phenyl) acetic acid, an early lead compound selective for PPARβ/δ (PPARα EC_{50} 90 nM, PPARβ/δ EC_{50} = 3 nM, PPARγ EC_{50} 300 nM; Jones, 2001).



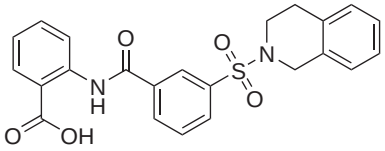
29 GW0742X (also GW610742), a selective PPARβ/δ agonist that reduces atherosclerosis in mice (hPPARα EC_{50} > 10 μM, hPPARβ/δ EC_{50} = 30 nM, hPPARγ EC_{50} > 10 μM; Graham et al., 2005a,b).



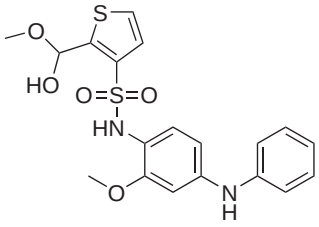
30 Treprostinil, a prostacyclin analog, activates PPARβ/δ and PPARγ, the latter in a prostacyclin receptor dependent way (Ali et al., 2006; Falcettia et al., 2007).



31 L-165041, a phenoxyacetic acid derivative, is selective for PPARβ/δ (hPPARα EC_{50} = 0.977 μM, hPPARβ/δ EC_{50} = 0.125 μM, hPPARγ EC_{50} = 1.824 μM; Berger et al., 1999; Basséne et al., 2006).

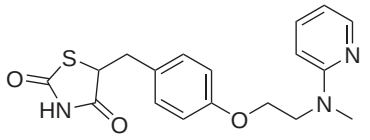


32 GW9371, an anthranilic acid derivative, is a partial PPARβ/δ agonist (hPPARβ/δ binding IC_{50} = 0.1 μM, inactive on hPPARα and hPPARγ at 10 μM; hPPARβ/δ EC_{50} = 1 μM; Shearer et al., 2008a).

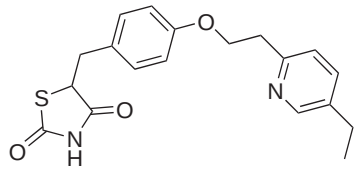


33 GSK0660 is a selective inverse agonist of PPARβ/δ (no agonist activity at 10 μM on all hPPAR subtypes; binding assay hPPARβ/δ IC_{50} = 155 nM; Shearer et al., 2008b).

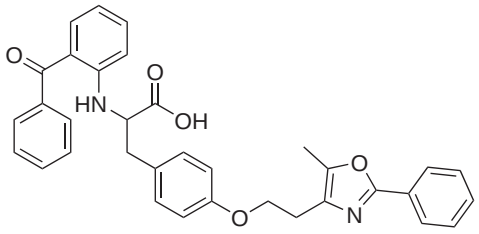
Scheme 4.4 Examples of PPARβ/δ ligands. PPAR = peroxisome proliferator-activated receptor, hPPAR = human PPAR, EC_{50} = half maximal effective concentration, IC_{50} = half maximal effective inhibition concentration. Activities measured in different assays.



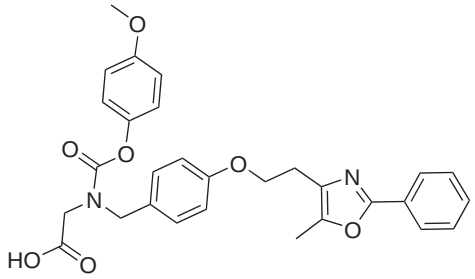
34 Rosiglitazone, a thiazolidinedione, is a marketed drug selective for PPARγ (hPPARα EC_{50} > 10 μM, hPPARβ/δ EC_{50} > 10 μM, hPPARγ EC_{50} = 18 nM; Oakes et al., 1994; Xu et al., 2001).



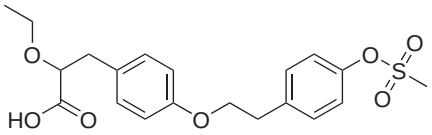
35 Pioglitazone, a thiazolidinedione, is another marketed drug that is selective for PPARγ (hPPARα EC_{50} > 10 μM, hPPARβ/δ EC_{50} > 10 μM, hPPARγ EC_{50} = 280 nM; Xu et al., 2001).



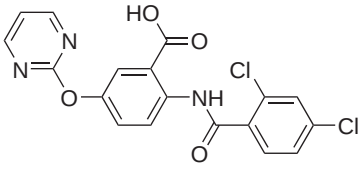
36 Farglitazar, an N-(2-benzoylphenyl)-L-tyrosine derivative, is currently in phase 2 clinical trials (hPPARα EC_{50} = 250 nM, hPPARβ/δ EC_{50} > 10 μM, hPPARγ EC_{50} = 0.2 nM; Henke et al., 1998; Xu et al., 2001).



37 Muraglitazar is a dual PPARα/γ agonist that was abandoned after phase 3 clinical trials (hPPARα EC_{50} = 320 nM, hPPARγ EC_{50} = 110 nM; Devasthale et al., 2005).

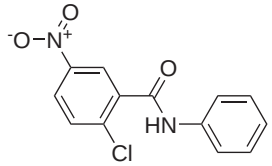


38 Tesaglitazar is another dual PPARα/γ agonist abandoned after phase 3 clinical trials (hPPARα EC_{50} = 1.7 μM, mPPARγ EC_{50} = 0.25 μM; Ljung et al., 2002).

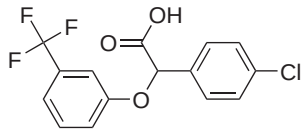


39 BVT.13, a 5-substituted 2-benzoylaminobenzoic acid (2-BABA) derivative with a binding epitope different from that of the thiazolidinediones (PPARγ EC_{50} = 1.3 μM, inactive on PPARα and PPARβ/δ at 10 μM; Östberg et al., 2004).

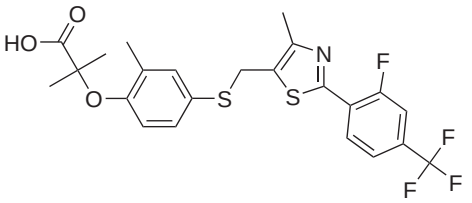
Scheme 4.5 Examples of synthetic PPARγ agonists. PPAR = peroxisome proliferator-activated receptor, hPPAR = human PPAR, mPPAR = mouse PPAR, EC_{50} = half maximal effective concentration. Activities measured in different assays.



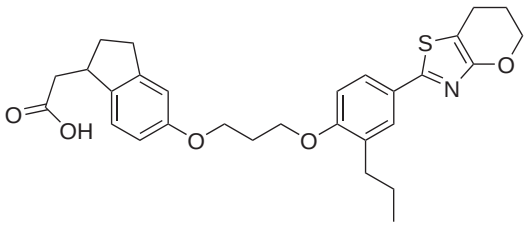
40 GW9662, a selective PPARγ full antagonist and partial PPARα agonist (PPARα EC_{50} = 26 nM, PPARβ/δ IC_{50} = 471 nM, PPARγ IC_{50} = 3.8 nM; Seimandi et al., 2005).



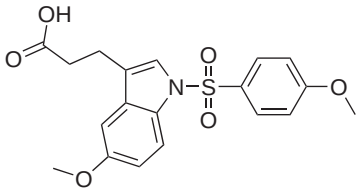
41 Metaglidasen (MBX-102, the (-) enantiomer of the clofibric acid analog halofenate), is a PPARγ modulator currently in phase 3 clinical trials. It exhibits partial agonist and antagonist activity (PPARγ EC_{50} = 18 μM, inactive on PPARα and PPARβ/δ; Allen et al., 2006; Meinke et al., 2006; Rubenstrunk et al., 2007).



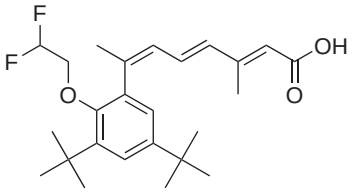
42 Sodelglitazar (GW677954), a PPAR pan agonist for the treatment of diabetes 2, is currently in phase 2 clinical studies (PPARα EC_{50} = 40 nM, PPARβ/δ EC_{50} = 1.3 nM, PPARγ EC_{50} = 63 nM; Evans et al., 2005).



43 Indanylacetic acid derivate 34r, a PPAR pan agonist (hPPARα EC_{50} = 101 nM, hPPARβ/δ EC_{50} = 4 nM, mPPARγ EC_{50} = 42 nM; Rudolph et al., 2007).

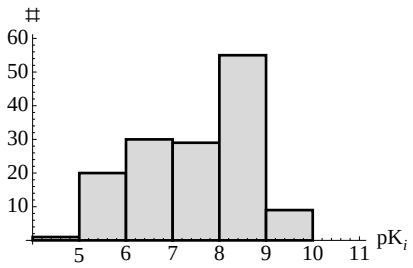


44 Indeglitazar is a recent PPAR pan agonist and partial PPARγ agonist currently in phase 2 clinical trials (PPARα EC_{50} = 0.99 μM, PPARβ/δ EC_{50} = 1.3 μM, PPARγ EC_{50} = 0.85 μM; Artis et al., 2009).

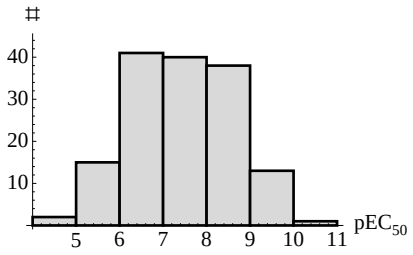


45 LG1506 $\big((2E,4E,6Z)$-7-(2-Alkoxy-3,5-dialkylbenzene)-3-methylocta-2,4,6-trienoic acid$\big)$ is a hetero-dimer-selective RXR modulator that activates RXR:PPARα and RXR:PPARγ, but does not suppress the thyroid hormone axis (RXR:PPARα EC_{50} = 15 nM, RXR:PPARγ EC_{50} = 3 nM; Michellys et al., 2003; Leibowitz et al., 2006).

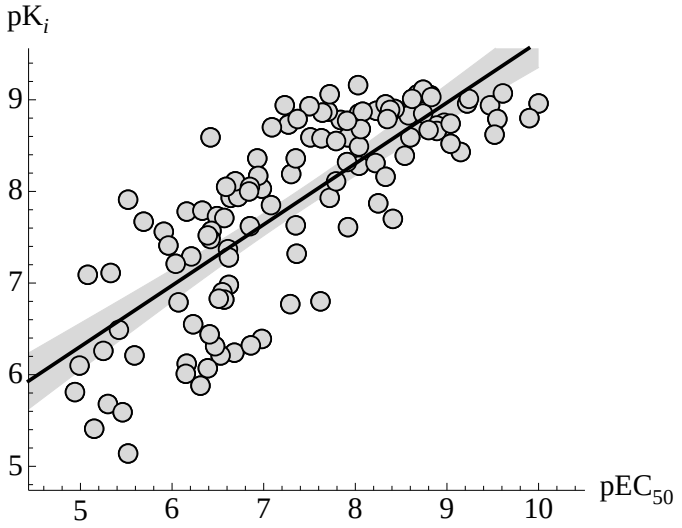
Scheme 4.6 Examples of synthetic PPAR modulators. PPAR = peroxisome proliferator-activated receptor, RXR = retinoid X receptor, hPPAR = human PPAR, EC_{50} = half maximal effective concentration, IC_{50} = half maximal effective inhibitory concentration. Activity values measured in different assays.



(a) Histogram of pK_i-values. $n = 144$ samples, 7 bins of unit length.



(b) Histogram of pEC_{50}-values. $n = 150$ samples, binning as in (a).



(c) pK_i versus pEC_{50} values. $n = 118$ samples (gray disks) with experimentally determined pK_i and pEC_{50} values. Squared correlation $r^2 = 0.62$. The solid black line shows the linear model $\mathrm{p}K_i = 2.98 + 0.67\mathrm{pEC}_{50}$ with the 95 % confidence band of the mean in gray.

Figure 4.6 Distribution of pK_i and pEC_{50} values in the `ppar` data set. 176 compounds in total, of which 144 have experimentally determined pK_i values, 150 have experimentally determined pEC_{50} values, and 118 have both values.

Compound classes

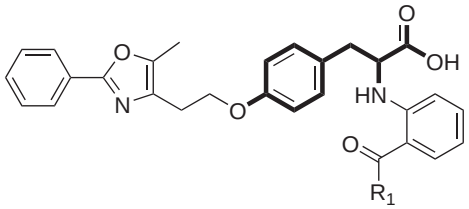
Based on Rücker et al. (2006), we divide the compounds into eleven classes, consisting of indoles, thiazolidinediones, thiazolidinedione-fatty acid hybrids, oxadiazoles, phenoxyisobutyric acids, fatty acids, and five classes of tyrosine derivatives (Scheme 4.7).

4.2.2 Descriptors and kernels

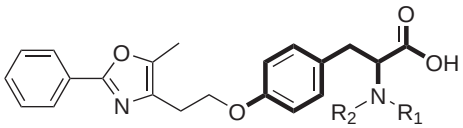
We described compounds with molecular descriptors, the annotated structure graph, and combinations of these. For numerical representations (CATS2D, MOE 2D, Ghose-Crippen), compounds were preprocessed with the software MOE (molecular operating environment, Chemical Computing Group, `www.chemcomp.com`) by removing salts, neutralizing compounds, and (de)protonating strong bases (acids). Implicit hydrogens were removed for graph representations.

CATS2D

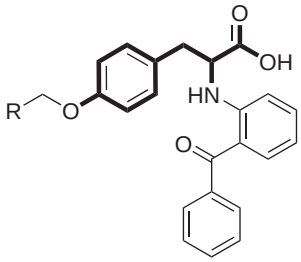
We used the CATS2D descriptor (chemically advanced template search, Schneider et al., 1999; Fechner et al., 2003), a topological pharmacophore-based auto-correlation vector, in a version extended by an aromaticity pharmacophore type. Atoms in a compound were assigned zero or more of the pharmacophore types hydrogen-bond acceptor (A), hydrogen-bond donor (D), negative charge (N), positive charge (P), lipophilic (L), and aromatic (M). The number of occurrences of each of the possible 21 pairs AA, AD, AN, AP, AL, AM, DD, DN, ..., MM within a certain topological distance were counted; we used distances up to ten bonds, resulting in a $21 \cdot 10 = 210$ dimensional vector.



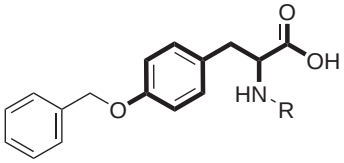
46 Tyrosines with an oxazole derivative as O-substituent, 18 compounds.



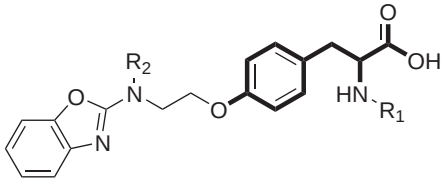
47 Tyrosines with oxazole O-substituent and small N-substituent, 12 compounds.



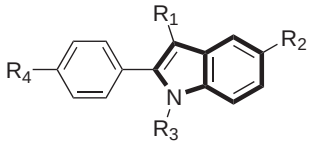
48 Tyrosines with a benzophenone N-substituent, 37 compounds.



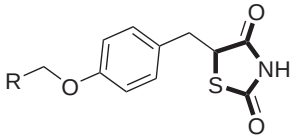
49 Tyrosines with ethylbenzene O-substituent, 7 compounds.



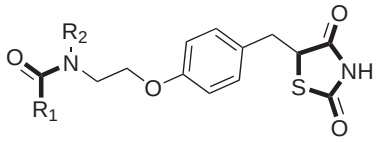
50 Tyrosines with benzoxazole derivatives as O-substituents, 31 compounds.



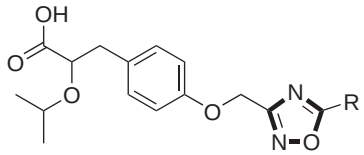
51 Indoles with 2-substituted 1,4-dimethylbenzene, 10 compounds.



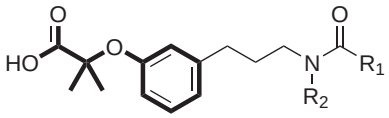
52 Thiazolidinediones with substituted p-ethylphenetole and no fatty acid group, 6 compounds.



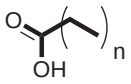
53 Thiazolidinedione-fatty acid hybrids with substituted p-ethylphenetole, 16 compounds.



54 1,2,4-oxadiazoles 3,5-substituted with isopropoxyacetic acid derivatives, 23 compounds.



55 Tertiary amides with phenoxyisobutyric acid derived substituents, 6 compounds.



56 Fatty acids of different length and degrees of saturation, 9 compounds.

Scheme 4.7 Classes in data set `ppar`. Primary name-giving scaffold shown in bold. 5 compounds are not assigned to a class.

MOE 2D

All 184 two-dimensional descriptors computed by MOE were used.

Ghose-Crippen

Ghose-Crippen fragment descriptors (Ghose and Crippen, 1986, 1987; Ghose et al., 1988; Viswanadhan et al., 1989) consist of 120 fragments, or atom types, originally introduced for the prediction of hydrophobic interactions of small organic compounds.

Annotated structure graph

We used the structure graph in three forms:

- *Topological:* No annotation was used, i. e., atoms correspond to vertices and covalent bonds correspond to edges. No information about atom or bond type was retained.
- *Element types:* Vertices were annotated with element type (atomic number), bonds with bond type (single, double, triple).
- *Pharmacophore types:* Atoms were annotated as potential pharmacophore points according to Table 2.5.

Kernels

For numerical representations, we used the standard (linear) inner product, the homogeneous polynomial kernel (p. 30), the Gaussian kernel (p. 30), and, the *rational quadratic kernel* (also Students-t kernel; Rasmussen and Williams, 2006)

$$k(\boldsymbol{x}, \boldsymbol{x}') = \left(1 + \frac{||\boldsymbol{x} - \boldsymbol{x}'||^2}{sl}\right)^{-\frac{s+d}{2}}, \tag{4.1}$$

where $s > 0$ is the shape parameter, $l > 0$ is the scale parameter, and d is the dimension of the input space. The shape parameter s defines the thickness of the kernel tails. The rational quadratic kernel corresponds to an infinite mixture of scaled squared exponentials; the squared exponential itself is recovered for $s \to \infty$. In Gaussian process models (Subsection 4.2.4), the rational quadratic kernel often yields predictions as accurate as those of the Gaussian kernel, but with better confidence estimates.

For graph representations, we used ISOAK (Chapter 2), with no vertex and edge kernels ($v_k = e_k = 1$) on the topological graph, and Dirac vertex and edge kernel (p. 72) on element type and pharmacophore type annotation.

Multiple kernel learning

Multiple kernel learning (Sonnenburg et al., 2006) combines different kernels by using the fact that linear combinations (with non-negative coefficients) of kernels are again kernels. Combining different kernels on the same input allows one to learn different types of patterns at the same time, e. g., polynomial and exponential patterns. Combining kernels on different inputs enables the combination of heterogeneous input representations, e. g., descriptors and structure graphs. Each additional kernel adds another free parameter; these can be optimized, e. g., using cross-validation and grid search or gradient descent.

4.2.3 Baseline models

We used linear ridge regression and support vector machines as baseline methods.

Ridge regression

Ridge regression (Hastie et al., 2003) is a regularized form of linear regression with quadratic loss. The learned model has the form $f(\boldsymbol{x}) = \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b$, where $\boldsymbol{w} \in \mathbb{R}^d$ is a weight vector and $b \in \mathbb{R}$ is a bias term. The solution minimizes the quadratic error on the training data and the norm of the weight vector,

$$\min_{\boldsymbol{w},b} \frac{1}{n} \sum_{i=1}^{n} \big(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b - y_i \big)^2 + \lambda \left\| \boldsymbol{w} \right\|^2 , \tag{4.2}$$

where $\lambda > 0$ determines the trade-off between goodness of fit and regularization.

In unregularized linear regression on correlated input, like CATS2D auto-correlation vectors, arbitrarily large weights on single components can be canceled by corresponding negative weights on other, correlated components. The regularizer $\left\| \boldsymbol{w} \right\|^2$ prevents this by penalizing the squared regression weights, ensuring small weights of similar magnitude.

Support vector machines

We used support vector machines (Subsection 2.4.3) for regression. For vectorial descriptors, we employed the Gaussian kernel, which is a suitable default choice based on both general considerations (p. 30) and practical experience.

4.2.4 Gaussian processes

Gaussian processes (GP) are a supervised machine learning approach originating from Bayesian statistics. A major advantage of GPs is that predictions come with a measure of confidence in the prediction, i. e., built-in domain of applicability (p. 49) estimation. We restrict our presentation to aspects of GPs required for this chapter, in particular GP regression. For further information, see Rasmussen and Williams (2006).

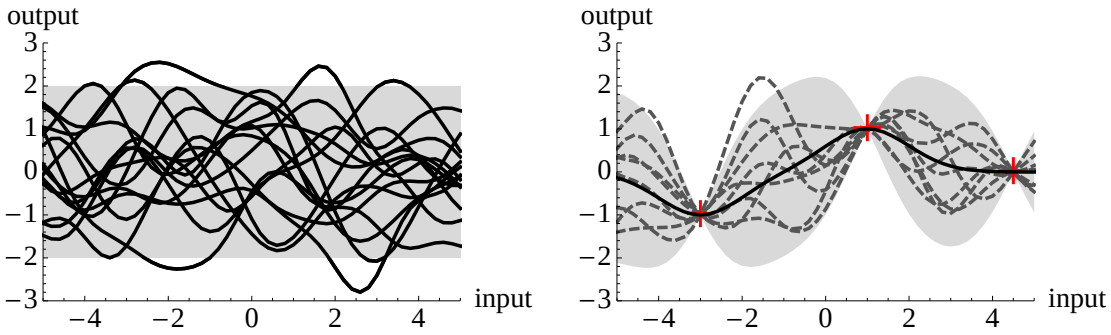
Introduction

Gaussian processes are a generalization of normal (Gaussian) distributions to functions, i. e., to infinitely many random variables such that every finite subset of these variables has again normal distribution. Formally, let $\mathcal{X}$ denote the index set (corresponding to input space) of a stochastic process $p(x) \in \mathbb{R}$, $x \in \mathcal{X}$, with real state space (corresponding to label space, in the case of regression real numbers). The stochastic process p is a *Gaussian process* iff $\big(p(x_1), \ldots, p(x_n)\big) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{K})$ for all $x_1, \ldots, x_n \in \mathcal{X}$, where $\boldsymbol{\mu} \in \mathbb{R}^n$ is the mean and $\boldsymbol{K} \in \mathbb{R}^{n \times n}$ is the (strictly positive definite) covariance matrix.[12] In other words, $p(x) = \big(p(x_1), \ldots, p(x_n)\big)$ has probability density

$$\frac{1}{\sqrt{(2\pi)^n \det \boldsymbol{K}}} \exp\left(-\frac{1}{2} \big(p(x) - \boldsymbol{\mu}\big)^T \boldsymbol{K}^{-1} \big(p(x) - \boldsymbol{\mu}\big) \right) , \tag{4.3}$$

where $\det \boldsymbol{K}$ is the determinant of $\boldsymbol{K}$. A GP is completely determined by its mean and covariance function.

[12] Note that since $\boldsymbol{K}$ is positive definite, the process is automatically *consistent* in the sense that if $(x, x') \sim \mathcal{N}\big((\boldsymbol{\mu}, \boldsymbol{\mu}'), \left(\begin{smallmatrix} \boldsymbol{K} & \boldsymbol{K}'' \\ \boldsymbol{K}''^T & \boldsymbol{K}' \end{smallmatrix}\right)\big)$, then $x \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{K})$, i. e., consideration of a larger set of variables does not change the distribution of the smaller set.



(a) Prior distribution. 15 samples (black lines) drawn from a Gaussian process with zero mean and squared exponential covariance function $k(x, x') = \exp(-\frac{1}{2}|x - x'|^2)$.

(b) Posterior distribution with mean function (black line). 8 samples (gray dashed lines) drawn from the posterior distribution after introduction of three observations (red crosses).

Figure 4.7 Idea of Gaussian process regression. Starting from the prior distribution (a), one conditions on the observed samples. The mean and variance of the posterior distribution (b) are used as predictor and confidence estimate. Shaded regions denote two standard deviations.

Idea

The intuition behind GP regression is to

1 Start with a class of admissible functions, here samples[13] from a GP (Figure 4.7a).

2 Put a suitable prior distribution over these functions.

3 Condition the process on observed sample-label pairs (x_i, y_i) (Figure 4.7b).

4 Use the mean and the covariance of the posterior distribution as predictor and confidence estimate, respectively.

Note that the distribution of the input samples is not modeled explicitly.

Linear regression

For linear regression in input space, we assume $\mathcal{X} = \mathbb{R}^d$ and that target labels are corrupted by additive i. i. d. Gaussian noise with zero mean and variance σ^2,

$$y_i = \langle \boldsymbol{x}_i, \boldsymbol{w} \rangle + b + \epsilon, \qquad \epsilon \sim \mathcal{N}\big(0, \sigma^2\big)\,, \tag{4.4}$$

where $b \in \mathbb{R}$ is a bias term and $\boldsymbol{w} \in \mathbb{R}^d$ is a weight vector. In the following, we assume $b = 0$ for simplicity.[14] For given weights $\boldsymbol{w}$ and training data matrix $\boldsymbol{X} = (\boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_n^T) \in \mathbb{R}^{n \times d}$, the *likelihood* of the labels $\boldsymbol{y} \in \mathbb{R}^n$ is given by

$$\begin{aligned} \mathbb{P}(\boldsymbol{y} \,|\, \boldsymbol{X}, \boldsymbol{w}) &= \prod_{i=1}^{n} \mathbb{P}(\boldsymbol{y}_i \,|\, \boldsymbol{x}_i, \boldsymbol{w}) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\boldsymbol{y}_i - \langle \boldsymbol{x}_i, \boldsymbol{w} \rangle)^2}{2\sigma^2}\right) \\ &= (2\pi\sigma^2)^{-n/2} \exp\left(-\frac{1}{2\sigma^2} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|^2\right) \sim \mathcal{N}\big(\boldsymbol{X}\boldsymbol{w}, \sigma^2 \boldsymbol{I}_n\big)\,. \end{aligned} \tag{4.5}$$

[13] In this chapter, the term *sample* can refer to either a sample from a GP, i. e., a function $\mathcal{X} \to \mathcal{Y}$, or, a (training or test) input sample, i. e., $x \in \mathcal{X}$.

[14] The bias can be treated either by explicitly incorporating it into the following calculations, by adding a component which is always 1 to the input vectors, or, by standardizing training data labels.

We put a $\mathcal{N}(0, \boldsymbol{\Sigma})$-distributed prior over the weights $\boldsymbol{w}$ and use *Bayes rule*

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}, \quad \mathbb{P}(\boldsymbol{w} \,|\, \boldsymbol{y}, \boldsymbol{X}) = \frac{\mathbb{P}(\boldsymbol{y} \,|\, \boldsymbol{X}, \boldsymbol{w})\, \mathbb{P}(\boldsymbol{w})}{\mathbb{P}(\boldsymbol{y} \,|\, \boldsymbol{X})} \tag{4.6}$$

to compute the posterior. Since $\mathbb{P}(\boldsymbol{y} \,|\, \boldsymbol{X})$ does not depend on $\boldsymbol{w}$, it is proportional to

$$\begin{aligned} \mathbb{P}(\boldsymbol{w} \,|\, \boldsymbol{y}, \boldsymbol{X}) &\propto \exp\left(-\frac{1}{2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^T \sigma^{-2} \boldsymbol{I}_n (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})\right) \exp\left(-\frac{1}{2}\boldsymbol{w}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{w}\right) \\ &\propto \exp\Big(-\frac{1}{2}(\boldsymbol{w} - \tilde{\boldsymbol{w}})^T \boldsymbol{A}\, (\boldsymbol{w} - \tilde{\boldsymbol{w}})\Big) \sim \mathcal{N}\big(\tilde{\boldsymbol{w}}, \boldsymbol{A}^{-1}\big)\,, \end{aligned} \tag{4.7}$$

with $\boldsymbol{A} = \sigma^{-2}\boldsymbol{X}^T\boldsymbol{X} + \boldsymbol{\Sigma}^{-1}$ and $\tilde{\boldsymbol{w}} = \sigma^{-2}\boldsymbol{A}^{-1}\boldsymbol{X}^T\boldsymbol{y}$.[15] Predicting the label y' of a test sample $\boldsymbol{x}'$ is done by averaging label predictions over all possible models $\boldsymbol{w}$, weighted by their posterior distribution, using Equations 4.5 and 4.7:

$$\begin{aligned} \mathbb{P}\big(y' \,\big|\, \boldsymbol{x}', \boldsymbol{y}, \boldsymbol{X}\big) &= \int_{\mathbb{R}^d} \mathbb{P}\big(y' \,\big|\, \boldsymbol{x}', \boldsymbol{w}\big)\, \mathbb{P}(\boldsymbol{w} \,|\, \boldsymbol{y}, \boldsymbol{X})\, d\boldsymbol{w} \\ &\sim \mathcal{N}\big(\sigma^{-2}\boldsymbol{x}'^T \boldsymbol{A}^{-1} \boldsymbol{X}^T \boldsymbol{y}, \boldsymbol{x}'^T \boldsymbol{A}^{-1} \boldsymbol{x}'\big). \end{aligned} \tag{4.8}$$

Note that the predictive variance does not include σ^2. Computing $\tilde{\boldsymbol{w}}$ and $\boldsymbol{A}^{-1}$ requires inversion of a $d \times d$ matrix, which can be done in time $O(d^3)$. Evaluating Equation 4.8 to predict a test sample has linear cost for the mean and quadratic cost for the variance. Figure 4.8 shows an example.

The kernel trick

Application of the kernel trick (p. 28) using a non-linear transformation $\phi : \mathcal{X} \to \mathcal{H}$, $\boldsymbol{x} \mapsto \phi(\boldsymbol{x})$ from input space into feature space and applying linear Bayesian regression there gives the feature space posterior distribution

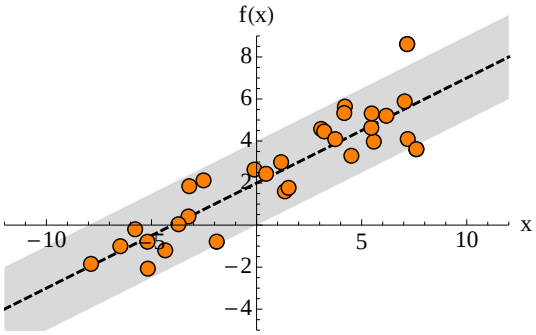
$$\mathbb{P}\big(y' \,\big|\, \boldsymbol{x}', \boldsymbol{y}, \boldsymbol{X}\big) \sim \mathcal{N}\Big(\sigma^{-2}\phi(\boldsymbol{x}')^T \boldsymbol{A}^{-1} \phi(\boldsymbol{X})^T \boldsymbol{y},\ \phi(\boldsymbol{x}')^T \boldsymbol{A}^{-1} \phi(\boldsymbol{x}')\Big)\,, \tag{4.9}$$

with $\boldsymbol{A} = \sigma^{-2}\phi(\boldsymbol{X})^T\phi(\boldsymbol{X}) + \boldsymbol{\Sigma}^{-1}$; here, the weight prior $\boldsymbol{\Sigma}$ has the dimensionality of the feature space. Since n training samples can span a subspace of dimension at most n, we can rewrite Equation 4.9 as (Rasmussen and Williams, 2006)

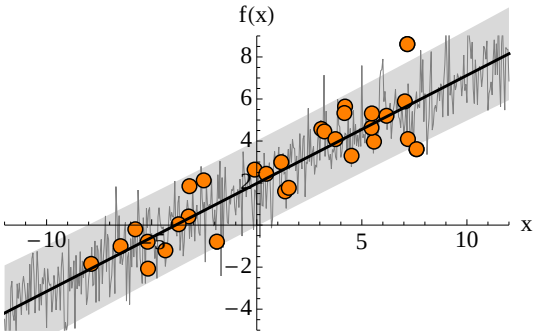
$$\begin{aligned} \mathbb{P}\big(y' \,\big|\, \boldsymbol{x}', \boldsymbol{y}, \boldsymbol{X}\big) \sim \mathcal{N}\Big(&\phi(\boldsymbol{x}')^T \boldsymbol{\Sigma} \phi(\boldsymbol{X})^T \big(\boldsymbol{K} + \sigma^2 \boldsymbol{I}\big)^{-1} \boldsymbol{y}, \\ &\phi(\boldsymbol{x}')^T \boldsymbol{\Sigma} \phi(\boldsymbol{x}') - \big(\phi(\boldsymbol{X}) \boldsymbol{\Sigma} \phi(\boldsymbol{x}')\big)^T \big(\boldsymbol{K} + \sigma^2 \boldsymbol{I}\big)^{-1} \phi(\boldsymbol{X}) \boldsymbol{\Sigma} \phi(\boldsymbol{x}')\Big), \end{aligned} \tag{4.10}$$

where $\boldsymbol{K} = \phi(\boldsymbol{X})\boldsymbol{\Sigma}\phi(\boldsymbol{X})^T$. Note that all feature space evaluations are of the form $\phi(\cdot)^T\boldsymbol{\Sigma}\phi(\cdot)$, which is an inner product weighted by the positive definite matrix $\boldsymbol{\Sigma}$, and can be replaced by a positive definite kernel $k(\cdot, \cdot)$.

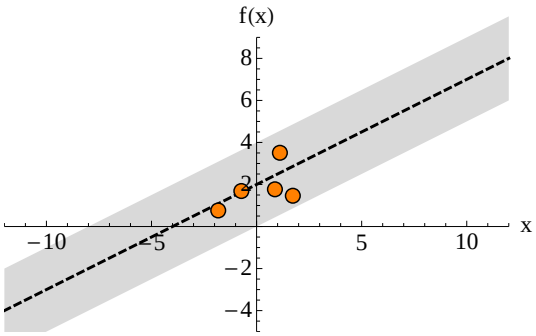
[15] To see this, use $\langle \boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}, \boldsymbol{y} - \boldsymbol{X}\boldsymbol{w} \rangle = \boldsymbol{y}^T\boldsymbol{y} - 2\sigma^{-2}\boldsymbol{y}^T\boldsymbol{X}\boldsymbol{w} + \sigma^{-2}\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w}$, drop $\boldsymbol{y}^T\boldsymbol{y}$ and add $\sigma^{-4}\boldsymbol{y}^T\boldsymbol{X}\boldsymbol{A}^{-1}\boldsymbol{X}^T\boldsymbol{y}$, both of which do not depend on $\boldsymbol{w}$.



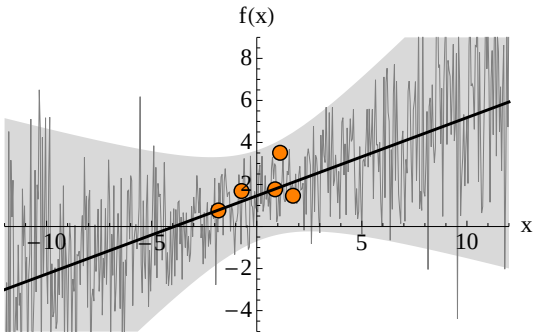
(a) Linear input data. $n = 30$ samples with independent x-coordinates drawn uniformly from $[-8, 8]$ and noisy labels $y = f(x) + \mathcal{N}(0, 1)$.



(b) Linear Bayesian regressor $0.51x + 1.98$. Enough samples result in a good fit with high confidence.



(c) Linear input data. As in (a), except $n = 5$ with x-coordinates from $[-2, 2]$.



(d) Linear Bayesian regressor $0.37x + 1.46$. Few samples result in a bad fit with rapidly increasing predictive variance.

Figure 4.8 Bayesian linear regression examples. Shown are samples (orange disks), the noise-free label function $f(x) = \frac{1}{2}x + 2$ (dashed line) $\pm$ two standard deviations (gray area), posterior distribution samples (gray lines), and Bayesian regressors (solid lines).

Non-linear regression

Gaussian process regression is linear Bayesian regression in feature space. We replace (weighted) inner products by kernel evaluations and shift to matrix notation. Let $\boldsymbol{K} \in \mathbb{R}^{n \times n}$, $\boldsymbol{K}_{i,j} = k(x_i, x_j)$, $\boldsymbol{K}' \in \mathbb{R}^{m \times m}$, $\boldsymbol{K}'_{i,j} = k(x'_i, x'_j)$, and $\boldsymbol{L} \in \mathbb{R}^{n \times m}$, $\boldsymbol{L}_{i,j} = k(x_i, x'_j)$ denote the kernel matrices on training samples, test samples, and training versus test samples, respectively. The prior over the noisy labels becomes

$$\operatorname{covar}(\boldsymbol{y}) = \boldsymbol{K} + \sigma^2 \boldsymbol{I}. \tag{4.11}$$

The joint distribution of the training and test labels according to this prior is

$$\begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{y}' \end{pmatrix} = \mathcal{N}\left(0, \begin{pmatrix} \boldsymbol{K} + \sigma^2 \boldsymbol{I} & \boldsymbol{L} \\ \boldsymbol{L}^T & \boldsymbol{K}' \end{pmatrix}\right). \tag{4.12}$$

Conditioning the joint distribution on the observed samples yields

$$\mathbb{P}\left(\boldsymbol{y}' \,\middle|\, \boldsymbol{y}, \boldsymbol{X}, \boldsymbol{X}'\right) \sim \mathcal{N}\Big(\boldsymbol{L}^T \big(\boldsymbol{K} + \sigma^2 \boldsymbol{I}\big)^{-1} \boldsymbol{y},\ \boldsymbol{K}' - \boldsymbol{L}^T \big(\boldsymbol{K} + \sigma^2 \boldsymbol{I}\big)^{-1} \boldsymbol{L}\Big). \tag{4.13}$$

Note that the prediction for a single test sample x' can be rewritten as

$$\sum_{i=1}^{n} \alpha_i k(x_i, x') \quad \text{with } \alpha = \big(\boldsymbol{K} + \sigma^2\big)^{-1} \boldsymbol{y}, \tag{4.14}$$

Table 4.4 Applications of Gaussian processes in cheminformatics, sorted by submission date where available. T = evaluation type, with R = retrospective, B = blind (separate test data available only after training, evaluation by different team), and P = prospective. Bz agonism = benzodiazepine receptor agonism, hERG inhib. = human ether-a-go-go-related gene inhibition, M_1 inhib. = M_1 muscarinic receptor inhibition, *T. pyriformis* toxicity = *Tetrahymena pyriformis* toxicity.

Application	T	Reference
Bz agonism, M_1 inhib., *T. pyriformis* toxicity	R	Burden (2001)
Lipophilicity	R	Enot et al. (2001)
Ovarian cancer identification from mass spectrometry	R	Yu and Chen (2005)
Aqueous solubility	B	Schwaighofer et al. (2007)
Blood-brain barrier, hERG inhib., aqueous solubility	R	Obrezanova et al. (2007)
Aqueous solubility	R	Schroeter et al. (2007c)
Lipophilicity	B	Schroeter et al. (2007b)
Lipophilicity	P	Schroeter et al. (2007a)
Metabolic stability	P	Schwaighofer et al. (2008)
Amphiphysin SH3 domain peptide binding	R	Zhou et al. (2008)
Spectroscopic calibration	R	Chen and Martin (2008)
Synthetic data	R	Sakiyama (2009)
Liquid chromatographic retention times	R	Tian et al. (2009)
Liquid chromatographic retention times	R	Zhou et al. (2009)
hERG inhib.	R	Hansen et al. (2009)

and that the predictive variance does not depend on the labels, but only on the distances between the samples. Since only inner products are used, samples do not have to be vectors anymore, but can be elements of any input space $\mathcal{X}$ endowed with a positive definite kernel k. Figure 4.9 shows an example of non-linear Gaussian process regression.

Applications in cheminformatics

GP theory goes back at least to the 1940s (Kolmogoroff, 1941; Doob, 1944; Wiener, 1949), and perhaps even earlier (Lauritzen, 1981). Despite many applications in diverse areas such as bioinformatics, environmental sciences, geostatistics (kriging), manufacturing, machine learning, medicine and health, music, physics, robotics, and others, GPs have only recently been introduced to cheminformatics, mainly for the prediction of physicochemical properties of small molecules (Table 4.4).

4.2.5 Performance estimation

We use clustered cross-validation and two distinct performance measures to retrospectively evaluate baseline and GP models.

Statistical validation

In virtual screening data sets, the assumption of independent and identically distributed training samples is violated (p. 35; Rupp et al., 2009); for the `ppar` data set this is particularly clear from its strong structure (Scheme 4.7). In such cases, statistical estimation procedures like leave-one-out cross-validation and k-fold cross-validation can overestimate performance.

We used a leave-k-cluster-out cross-validation strategy by partitioning the data set into $k = 10$ clusters using GeoClust (Choudhury et al., 2002; Wen et al., 2005) and applying 10 runs of leave-5-clusters-out cross-validation. The same folds were used for all models.

Performance measures

Let y_i denote the target labels, let $\hat{y}_i$ denote their estimates, and, let Y and $\hat{Y}$ denote corresponding random variables. As performance measures, we employed the correlation coefficient r (Pearsons correlation), the mean absolute error MAE, the root mean squared error RMSE,

$$r = \frac{\operatorname{covar}\big(Y, \hat{Y}\big)}{\sqrt{\operatorname{var}\big(Y\big)\operatorname{var}\big(\hat{Y}\big)}} = \frac{\sum_{i=1}^{n}\big(y_i - \frac{1}{n}\sum_{j=1}^{n} y_j\big)\big(\hat{y}_i - \frac{1}{n}\sum_{j=1}^{n} \hat{y}_j\big)}{\sqrt{\sum_{i=1}^{n}\big(y_i - \frac{1}{n}\sum_{j=1}^{n} y_j\big)^2 \sum_{i=1}^{n}\big(\hat{y}_i - \frac{1}{n}\sum_{j=1}^{n} \hat{y}_j\big)^2}}, \tag{4.15}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|\,, \qquad \mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \tag{4.16}$$

and, cumulative histograms. For our application, i. e., the selection of a model suited for later prospective virtual screening with a small number of assay tests, erroneous predictions within the top-ranked compounds (false positives) are more costly than errors for lower-ranked compounds, making early recognition (p. 42) important. We therefore used an additional performance measure to account for the early recognition problem, the fraction of inactives among the 20 best-ranked compounds (FI_{20}).

Ranking

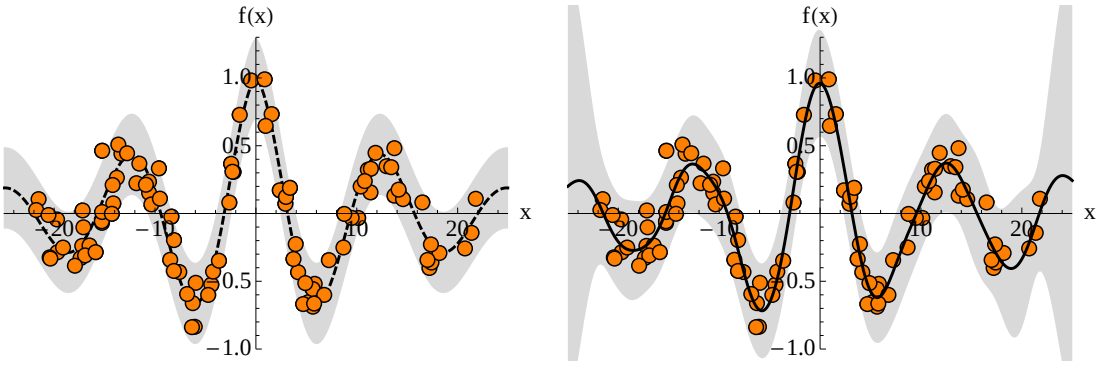
For ridge regression and support vector machines, compounds were ranked according to predicted affinity. For Gaussian process models, both the estimated affinity $\hat{\mu}$ and the estimation confidence $\hat{\sigma}^2$ had to be incorporated into the ranking. We combined both into a single score $\hat{\mu} - \hat{\sigma}^2$ by subtracting the variance of a prediction from its mean (Guiver and Snelson, 2008). This is valid because variance and mean have the same scale, and prefers compounds with high predicted affinity and high confidence in the prediction.

4.2.6 Results

The described molecular representations, kernels, and machine learning algorithms were combined into 16 models (Figure 4.10), each model was subjected to 10 runs of leave-5-clusters-out cross-validation, and all performance measures were computed, averaged over all runs. The performance measures r, MAE, RMSE, and cumulative histograms were highly correlated (correlation > 0.95); in the following, we report only the MAE.

Models

Models 1, 2, and 3 are baseline linear ridge regression models using three different vectorial descriptors. They represent linear methods with standard chemical descriptors. Models 4, 5, and 6 are baseline support vector regression models with Gaussian kernels and the same vectorial descriptors. They represent established non-linear methods with standard chemical descriptors. Models 7, 8, and 9 are Gaussian process models with Gaussian and rational quadratic kernels using the same vectorial descriptors. They represent the Gaussian process approach with standard chemical descriptors.



(a) Decaying sine input data. $n = 100$ samples (orange disks) with independent x-coordinates drawn uniformly from $[-22, 22]$ and noisy labels $y = f(x) + \mathcal{N}(0, 0.15)$, where $f(x) = \sin\left(\frac{1}{2}(x+\pi)\right)\exp(-\frac{1}{15}|x|)$. The dashed line indicates the noise-free label function $f \pm$ two standard deviations (gray area).

(b) Gaussian process regressor (black line) $\pm$ two standard deviations (gray area), using a squared exponential kernel $k(x_i, x_j) = \exp\left(-\frac{1}{2 \cdot 3^2}(x_i - x_j)^2\right) + \sigma^2 1_{\{i=j\}}$. The length-scale of 3 enforces a smooth regressor. Note how variance is higher in areas with fewer samples, e. g., at $x = 19$ as compared to $x = -19$.

Figure 4.9 Example of Gaussian process regression.

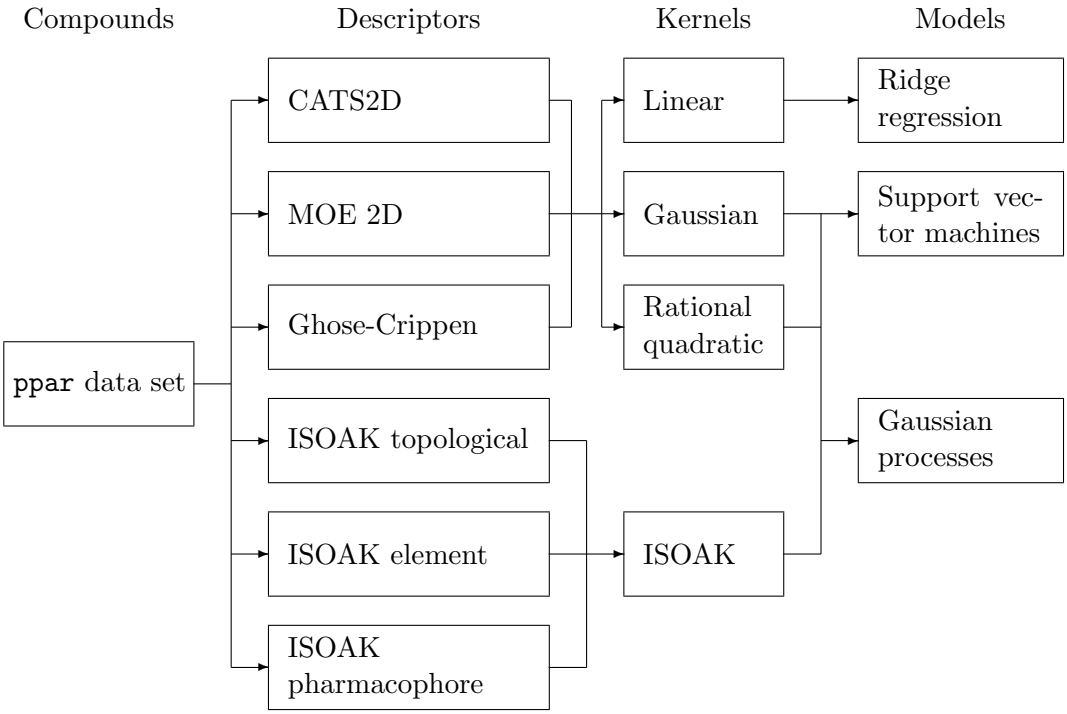


Figure 4.10 Diagram of the used descriptors, kernels, and models.

Models 10, 11, and 12 are Gaussian process models using different annotations of the structure graph. They differ from models 7, 8, and 9 in that they use the ISOAK (Chapter 2) graph kernel. Models 13, 14, 15, and 16 are Gaussian process models based on combinations of vectorial descriptors and graph representations using multiple kernel learning. The structure graphs are un-annotated as element type and pharmacophore information is already contained in the vectorial descriptors. Model 13 uses ISOAK as well as a Gaussian kernel and a rational quadratic kernel, operating on the concatenated chemical descriptors. Model 15 uses one Gaussian kernel and one rational quadratic kernel on each chemical descriptor. Models 14 and 16 are like the previous two models, except that training compounds are weighted according to their target value.

Performance

Based on the results (Table 4.5), three models were selected for prospective virtual screening. Models 14 and 16 were selected for best performance as measured by MAE and FI_{20}, respectively. Model 7 was selected because it performed as well as model 14 while being markedly simpler.

y-scrambling

As a negative control, we carried out the described validation procedures with randomly permuted target labels for the baseline models and the models selected for prospective screening (Table 4.6). The performance of linear ridge regression did not change, indicating that the linear model was not able to learn from the data in the beginning, whereas the performance of all other models decreased and the variance of their performance estimates increased, as should be.

4.3 Prospective screening

The three models 7, 14, and 15 with best retrospective performance were used to virtually screen a vendor library for novel agonists of PPARγ. From the results, 15 compounds were manually selected and tested in a transfection assay for activation of the receptor. The most PPARγ-selective compound was investigated further, including the elucidation of its conformation, and identified as a natural product.

4.3.1 Virtual screening

Models 7, 14, and 15 (Table 4.5), selected due to their retrospective performance, were trained on the whole `ppar` data set.

Screening library

The Asinex (`www.asinex.com`) gold and platinum libraries (version of November 2007, including updates) were joined into data set `asinex`. Preprocessing and descriptor computation were carried out as for retrospective evaluation (Subsection 4.2.2). Duplicates were removed, resulting in 360 150 compounds altogether.

Compound selection

The selected models were applied to the `asinex` data set, yielding an affinity prediction and confidence estimate for each compound. The data set was ranked according to each

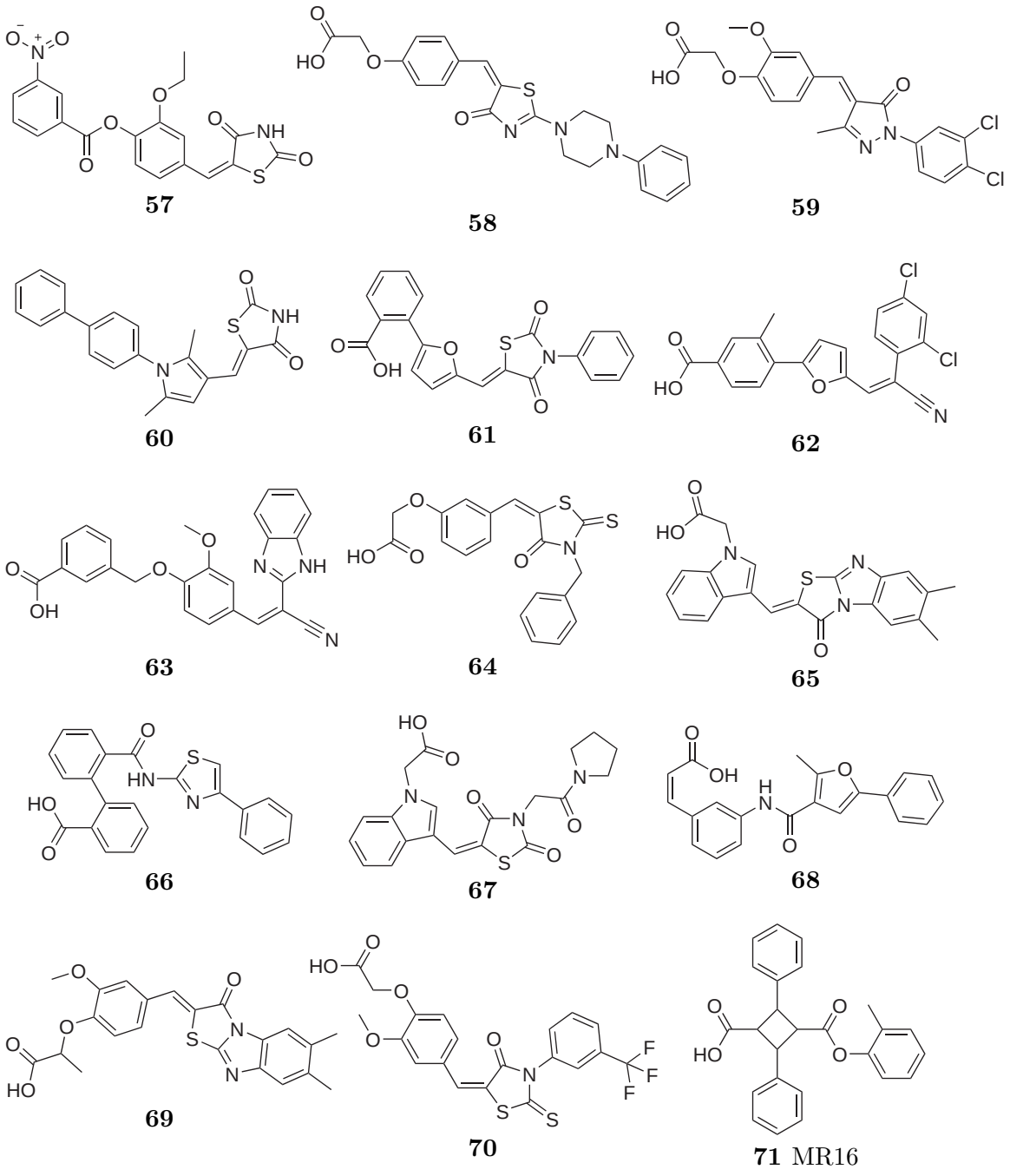
Table 4.5 Retrospective performance of two baseline models (linear ridge regression, support vector machines) and various Gaussian process models. Model names have the form descriptor / kernel, where + indicates combinations of kernels using multiple kernel learning. Performance of the best model in each block is in bold face. Stars ⋆ indicate models selected for prospective screening. # = model number, MAE = mean absolute error, FI_{20} = fraction of inactives among the 20 top-ranked compounds, CATS2D = chemically advanced template search descriptor, MOE 2D = molecular operating environment 2D descriptors, all = CATS2D, MOE 2D, and Ghose-Crippen descriptors together, allmkl = as all, but with individual kernel on each descriptor using multiple kernel learning, topo = unannotated structure graph, elem = structure graph with element and bond type annotation, ppp = structure graph with potential pharmacophore point annotation, RBF = radial basis function kernel (Gaussian kernel), RQ = rational quadratic kernel, ISOAK = iterative similarity optimal assignment kernel, W = compounds weighted by activity.

#	Model	MAE	FI_{20}	⋆
	Linear ridge regression			
1	CATS2D / none	1.70 ± 0.14	0.80 ± 0.08	
2	MOE 2D / none	$\mathbf{1.45} \pm 0.04$	$\mathbf{0.78} \pm 0.05$	
3	Ghose-Crippen / none	1.70 ± 0.08	0.79 ± 0.04	
	Support vector machines			
4	CATS2D / RBF	$\mathbf{0.68} \pm 0.06$	0.33 ± 0.08	
5	MOE 2D / RBF	0.69 ± 0.08	$\mathbf{0.29} \pm 0.14$	
6	Ghose-Crippen / RBF	0.86 ± 0.12	0.41 ± 0.09	
	Gaussian process regression			
7	CATS2D / RBF + RQ	$\mathbf{0.66} \pm 0.09$	0.27 ± 0.14	⋆
8	MOE 2D / RBF + RQ	0.76 ± 0.06	0.25 ± 0.12	
9	Ghose-Crippen / RBF + RQ	0.86 ± 0.07	0.33 ± 0.12	
10	topo / ISOAK	0.68 ± 0.06	0.33 ± 0.15	
11	elem / ISOAK	0.74 ± 0.06	0.32 ± 0.14	
12	ppp / ISOAK	0.70 ± 0.06	0.38 ± 0.09	
13	topo / ISOAK + all / RBF + all / RQ	0.67 ± 0.08	0.31 ± 0.14	
14	topo / ISOAK + all / RBF + all / RQ + W	$\mathbf{0.66} \pm 0.07$	0.32 ± 0.15	⋆
15	topo / ISOAK + allmkl / RBF + allmkl / RQ	0.70 ± 0.11	$\mathbf{0.21} \pm 0.09$	⋆
16	topo / ISOAK + allmkl / RBF + allmkl / RQ + W	0.71 ± 0.12	0.26 ± 0.12	

model by the $\hat{\mu} - \hat{\sigma}^2$ statistic used in retrospective evaluation. A filter was used to ensure that all compounds had either a carboxylic, a tetrazole, or, a thiazolidinedione group.[16]

From the 30 top-ranked compounds of each model (Schemes A.1, A.2, and A.3), 15 compounds were manually selected ("cherry-picked") by a panel of human experts, based on presumed activity and novelty of scaffold (Scheme 4.8).

[16] Filtering was done using MOE (molecular operating environment, Chemical Computing Group, `www.chemcomp.com`) and the SMARTS strings C(=O)[OH], [#6]1[#7][#7][#7][#7]1, and C1C(=O)NC(=O)S1. SMARTS (SMILES arbitrary target specification; Daylight Chemical Information Systems, `www.daylight.com`) is an extension of SMILES (simplified molecular input line entry specification; Weininger, 1988; Weininger et al., 1989), a string representation of molecules, to patterns for substructure searching.



Scheme 4.8 Compounds selected for assay tests.

Table 4.6 Retrospective performance of baseline and selected models under y-scrambling. Model names and abbreviations as in Table 4.5.

#	Model	MAE	FI_{20}
Linear ridge regression			
1	CATS2D / none	1.70 ± 0.14	0.80 ± 0.08
2	MOE 2D / none	1.45 ± 0.04	0.78 ± 0.05
3	Ghose-Crippen / none	1.70 ± 0.08	0.79 ± 0.04
Support vector machines			
4	CATS2D / RBF	1.09 ± 0.08	0.66 ± 0.21
5	MOE 2D / RBF	1.10 ± 0.10	0.68 ± 0.24
6	Ghose-Crippen / RBF	1.12 ± 0.06	0.64 ± 0.20
Gaussian process regression			
7	CATS2D / RBF + RQ	1.08 ± 0.02	0.57 ± 0.17
14	topo / ISOAK + all / RBF + all / RQ + W	1.08 ± 0.02	0.70 ± 0.22
15	topo / ISOAK + allmkl / RBF + allmkl / RQ	1.11 ± 0.06	0.65 ± 0.12

4.3.2 Transactivation assay

The compounds in Scheme 4.8 were tested in an in vitro cell-based transactivation assay. We briefly summarize the procedure; the assay was established and validated by Rau et al. (2006), and is based on work by Fu et al. (2003) and Takamura et al. (2004).

Idea

Cultured mammalian cells, here the immortalized simian kidney cell line Cos7, are co-transfected with a PPAR expression plasmid and a PPAR response element-linked reporter plasmid, in our case encoding for luciferase, a bioluminescent enzyme. After transfection, the cells express the PPAR in its inactive conformation. Upon treatment with a PPAR agonist, the receptor is activated and the reporter gene is expressed. The resulting luciferase-induced luminescence can be measured and compared to that of the positive control, here pioglitazone (Compound 35), resulting in a measure of relative activation of PPAR.

Reagents and materials

Foetal calf serum, dimethyl sulfoxide (DMSO) solvent, and ethanol absolute were obtained from Sigma-Aldrich (`www.sigmaaldrich.com`), Dulbecco's modified Eagle's medium (DMEM) and Optimem^{TM} from Gibco (Carlsbad, California, USA); sodium pyruvate solution, glutamine, penicillin and streptomycin stock solutions from PAA Laboratories GmbH (Pasching, Austria); Lipofectamine^{TM} 2000 from Invitrogen (Carlsbad, California, USA); DualGlo^{TM} luciferase assay system from Promega (Madison, Wisconsin, USA).

Plasmids

The Gal4 fusion receptor plasmids pFA-CMV-*h*PPARα-LBD, pFA-CMV-*h*PPARβ/δ-LBD and pFA-CMV-*h*PPARγ-LBD, containing the respective hinge regions and the ligand binding domains, were constructed by integrating cDNA fragments obtained from polymerase chain reaction amplification of human monocytes, into the *Sma*I/*Xba*I sites of the pFA-CMV vector (Stratagene; La Jolla, California, USA). The cDNA fragments contained base pairs 499–1 407 (NM_005 036;[17] *h*PPARα), base pairs 412–1 323 (NM_006 238; *h*PPARβ/δ) and base pairs 610–1 518 (NM_015 869; *h*PPARγ). Frame and sequence of the fusion receptors were verified by sequencing. pFR-Luc (Stratagene, `www.stratagene.com`) was used as reporter plasmid and pRL-SV40 (Promega, `www.promega.com`) for normalization of transfection efficacy.

Cell culture and transfection

Cos7 cells (kindly provided by Dieter Steinhilber; University of Frankfurt, Germany) were cultured in DMEM high-glucose supplemented with 10 % foetal calf serum containing 100 U/mL penicillin, 100 μg/mL streptomycin, 2 mM glutamine and 1 mM sodium pyruvate at 37 °C and 10 % CO_2. Cells were seeded at 30 000 cells/well in a 96 wellplate. After 24 h transfection was carried out using Lipofectamine^{TM} 2000 according to manufacturer's protocol. Transfection mixes contained 0.8 μL LF2000, 280 ng pFR-Luc, 2 ng pRL-SV40, and 14 ng of the appropriate fusion receptor plasmid for each well. 4 h after transfection the medium was changed to DMEM without phenol red containing 100 U/mL penicillin, 100 μg/mL streptomycin, 2 mM glutamine, 1 mM sodium pyruvate, the appropriate concentration of the test substance and 0.1 % DMSO. In every single experiment each concentration was tested in triplicate wells. Cells were incubated overnight and assayed for reporter gene activity with the DualGlo^{TM} luciferase assay system. Luminescence of both luciferases was measured in GENiosPro Luminometer (Tecan; Zurich, Switzerland). Each experiment was repeated independently at least three times.

Calculations

Luciferase activity for all assays was corrected by subtracting background activity obtained from non-transfected controls. Relative light units were calculated by dividing firefly light units by renilla light units. Activation factors were determined by dividing mean values of relative light units for each concentration of the agonist by mean relative light values of the DMSO control. Relative activation was calculated by dividing the activation factors by the activation factor obtained with 1 μM pioglitazone (Compound 35), the positive control in each experiment. All data are presented as mean ± standard deviation of 3 independent experiments. EC_{50} values were calculated based on the mean of the relative activation for each tested concentration of at least 3 individual experiments. SigmaPlot (SPSS, version 2001, `www.spss.com`) was used to fit the four parameter (min, max, EC_{50}, s) logistic regression function

$$f(x) = \min + \frac{\max - \min}{1 + \left(\frac{x}{EC_{50}}\right)^{-s}}, \tag{4.17}$$

where min is the minimum activation, max is the maximum activation, EC_{50} is the half maximal effective concentration, and s is the slope parameter.

[17] NM_xxx xxx codes are GenBank (`www.ncbi.nlm.nih.gov/Genbank`) identifiers.

#	*h*PPAR Activation [%] at 10 μM		
	α	β/δ	γ
57	30.6 ± 16	i. a.	35.5 ± 11
58	16.6 ± 8	i. a.	i. a.
59	22.9 ± 1	i. a.	i. a.
60	$\mathbf{69.9 \pm 10}$	i. a.	i. a.
62	$\mathbf{92.2 \pm 17}$	i. a.	$\mathbf{123.6 \pm 12}$
63	30.0 ± 11	i. a.	i. a.
68	$\mathbf{34.1 \pm 8}$	i. a.	32.5 ± 4
71	i. a.	i. a.	$\mathbf{73.1 \pm 17}$

Table 4.7 Activation of human PPAR subtypes by the selected compounds at 10 μM concentration in dimethylsulfoxide. Compounds 61, 64, 65, 66, 67, 69, and 70 were inactive on all three subtypes (not shown). Entries for which EC_{50} values were determined are shown in bold face. PPAR = peroxisome proliferator activated receptor, *h*PPAR = human PPAR, # = compound number, i. a. = inactive.

4.3.3 Results

Of the 15 selected compounds, 7 showed no activity on PPARα, PPARβ/δ, and PPARγ at 10 μM concentration, and were not investigated further. Based on their relative activation values (Table 4.7), EC_{50} values were determined for 4 of the remaining compounds (Figure 4.11). In total, a partial PPARα agonist (Compound 68), a selective partial PPARα agonist (Compound 60), a dual PPARα/γ agonist (Compound 62), and, a selective full PPARγ agonist (Compound 71; MR16) were discovered, resulting in an overall hit rate of 27 %.

4.3.4 Compound MR16

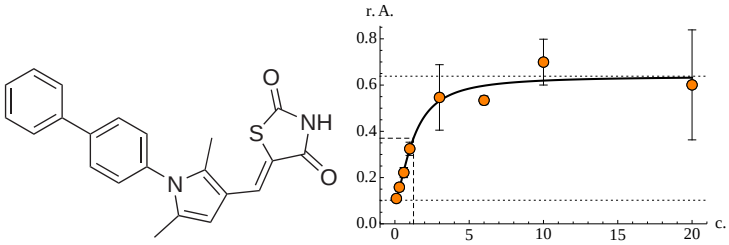
Compound 71 (2,4-diphenyl-3-(o-tolyloxycarbonyl)cyclobutane-carboxylic acid, MR16) is a selective full PPARγ-agonist with $EC_{50} = 10.03 \pm 0.2\,\mu$M, a maximum activation of 138 $\pm$ 2 % (Figure 4.11d), and an interesting cyclobutane-based scaffold. It showed no cytotoxicity in cell-based assay tests. Its presumed binding mode as determined by docking is shown in Figure 4.12.

Natural product

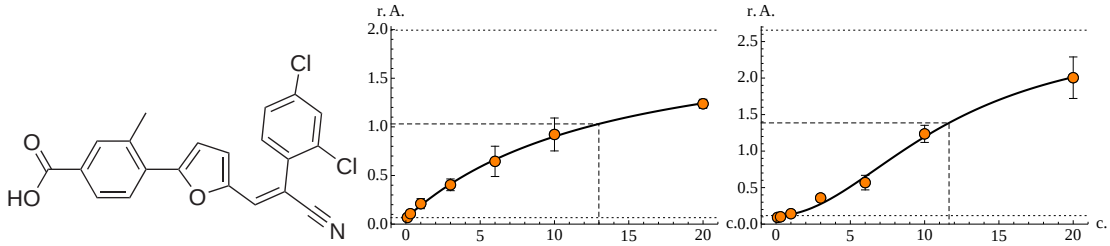
Compound MR16 is a truxillic acid derivative (Scheme 4.9). Truxillic acid (Compound 73) is synthesized in plant cell walls by photo-dimerization of trans-cinnamic acid (Compound 74; Bernstein and Quimby, 1943), as well as the latters hydroxy derivatives *p*-coumaric acid and ferulic acid (Compounds 75 and 76; Morrison III et al., 1992). Plant cell walls often contain phenolic acids such as (di)ferulic acid, *p*-coumaric acid, and truxillic acid. They are bound by ester linkage to the polysaccharide constituents of cell walls, especially of grasses (Krishnamurthy, 1999).

Stereochemistry

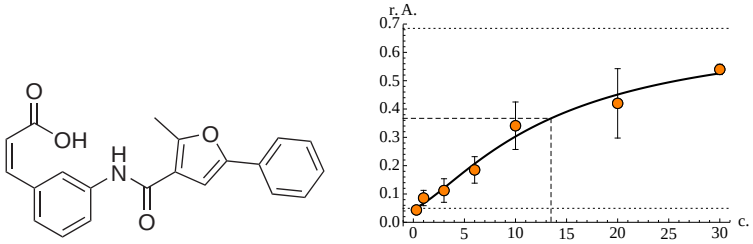
The stereochemical configuration of MR16 was determined by nuclear magnetic resonance spectroscopy (NMR; Slichter, 1990). Scheme 4.10 shows all possible configurations of MR16. Analysis of the ^{1}H-spectrum (Figure A.1) allows a first discrimination between the different configurations. The chemical shifts of the proton signals for the cyclobutane protons next to the phenyl rings would be identical for all cis-diphenyl isomers, yielding three peaks (1:2:1), whereas the trans-diphenyl isomers result in four peaks (1:1:1:1). The spectrum therefore rules out all cis-isomers.



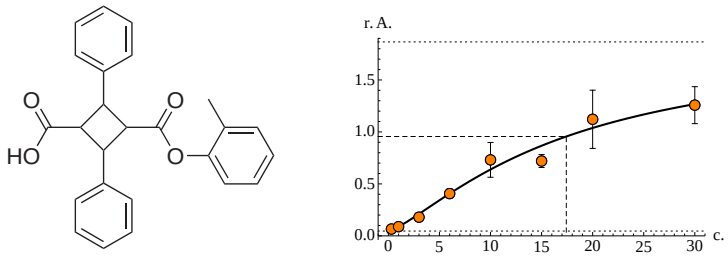
(a) Compound 60 (left) and dose-response curve for *h*PPARα activation (right; *h*PPARα EC_{50} = $1.25 \pm 0.37\,\mu$M, maximum activation of $65 \pm 5\,\%$). A thiazolidinedione-based selective partial Compound 60PPARα agonist.



(b) Compound 62 (left) and dose-response curves for *h*PPARα activation (middle; *h*PPARα EC_{50} = $12.98 \pm 4.21\,\mu$M, maximum activation of $200 \pm 32\,\%$) and *h*PPARγ activation (right; *h*PPARγ EC_{50} = $3.75 \pm 0.2\,\mu$M, maximum activation = $76 \pm 3\,\%$). A dual PPARα/γ agonist, with full Compound 62activity on PPARα and partial PPARγ agonism.



(c) Compound 68 (left) and dose-response curve for *h*PPARα (right; *h*PPARα EC_{50} = Compound 6813.48 $\pm$ 8.53 μM, maximum activation = $68 \pm 24\,\%$). A PPARα partial agonist.



(d) Compound 71 (left) and dose-response curve for *h*PPARγ (right; *h*PPARγ EC_{50} = Compound 7110.03 $\pm$ 0.2 μM, maximum activation = $138 \pm 2\,\%$). A full selective PPARγ agonist.

Figure 4.11 Summary of virtual screening hits. Out of 15 selected compounds, 4 were active on PPAR (27 % hit rate). Shown are assay measurements ($n \geq 3$, orange disks) with standard deviation (error bars), fitted dose-response curves (solid lines), minimum and maximum activation (dotted lines), and EC_{50} values (dashed lines). Note that plot axes have linear scale. r. A. = activation relative to control, c. = ligand concentration, PPAR = peroxisome proliferator-activated receptor, *h*PPAR = human PPAR, EC_{50} = half maximal effective concentration.

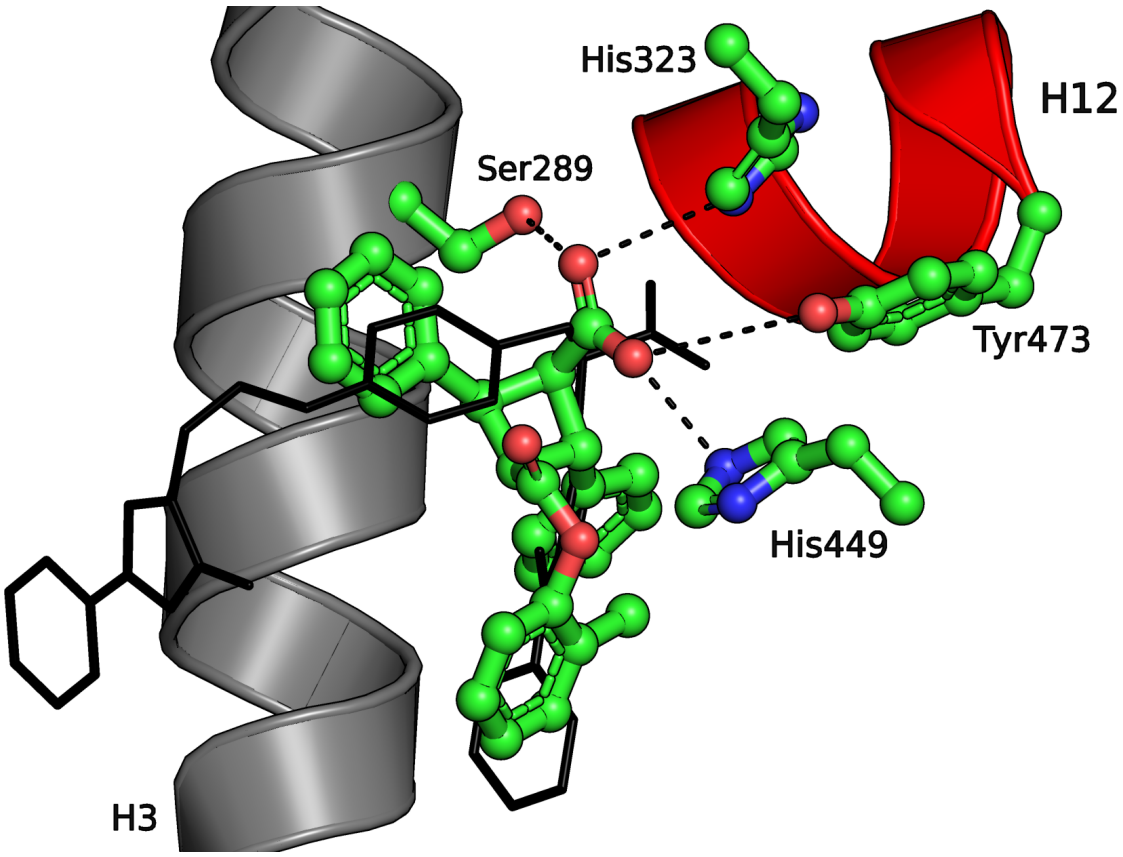


Figure 4.12 Binding mode of Compound MR16 for PPARγ, as obtained by the docking software GOLD (version 4.0.1, Cambridge crystallographic data centre, Cambridge, England). Shown are MR16 (green ball-and-stick model), helix 3 (gray cartoon model) and helix 12 (activation function 2, red cartoon model) of the PPARγ ligand binding domain (PDBid 1fm9; Gampe et al., 2000), four amino acids (green ball-and-stick models) and their interactions with the carboxylic acid group of MR16 (dashed black lines), as well as the binding mode of the PPARγ agonist farglitazar (black line model; Compound 36, Figure 4.3) for comparison. PPAR = peroxisome proliferator-activated receptor, PDBid = protein data bank identifier, H = helix, Ser = serine, His = histidine, Tyr = tyrosine.

To discriminate between the remaining two pairs, the ratio of the coupling constants and the nuclear Overhauser effect (Overhauser, 1953; Anet and Bourn, 1965) were used. Consider the neighborhood of the cyclobutane-phenyl protons in Compounds 77 and 78: One of them has two cis-neighboring protons, the other has two trans-neighboring protons. In Compounds 79 and 80, both protons have one cis- and one trans-neighboring proton. The coupling constants of the two protons have different values ($\Delta J = 0.7\,\mathrm{Hz}$), and therefore identify the configuration of Compound MR16 to be the one given by Compounds 77 and 78. Differential nuclear Overhauser effect spectra (Figures A.2, A.3) and a rotating frame nuclear Overhauser effect spectrum (Figure A.4) confirmed this.

In summary, nuclear magnetic resonance spectroscopy identified a racemate of Compounds 77 and 78 as the investigated configuration of Compound MR16.

4.4 Conclusions

Kernel-based learning approaches, in particular graph kernels and Gaussian processes, were successfully applied to ligand-based virtual screening, resulting in the discovery of MR16, a selective full PPARγ agonist with novel scaffold.

4.4.1 Summary

The peroxisome proliferator-activated receptor (PPAR) is a nuclear transcription factor involved in the regulation of lipid and glucose metabolism that plays a crucial role in the development of diseases like type 2 diabetes and dyslipidemia. We established a Gaussian process regression model for PPARγ agonism using a public data set of 144 compounds annotated with pK_i-values measured in scintillation proximity assays. The compounds were represented using a combination of chemical descriptors and the iterative similarity optimal assignment kernel (Chapter 2) via multiple kernel learning. Screening of a large ($3.6 \cdot 10^5$ compounds) vendor library and subsequent testing of 15 selected compounds in a cell-based transactivation assay resulted in 4 active compounds. One compound, a natural product with cyclobutane scaffold, is a full selective PPARγ-agonist ($\mathrm{EC}_{50} = 10 \pm 0.2\,\mu\mathrm{M}$, inactive on PPAR$\alpha$ and PPARβ/δ at $10\,\mu\mathrm{M}$). Our study delivered a novel agonist, de-orphanized a natural bioactive product, and, hints at the natural product origins of pharmacophore patterns in synthetic ligands.

4.4.2 Retrospective evaluation

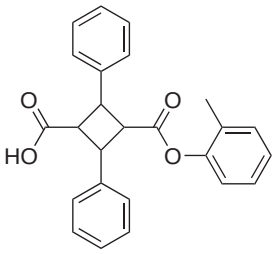
We discuss retrospective evaluation results (Tables 4.5 and 4.6).

Non-linear nature of PPARγ agonism

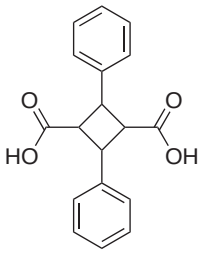
The performance of ridge regression models with different descriptors, which were used as linear baseline models, did not change under y-scrambling, whereas the performance of all other methods did. Together with the good performance (Table 4.5) of non-linear models with the same descriptors, this implies that PPARγ binding is a non-linear process that linear models are not able to capture.

Suitability of Ghose-Crippen descriptors

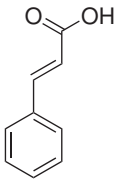
In all non-linear models explicitly based on descriptors (models 4–9), Ghose-Crippen fragment descriptors perform markedly worse than CATS2D and MOE 2D descriptors, with an average RMSE difference of 0.16, and 0.09 for FI_{20}. Ghose-Crippen fragment descriptors seem not suited to PPAR activity estimation.



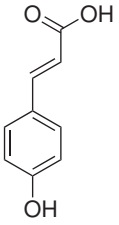
72 Compound MR16



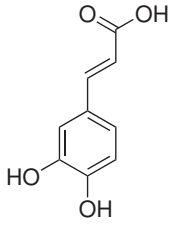
73 Truxillic acid



74 trans-cinnamic acid

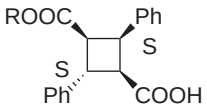


75 *p*-coumaric acid

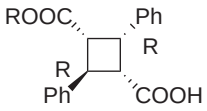


76 Ferulic acid

Scheme 4.9 Structures of MR16, truxillic acid, and monomer components. MR16 is a truxillic acid derivative, which in turn can be synthesized from Compounds 74, 75, and 76. *p*-coumaric acid and ferulic acid are hydroxy derivatives of trans-cinnamic acid.

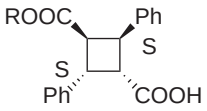


77

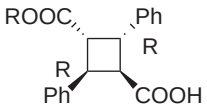


78

1:1:1:1, trans-diphenyl, racemic

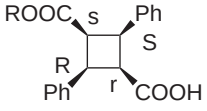


79

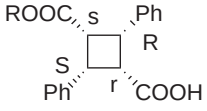


80

1:1:1:1, trans-diphenyl, racemic

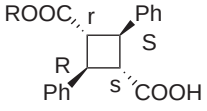


81

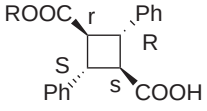


82

1:2:1, cis-diphenyl, meso compound

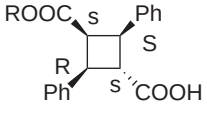


83

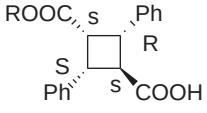


84

1:2:1, cis-diphenyl, meso compound

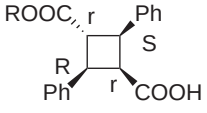


85

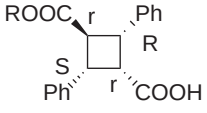


86

1:2:1, cis-diphenyl, meso-compound



87



88

1:2:1, cis-diphenyl, meso-compound

Scheme 4.10 Possible absolute configurations of Compound MR16. The labeling gives the number of peaks of the cyclobutane protons in the ^{1}H-NMR spectrum (Figure A.1), as well as configuration isomerism.

Compound weighting

Models 13 and 14, as well as models 15 and 16, differ only in that models 14 and 16 weight compounds by their activity. Models 13 and 14 perform equally well (differences in RMSE and FI_{20} were less than 0.01); models 15 and 16 differ by 0.01 in RMSE, while for FI_{20}, performance is worse for the weighted model (by 0.05, within the standard deviation of 0.09). Focusing on highly active compounds did not improve predictive performance, not even for other highly active compounds, as indicated by FI_{20}. This might indicate that in this study, the advantage of improved generalization capability gained by consideration of all samples outweighed the specificity advantage gained by concentrating on a small sample group.

Multiple kernel learning

Using separate kernels for the vectorial descriptors (models 15 and 16 versus models 13 and 14) slightly worsened mean absolute error, but improved (lessened) the fraction of inactives in the 20 top-ranked compounds. Descriptors were standardized, i. e., using the same length scale, which allowed using a single kernel for several descriptors in the first place. A tentative explanation is that a single kernel might generalize better, thereby improving MAE, while separate kernels might fit specific compound groups more closely, possibly leading to improved FI_{20}.

4.4.3 Prospective screening

We discuss topics related to the discovered hits.

Activity on PPARα and PPARβ/δ

All selected compounds were predicted by our models to be active on PPARγ. Of the four hits, one was selective for PPARγ, one was active on PPARα and PPARγ, and two were selective for PPARα. No compound was active on PPARβ/δ. We attribute this to properties of the binding pockets, and the fuzzy nature of ligand-based virtual screening.

The binding pocket of PPARβ/δ is the smallest of the three, and can not accommodate many of the other two subtypes ligands for sterical reasons. The pocket of PPARα is the largest one, and more similar to the one of PPARγ. Together, this might explain why compounds were inactive on PPARβ/δ.

Subtype selectivity is often achieved via differences in the left proximal and distal subpockets, which are not necessarily directly relevant to activity. The model was trained on activity alone, with no consideration of selectivity. This might account for the compound active on PPARα and PPARγ.

Ligand-based virtual screening is based on the similarity principle, and therefore inherently fuzzy in the sense that changes in activity induced by structural modifications can be estimated only with limited accuracy. From another point of view, the complexity of the modeled biochemical process, the high (measurement) noise levels, and the relative scarcity of samples render activity estimation difficult. This effect is particularly pronounced when, as in this study, scaffold hopping is involved, which by definition leaves or at least strains the domain of applicability. The inactive compounds, as well as the presence of compounds selective for PPARα might be attributed to these aspects.

The above considerations offer a possible explanation of the observed facts; however, due to the small sample size ($n = 15$), no definite conclusions can be drawn.

De-orphanization of a natural product

The identification of Compound MR16 as a selective full PPARγ agonist de-orphanizes truxillic acid derivatives in the sense that it provides a possible explanation for the empirically known anti-inflammatory effects of this compound class. Traditional medicines, e. g., herbal plant extracts, often rely exclusively on empirically observed effects. The identification of the molecular mechanisms behind such drugs benefits both their therapeutic use and the development of novel lead compounds based on natural substances.

Natural product origins

Known ligands are the result of incremental drug research processes, with origins in (Ji et al., 2009) and continuing input from (Newman and Cragg, 2007) natural products. Consequently, pharmacophoric patterns of natural products are encoded in synthetic drugs. We believe this to be the underlying fact that allowed us to discover the pharmaceutical activity of a natural product on the basis of mostly synthetic compounds.

4.4.4 Outlook

We propose ideas for further research.

- *Utility of graph kernels:* The present study adds to the growing body of research (Subsection 2.2.7) where graph kernels improve predictive performance in bio- and cheminformatics applications. Evidence is, however, still preliminary, and further systematic studies of the utility of graph kernels, in particular with regard to ligand-based virtual screening and quantitative structure-property relationships, are warranted.

- *Reliability of Gaussian process confidence estimates:* Gaussian process regression provides an implicit domain of applicability in the form of confidence estimates, or predictive variance. Advantages of these estimates include their analytical form, and that they constitute an implicit and integral part of the algorithm. Considering that their utility for quantitative structure-property relationships has already been demonstrated (Table 4.4), Gaussian processes should be of particular interest for reliable toxicity prediction within the European Unions registration, evaluation, authorization and restriction of chemicals legislative (REACH; regulation EC 1907/2006). In the next 5–7 years, the evaluation of large numbers of chemicals with production volumes of less than 1000 tons / year will have to rely increasingly on computational methods. To further qualify Gaussian process regression for regulatory-purpose toxicity prediction, quantitative validations of the reliability of their confidence estimates in this context would be beneficial.

- *Cynodon dactylon:* For some herbals there is evidence from animal studies that they improve diabetic disorders, but no molecular mechanism is known. One example for such a plant is the the grass *Cynodon dactylon*, whose extract was shown to have anti-diabetic potential by lowering blood glucose levels and additionally improving hyperlipidemia in rats (Singh et al., 2007). *Cynodon dactylon* contains several flavonoids and sterols which could potentially cause these effects, but it also contains a high amount of substituted truxillic acids in its cell walls (Hartley et al., 1990). Truxillic acid shows anti-inflammatory activity in mice formalin tests (significantly reduced pawn licking time after formalin injection; its dimeric structure is thought to be relevant for this effect; Chi et al., 2005), and PPARγ is involved in inflammatory response regulation (p. 144; compare Compounds 12 and 21). We are currently investigating whether the anti-diabetic effects of *Cynodon dactylon* are mediated by truxillic acids or their derivatives such as Compound MR16.

References

Richard Alexander, Marcus Wright, Michael Gorczynski, Pamela Smitherman, Taro Akiyama, Harold Wood, Joel Berger, Bruce King, Charles Morrow. *Differential potencies of naturally occurring regioisomers of nitrolinoleic acid in PPARγ activation.* Biochemistry, 48(2): 492–498, 2009.

Ferhana Ali, Karine Egan, Garret FitzGerald, Béatrice Desvergne, Walter Wahli, David Bishop-Bailey, Timothy Warner, Jane Mitchell. *Role of prostacyclin versus peroxisome proliferator-activated receptor β receptors in prostacyclin sensing by lung fibroblasts.* American Journal of Respiratory Cell and Molecular Biology, 34(2): 242–246, 2006.

Tamara Allen, Fang Zhang, Shonna Moodie, Edward Clemens, Aaron Smith, Francine Gregoire, Andrea Bell, George Muscat, Thomas Gustafson. *Halofenate is a selective peroxisome proliferator-activated receptor γ modulator with antidiabetic activity.* Diabetes, 55(9): 2523–2533, 2006.

Frank Anet, Tony Bourn. *Nuclear magnetic resonance spectral assignments from nuclear Overhauser effects.* Journal of the American Chemical Society, 87(22): 5250–5251, 1965.

Dean Artis, Jack Lin, Chao Zhang, Weiru Wang, Upasana Mehra, Mylene Perreault, David Erbe, Heike Krupka, Bruce England, James Arnold, Alexander Plotnikov, Adhirai Marimuthu, Hoa Nguyen, Sarah Will, Maxime Signaevsky, John Kral, John Cantwell, Calvin Settachatgull, Douglas Yan, Daniel Fong, Angela Oh, Shenghua Shi, Patrick Womack, Benjamin Powell, Gaston Habets, Brian West, Kam Zhang, Michael Milburn, George Vlasuk, Peter Hirth, Keith Nolop, Gideon Bollag, Prabha Ibrahim, James Tobin. *Scaffold-based discovery of indeglitazar, a PPAR pan-active anti-diabetic agent.* Proceedings of the National Academy of Sciences of the United States of America, 106(1): 262–267, 2009.

Paul Baker, Yiming Lin, Francisco Schopfer, Steven Woodcock, Alison Groeger, Carlos Batthyany, Scott Sweeney, Marshall Long, Karen Iles, Laura Baker, Bruce Branchaud, Yuqing Chen, Bruce Freeman. *Fatty acid transduction of nitric oxide signaling. Multiple nitrated unsaturated fatty acid derivatives exist in human blood and urine and serve as endogenous peroxisome proliferator-activated receptor ligands.* Journal of Biological Chemistry, 280(51): 42 464–42 475, 2005.

Carine Ekambomé Basséne, Franck Suzenet, Nathalie Hennuyer, Bart Staels, Daniel-Henri Caignard, Catherine Dacquet, Pierre Renard, Gérald Guillaumet. *Studies towards the conception of new selective PPARβ/δ ligands.* Bioorganic & Medicinal Chemistry Letters, 16(17): 4528–4532, 2006.

Brock Beamer, Carlo Negri, Chung-Jen Yen, Oksana Gavrilova, John Rumberger, Michael Durcan, David Yarnall, Anita Hawkins, Constance Griffin, Daniel Burns, Jesse Roth, Marc Reitman, Alan Shuldiner. *Chromosomal localization and partial genomic structure of the human peroxisome proliferator activated receptor-gamma (hPPARγ) gene.* Biochemical and Biophysical Research Communications, 233(3): 756–759, 1997.

Joel Berger, Mark Leibowitz, Thomas Doebber, Alex Elbrecht, Bei Zhang, Gaochou Zhou, Chhabi Biswas, Catherine Cullinan, Nancy Hayes, Ying Li, Michael Tanen, John Ventre, Margaret Wu, Gregory Berger, Ralph Mosley, Robert Marquis, Conrad Santini, Soumya Sahoo, Richard Tolman, Roy Smith, David Moller. *Novel peroxisome proliferator-activated receptor (PPAR) γ and PPARδ ligands produce distinct biological effects.* Journal of Biological Chemistry, 274(10): 6718–6725, 1999.

Helen Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady Bhat, Helge Weissig, Ilya Shindyalov, Philip Bourne. *The protein data bank.* Nucleic Acids Research, 28(1): 235–242, 2000.

Herbert Bernstein, William Quimby. *The photochemical dimerization of trans-cinnamic acid.* Journal of the American Chemical Society, 65(10): 1845–1846, 1943.

William Bourguet, Pierre Germain, Hinrich Gronemeyer. *Nuclear receptor ligand-binding domains: Three-dimensional structures, molecular interactions and pharmacological implications.* Trends in Pharmacological Sciences, 21(10): 381–388, 2000.

Frank Burden. *Quantitative structure-activity relationship studies using Gaussian processes.* Journal of Chemical Information and Computer Sciences, 41(3): 830–835, 2001.

Andrew Burdick, Dae Kim, Marjorie Peraza, Frank Gonzalez, Jeffrey Peters. *The role of peroxisome proliferator-activated receptor-β/δ in epithelial cell growth and differentiation.* Cellular Signalling, 18(1): 9–20, 2006.

Katherine Burns, John Vanden Heuvel. *Modulation of PPAR activity via phosphorylation.* Biochimica et Biophysica Acta - Molecular and Cell Biology of Lipids, 1771(8): 952–960, 2007.

Vikas Chandra, Pengxiang Huang, Yoshitomo Hamuro, Srilatha Raghuram, Yongjun Wang, Thomas Burris, Fraydoon Rastinejad. *Structure of the intact PPAR-γ-RXR-α nuclear receptor complex on DNA.* Nature, 456(7220): 350–357, 2008.

Rajnish Chaturvedi, Flint Beal. *PPAR: A therapeutic target in Parkinson's disease.* Journal of Neurochemistry, 106(2): 506–518, 2008.

Tao Chen, Elaine Martin. *Bayesian linear regression and variable selection for spectroscopic calibration.* Analytica Chimica Acta, 631(1): 13–21, 2008.

Peter Cheng, Ranjan Mukherjee. *PPARs as targets for metabolic and cardiovascular diseases.* Mini Reviews in Medicinal Chemistry, 5(8): 741–753, 2005.

Yu-Ming Chi, Motoyuki Nakamura, Toyokichi Yoshizawa, Xi-Ying Zhao, Wen-Mei Yan, Fumio Hashimoto, Junei Kinjo, Toshihiro Nohara, Shinobu Sakurada. *Anti-inflammatory activities of α-truxillic acid derivatives and their monomer components.* Biological & Pharmaceutical Bulletin, 28(9): 1776–1778, 2005.

Min-Chul Cho, Kyoung Lee, Sang-Gi Paik, Do-Young Yoon. *Peroxisome proliferators-activated receptor (PPAR) modulators and metabolic disorders.* PPAR Research, article 679137, 2008.

Han Chung Chong, Ming Jie Tan, Virginie Philippe, Siew Hwey Tan, Chek Kun Tan, Chee Wai Ku, Yan Yih Goh, Walter Wahli, Liliane Michalik, Nguan Soon Tan. *Regulation of epithelialmesenchymal IL-1 signaling by PPARβ/δ is essential for skin homeostasis and wound healing.* The Journal of Cell Biology, 184(6): 817–831, 2009.

Arindam Choudhury, Prasanth Nair, Andy Keane. *A data parallel approach for large-scale Gaussian process modeling.* In *Proceedings of the 2nd SIAM International Conference on Data Mining (SDM 2002), Arlington, Virginia, USA, April 11–13.* Society for Industrial and Applied Mathematics, 2002.

Christopher Corton, Paula Lapinskas. *Peroxisome proliferator-activated receptors: Mediators of phthalate ester-induced effects in the male reproductive tract?* Toxicological Sciences, 83(1): 4–17, 2005.

Juraj Culman, Yi Zhao, Peter Gohlke, Thomas Herdegen. *PPAR-γ: Therapeutic target for ischemic stroke.* Trends in Pharmacological Sciences, 28(5): 244–249, 2007.

Salvatore Cuzzocrea, Barbara Pisano, Laura Dugo, Angela Ianaro, Pasquale Maffia, Nimesh Patel, Rosanna Di Paola, Armando Ialenti, Tiziana Genovese, Prabal Chatterjee, Massimo Di Rosa, Achille Caputi, Christoph Thiemermann. *Rosiglitazone, a ligand of the peroxisome proliferator-activated receptor-γ, reduces acute inflammation.* European Journal of Pharmacology, 483(1): 79–93, 2004.

Tatjana Degenhardt, Anna Saramäki, Marjo Malinen, Markus Rieck, Sami Väisänen, Anne Huotari, Karl-Heinz Herzig, Rolf Müller, Carsten Carlberg. *Three members of the human pyruvate dehydrogenase kinase gene family are direct targets of the peroxisome proliferator-activated receptor β/δ.* Journal of Molecular Biology, 372(2): 341–355, 2007.

Philippe Delerive, Jean-Charles Fruchart, Bart Staels. *Peroxisome proliferator-activated receptors in inflammation control.* Journal of Endocrinology, 169: 453–459, 2001.

Béatrice Desvergne, Walter Wahli. *Peroxisome proliferator-activated receptors: Nuclear control of metabolism.* Endocrine Reviews, 20(5): 649–688, 1999.

Pratik Devasthale, Sean Chen, Yoon Jeon, Fucheng Qu, Chunning Shao, Wei Wang, Hao Zhang, Michael Cap, Dennis Farrelly, Rajasree Golla, Gary Grover, Thomas Harrity, Zhengping Ma, Lisa Moore, Jimmy Ren, Ramakrishna Seethala, Lin Cheng, Paul Sleph, Wei Sun, Aaron Tieman, John Wetterau, Arthur Doweyko, Gamini Chandrasena, Shu Chang, Griffith Humphreys, Vito Sasseville, Scott Biller, Denis Ryono, Fred Selan, Narayanan Hariharan, Peter Cheng. *Design and synthesis of N-[(4-methoxyphenoxy)carbonyl]-N-[[4-[2-(5-methyl-2-phenyl-4-oxazolyl)ethoxy]phenyl]methyl]glycine [muraglitazar / BMS-298585], a novel peroxisome proliferator-activated receptor α/γ dual agonist with efficacious glucose and lipid-lowering activities.* Journal of Medicinal Chemistry, 48(6): 2248–2250, 2005.

Joseph Doob. *The elementary Gaussian processes.* Annals of Mathematical Statistics, 15(3): 229–282, 1944.

Paul Drew, Jihong Xu, Michael Racke. *PPAR-γ: Therapeutic potential for multiple sclerosis.* PPAR Research, article 627463, 2008.

Christine Dreyer, Grigorios Krey, Hansjörg Keller, Francoise Givel, Gerd Helftenbein, Walter Wahli. *Control of the peroxisomal β-oxidation pathway by a novel family of nuclear hormone receptors.* Cell, 68(5): 879–887, 1992.

Laurent Dubuquoy, Christel Rousseaux, Xavier Thuru, Laurent Peyrin-Biroulet, Olivier Romano, Philippe Chavatte, Mathias Chamaillard, Pierre Desreumaux. *PPARγ as a new therapeutic target in inflammatory bowel diseases.* Gut, 55: 1341–1349, 2006.

Caroline Duval, Michael Müller, Sander Kersten. *PPARα and dyslipidemia.* Biochimica et Biophysica Acta, 1771(8): 961–971, 2007.

David Enot, Régis Gautier, Jean Le Marouille. *Gaussian Process: An efficient technique to solve quantitative structure-property relationship problems.* SAR and QSAR in Environmental Research, 12(5): 461–469, 2001.

Joseph Evans, Jack Lin, Ira Goldfine. *Novel approach to treat insulin resistance, type 2 diabetes, and the metabolic syndrome: Simultaneous activation of PPARα, PPARγ, and PPARδ.* Current Diabetes Reviews, 1(3): 299–307, 2005.

Lluis Fajas, Didier Auboeuf, Eric Raspé, Kristina Schoonjans, Anne-Marie Lefebvre, Regis Saladin, Jamilla Najib, Martine Laville, Jean-Charles Fruchart, Samir Deeb, Antonio Vidal-Puig, Jeffrey Flier, Michael Briggs, Bart Staels, Hubert Vidal, Johan Auwerx. *The organization, promoter*

analysis, and expression of the human PPAR gene. Journal of Biological Chemistry, 272(30): 18 779–18 789, 1997.

Emilia Falcettia, David Flavella, Bart Staels, Andrew Tinker, Sheila Haworth, Lucie Clapp. *IP receptor-dependent activation of PPARγ by stable prostacyclin analogues.* Biochemical and Biophysical Research Communications, 360(4): 821–827, 2007.

Uli Fechner, Lutz Franke, Steffen Renner, Petra Schneider, Gisbert Schneider. *Comparison of correlation vector methods for ligand-based similarity searching.* Journal of Computer-Aided Molecular Design, 17(10): 687–698, 2003.

Ana Fernandez. *PPARs as targets for the modulation of cardiovascular risk factors associated with the metabolic syndrome.* Current Opinion in Investigational Drugs, 5(9): 936–940, 2004.

Barry Forman, Peter Tontonoz, Jasmine Chen, Regina Brun, Bruce Spiegelman, Ronald Evans. *15-deoxy-$\delta^{12,14}$-prostaglandin J_2 is a ligand for the adipocyte determination factor PPARγ.* Cell, 83(5): 803–812, 1995.

Jin Fu, Silvana Gaetani, Fariba Oveisi, Jesse Lo Verme, Antonia Serrano, Fernando Rodríguez de Fonseca, Anja Rosengarth, Hartmut Luecke, Barbara Di Giacomo, Giorgio Tarzia, Daniele Piomelli. *Oleylethanolamide regulates feeding and body weight through activation of the nuclear receptor PPAR-α.* Nature, 425(6953): 90–93, 2003.

Stewart Fyffe, Magnus Alphey, Lori Buetow, Terry Smith, Michael Ferguson, Morten Sørensen, Fredrik Björkling, William Hunter. *Recombinant human PPAR-β/δ ligand-binding domain is locked in an activated conformation by endogenous fatty acids.* Journal of Molecular Biology, 356(4): 1005–1013, 2006.

Robert Gampe, Valerie Montana, Millard Lambert, Aaron Miller, Randy Bledsoe, Michael Milburn, Steven Kliewer, Timothy Willson, Eric Xu. *Asymmetry in the PPARγ/RXRα crystal structure reveals the molecular basis of heterodimerization among nuclear receptors.* Molecular Cell, 5(3): 545–555, 2000.

Osman Gani, Ingebrigt Sylte. *Ligand-induced stabilization and activation of peroxisome proliferator-activated receptor γ.* Chemical Biology & Drug Design, 72(1): 50–57, 2008.

Pierre Germain, Bart Staels, Catherine Dacquet, Michael Spedding, Vincent Laudet. *Overview of nomenclature of nuclear receptors.* Pharmacological Reviews, 58(4): 685–704, 2006.

Philippe Gervois, Robert Kleemann, Antoine Pilon, Frédéric Percevault, Wolfgang Koenig, Bart Staels, Teake Kooistra. *Global suppression of IL-6-induced acute phase response gene expression after chronic in vivo treatment with the peroxisome proliferator-activated receptor-α activator fenofibrate.* Journal of Biological Chemistry, 279(16): 16 154–16 160, 2004.

Arup Ghose, Gordon Crippen. *Atomic physicochemical parameters for three-dimensional structure-directed quantitative structure-activity relationships 1. Partition coefficients as a measure of hydrophobicity.* Journal of Computational Chemistry, 7(4): 565–577, 1986.

Arup Ghose, Gordon Crippen. *Atomic physicochemical parameters for three-dimensional structure-directed quantitative structure-activity relationships 2. Modeling dispersive and hydrophobic interactions.* Journal of Chemical Information and Computer Sciences, 27(1): 21–35, 1987.

Arup Ghose, Avis Pritchett, Gordon Crippen. *Atomic physicochemical parameters for three-dimensional structure-directed quantitative structure-activity relationships 3. Modeling hydrophobic interactions.* Journal of Computational Chemistry, 9(1): 80–90, 1988.

Kristina Grabowski, Karl-Heinz Baringhaus, Gisbert Schneider. *Scaffold diversity of natural products: Inspiration for combinatorial library design.* Natural Product Reports, 25(5): 892–904, 2008.

Tracey Graham, Claudette Mookherjee, Keith Suckling, Colin Palmer, Lisa Patel. *The PPARδ agonist GW0742X reduces atherosclerosis in LDLR$^{-/-}$ mice.* Atherosclerosis, 181(1): 29–37, 2005a.

Tracey Graham, Claudette Mookherjee, Keith Suckling, Colin Palmer, Lisa Patel. *Corrigendum to "The PPARδ agonist GW0742X reduces atherosclerosis in LDLR$^{-/-}$ mice".* Atherosclerosis, 182(2): 381, 2005b.

Paul Grimaldi. *Regulatory functions of PPARβ in metabolism: Implications for the treatment of metabolic syndrome.* Biochimica et Biophysica Acta - Molecular and Cell Biology of Lipids, 1771(8): 983–990, 2007.

Hinrich Gronemeyer, Jan Åke Gustafsson, Vincent Laudet. *Principles for modulation of the nuclear receptor superfamily.* Nature Reviews Drug Discovery, 3(11): 950–964, 2004.

John Guiver, Edward Snelson. *Learning to rank with SoftRank and Gaussian processes.* In *Proceedings of the 31st Annual International ACM Conference of the special interest group on information retrieval (SIGIR 2008), Singapore, July 20–24*, 2008.

Rajnish Gupta, Jian Tan, Wade Krause, Mark Geraci, Timothy Willson, Sudhansu Dey, Raymond DuBois. *Prostacyclin-mediated activation of peroxisome proliferator-activated receptor δ in colorectal cancer.* Proceedings of the National Academy of Sciences of the United States of America, 97(24): 13 275–13 280, 2000.

Kyu Lee Han, Joo Sun Choi, Jae Young Lee, Jihyun Song, Myung Kuk Joe, Myeong Ho Jung,

Jae-Kwan Hwang. *Therapeutic potential of peroxisome proliferators-activated receptor-α/γ dual agonist with alleviation of endoplasmic reticulum stress for the treatment of diabetes.* Diabetes, 57(3): 737–745, 2008.

Katja Hansen, Fabian Rathke, Timon Schroeter, Georg Rast, Thomas Fox, Jan Kriegl, Sebastian Mika. *Bias-correction of regression models: A case study on hERG inhibition.* Journal of Chemical Information and Modeling, 49(6): 1486–1496, 2009.

Roy Hartley, Herbert Morrison III, Felipe Balza, Neil Towers. *Substituted truxillic and truxinic acids in cell walls of Cynodon dactylon.* Phytochemistry, 29(12): 3699–3703, 1990.

Trevor Hastie, Robert Tibshirani, Jerome Friedman. *The Elements of Statistical Learning. Data Mining, Inference, and Prediction.* Springer, New York, 2003.

Merja Heinäniemi, Oskari Uski, Tatjana Degenhardt, Carsten Carlberg. *Meta-analysis of primary target genes of peroxisome proliferator-activated receptors.* Genome Biology, 8(7): r147, 2007.

Leon Hellman, Barnett Zumoff, Gerald Kessler, Elmer Kara, Ira Rubin, Robert Rosenfeld. *Reduction of cholesterol and lipids in man by ethyl p-chlorophenoxyisobutyrate.* Annals of Internal Medicine, 59(4): 477–494, 1963.

Brad Henke. *Peroxisome proliferator-activated receptor gamma (PPARγ) ligands and their therapeutic utility.* In *Progress in Medicinal Chemistry*, volume 42, 1–53. Elsevier, 2004.

Brad Henke, Steven Blanchard, Marcus Brackeen, Kathleen Brown, Jeff Cobb, Jon Collins, Wallace Harrington Jr., Mir Hashim, Emily Hull-Ryde, Istvan Kaldor, Steven Kliewer, Debra Lake, Lisa Leesnitzer, Jürgen Lehmann, James Lenhard, Lisa Orband-Miller, John Miller, Robert Mook Jr., Stewart Noble, William Oliver Jr., Derek Parks, Kelli Plunket, Jerzy Szewczyk, Timothy Willson. *N-(2-benzoylphenyl)-L-tyrosine PPARγ agonists. 1. Discovery of a novel series of potent antihyperglycemic and antihyperlipidemic agents.* Journal of Medicinal Chemistry, 41(25): 5020–5036, 1998.

Thomas Henkel, Roger Brunne, Hartwig Müller, Felix Reichel. *Statistical investigation into the structural complementarity of natural products and synthetic compounds.* Angewandte Chemie International Edition, 38(5): 643–647, 1999.

Mei-Hui Hsu, Colin Palmer, Wu Song, Keith Griffin, Eric Johnson. *A carboxyl-terminal extension of the zinc finger domain contributes to the specificity and polarity of peroxisome proliferator-activated receptor DNA binding.* Journal of Biological Chemistry, 273(43): 27 988–27 997, 1998.

Seung-Soon Im, Jae-Woo Kim, Tae-Hyun Kim, Xian-Li Song, So-Youn Kim, Ha-Il Kim, Yong-Ho Ahn. *Identification and characterization of peroxisome proliferator response element in the mouse GLUT2 promoter.* Experimental and Molecular Medicine, 37(2): 101–110, 2005.

Isabelle Issemann, Stephen Green. *Activation of a member of the steroid hormone receptor superfamily by peroxisome proliferators.* Nature, 347(6294): 645–650, 1990.

Toshimasa Itoh, Louise Fairall, Kush Amin, Yuka Inaba, Attila Szanto, Balint Balint, Laszlo Nagy, Keiko Yamamoto, John Schwabe. *Structural basis for the activation of PPARγ by oxidized fatty acids.* Nature Structural and Molecular Biology, 15(9): 924–931, 2008.

Hong-Fang Ji, Xue-Juan Li, Hong-Yu Zhang. *Natural products and drug discovery.* EMBO Reports, 10(3): 194–200, 2009.

Brian Jones. *Peroxisome proliferator-activated receptor (PPAR) modulators: Diabetes and beyond.* Medicinal Research Reviews, 21(6): 540–552, 2001.

Cristiana Juge-Aubry, Agnès Pernin, Tatiana Favez, Albert Burger, Walter Wahli, Christophe Meier, Béatrice Desvergne. *DNA binding properties of peroxisome proliferator-activated receptor subtypes on various natural peroxisome proliferator response elements. Importance of the 5'-flanking region.* Journal of Biological Chemistry, 272(40): 25 252–25 259, 1997.

Mahmoud Kiaei. *Peroxisome proliferator-activated receptor-γ in amyotrophic lateral sclerosis and Huntington's disease.* PPAR Research, article 418765, 2008.

Steven Kliewer, Barry Forman, Bruce Blumberg, Estelita Ong, Uwe Borgmeyer, David Mangelsdorf, Kazuhiko Umesono, Ronald Evans. *Differential expression and activation of a family of murine peroxisome proliferator-activated receptors.* Proceedings of the National Academy of Sciences of the United States of America, 91(15): 7355–7359, 1994.

Steven Kliewer, Kazuhiko Umesono, Daniel Noonan, Richard Heyman, Ronald Evans. *Convergence of 9-cis retinoic acid and peroxisome proliferator signalling pathways through heterodimer formation of their receptors.* Nature, 358(6389): 771–774, 1992.

Andrei Kolmogoroff. *Interpolation und Extrapolation von stationären zufälligen Folgen.* Izvestiya Rossiiskoi Akademii Nauk. SSSR, Seriya Matematicheskaya, 5(1): 3–14, 1941.

Carolyn Komar. *Peroxisome proliferator-activated receptors (PPARs) and ovarian function implications for regulating steroidogenesis, differentiation, and tissue remodeling.* Reproductive Biology and Endocrinology, 3(41), 2005.

David Krämer, Lubna Al-Khalili, Bruno Guigas, Ying Leng, Pablo Garcia-Roves, Anna Krook. *Role of AMP kinase and PPARδ in the regulation of*

lipid and glucose metabolism in human skeletal muscle. Journal of Biological Chemistry, 282(27): 19 313–19 320, 2007.

K. V. Krishnamurthy. *Methods in Cell Wall Cytochemistry.* CRC Press, Boca Raton, 1999.

Bernd Kuhn, Hans Hilpert, Jörg Benz, Alfred Binggeli, Uwe Grether, Roland Humm, Hans Märki, Markus Meyer, Peter Mohr. *Structure-based design of indole propionic acids as novel PPARα/γ co-agonists.* Bioorganic & Medicinal Chemistry Letters, 16(15): 4016–4020, 2006.

Markus Kummer, Michael Heneka. *PPARs in Alzheimer's disease.* PPAR Research, article 403896, 2008.

Steffen Lauritzen. *Time series analysis in 1880. A discussion of contributions made by T.N. Thiele.* International Statistical Review, 49(3): 319–333, 1981.

Jürgen Lehmann, Linda Moore, Tracey Smith-Oliver, William Wilkison, Timothy Willson, Steven Kliewer. *An antidiabetic thiazolidinedione is a high affinity ligand for peroxisome proliferator-activated receptor γ (PPARγ).* Journal of Biological Chemistry, 270(22): 12 953–12 956, 1995.

Mark Leibowitz, Robert Ardecky, Marcus Boehm, Carol Broderick, Mark Carfagna, Diane Crombie, Jennifer D'Arrigo, Garrett Etgen, Margaret Faul, Timothy Grese, Henry Havel, Nancy Hein, Richard Heyman, Diane Jolley, Kay Klausing, Sha Liu, Dale Mais, Christopher Mapes, Keith Marschke, Pierre-Yves Michellys, Chahrzad Montrose-Rafizadeh, Kathleen Ogilvie, Bernadette Pascual, Deepa Rungta, John Tyhonas, Mary Urcan, Marilyn Wardlow, Nathan Yumibe, Anne Reifel-Miller. *Biological characterization of a heterodimer-selective retinoid X receptor modulator: Potential benefits for the treatment of type 2 diabetes.* Endocrinology, 147(2): 1044–1053, 2006.

Jilin Liu, Hui Li, Sumner Burstein, Robert Zurier, Don Chen. *Activation and binding of peroxisome proliferator-activated receptor γ by synthetic cannabinoid ajulemic acid.* Molecular Pharmacology, 63(5): 983–992, 2003.

Bengt Ljung, Krister Bamberg, Björn Dahllöf, Ann Kjellstedt, Nicholas Oakes, Jörgen Östling, Lennart Svensson, Germán Camejo. *AZ 242, a novel PPARα/γ agonist with beneficial effects on insulin resistance and carbohydrate and lipid metabolism in ob/ob mice and obese Zucker rats.* Journal of Lipid Research, 43(11): 1855–1863, 2002.

Patrick Markt, Daniela Schuster, Johannes Kirchmair, Christian Laggner, Thierry Langer. *Pharmacophore modeling and parallel screening for PPAR ligands.* Journal of Computer-Aided Molecular Design, 21(10–11): 575–590, 2007.

Peter Meinke, Harold Wood, Jason Szewczyk. *Nuclear hormone receptor modulators for the treatment of diabetes and dyslipidemia.* In Anthony Wood (editor), *Annual Reports in Medicinal Chemistry*, volume 41, 99–126. Academic Press, London, 2006.

Liliane Michalik, Johan Auwerx, Joel Berger, Krishna Chatterjee, Christopher Glass, Frank Gonzalez, Paul Grimaldi, Takashi Kadowaki, Mitchell Lazar, Stephen O'Rahilly, Colin Palmer, Jorge Plutzky, Janardan Reddy, Bruce Spiegelman, Bart Staels, Walter Wahli. *International union of pharmacology. LXI. Peroxisome proliferator-activated receptors.* Pharmacological Reviews, 58(4): 726–741, 2006.

Liliane Michalik, Béatrice Desvergne, Christine Dreyer, Mathilde Gavillet, Ricardo Laurini, Walter Wahli. *PPAR expression and function during vertebrate development.* International Journal of Developmental Biology, 46(1): 105–114, 2002.

Pierre-Yves Michellys, Robert Ardecky, Jyun-Hung Chen, Diane Crombie, Garrett Etgen, Margaret Faul, Amy Faulkner, Timothy Grese, Richard Heyman, Donald Karanewsky, Kay Klausing, Mark Leibowitz, Sha Liu, Dale Mais, Christopher Mapes, Keith Marschke, Anne Reifel-Miller, Katheen Ogilvie, Deepa Rungta, Anthony Thompson, John Tyhonas, Marcus Boehm. *Novel (2e,4e,6z)-7-(2-alkoxy-3,5-dialkylbenzene)-3-methylocta-2,4,6-trienoic acid retinoid X receptor modulators are active in models of type 2 diabetes.* Journal of Medicinal Chemistry, 46(13): 2683–2696, 2003.

Yoshinori Miyazaki, Archana Mahankali, Masafumi Matsuda, Srikanth Mahankali, Jean Hardies, Kenneth Cusi, Lawrence Mandarino, Ralph DeFronzo. *Effect of pioglitazone on abdominal fat distribution and insulin sensitivity in type 2 diabetic patients.* Journal of Clinical Endocrinology & Metabolism, 87(6): 2784–2791, 2002.

Herbert Morrison III, Roy Hartley, David Himmelsbach. *Synthesis of substituted truxillic acids from p-coumaric and ferulic acid: Simulation of photodimerization in plant cell walls.* Journal of Agricultural and Food Chemistry, 40(5): 768–771, 1992.

Sunder Mudaliar, Robert Henry. *PPAR agonists in health and disease: A pathophysiologic and clinical overview.* Current Opinion in Endocrinology Diabetes and Obesity, 9(4): 285–302, 2002.

Kendall Nettles. *Insights into PPARγ from structures with endogenous and covalently bound ligands.* Nature Structural and Molecular Biology, 15(9): 893–895, 2008.

David Newman, Gordon Cragg. *Natural products as sources of new drugs over the last 25 years.* Journal of Natural Products, 70(3): 461–477, 2007.

James Nichols, Derek Parks, Thomas Consler, Steven Blanchard. *Development of a scintillation proximity assay for peroxisome proliferator-activated receptor γ ligand binding domain.* Analytical Biochemistry, 257(2): 112–119, 1998.

Robert Nolte, Bruce Wisely, Stefan Westin, Jeffery Cobb, Millard Lambert, Riki Kurokawa, Michael Rosenfeld, Timothy Willson, Christopher Glass, Michael Milburn. *Ligand binding and co-activator assembly of the peroxisome proliferator-activated receptor-γ.* Nature, 395(6698): 137–143, 1998.

Nuclear Receptors Nomenclature Committee. *A unified nomenclature system for the nuclear receptor superfamily.* Cell, 97(2): 161–163, 1999.

Nicholas Oakes, Craig Kennedy, Arthur Jenkins, Ross Laybutt, Don Chisholm, Edward Kraegen. *A new antidiabetic agent, BRL 49653, reduces lipid availability and improves insulin action and glucoregulation in the rat.* Diabetes, 43(10): 1203–1210, 1994.

Olga Obrezanova, Gábor Csányi, Joelle Gola, Matthew Segall. *Gaussian processes: A method for automatic QSAR modeling of ADME properties.* Journal of Chemical Information and Modeling, 47(5): 1847–1857, 2007.

Tove Östberg, Stefan Svensson, Göran Selén, Jonas Uppenberg, Markus Thor, Maj Sundbom, Mona Sydow-Bäckman, Anna-Lena Gustavsson, Lena Jendeberg. *A new class of peroxisome proliferator-activated receptor agonists with a novel binding epitope shows antidiabetic effects.* Journal of Biological Chemistry, 279(39): 41 124–41 130, 2004.

Eckhard Ottow, Hilmar Weinmann (editors). *Nuclear Receptors as Drug Targets*, volume 39 of *Methods and Principles in Medicinal Chemistry.* Wiley, Weinheim, 2008.

Albert Overhauser. *Polarization of nuclei in metals.* Physical Review, 92(2): 411–415, 1953.

John Overington, Bissan Al-Lazikani, Andrew Hopkins. *How many drug targets are there?* Nature Reviews Drug Discovery, 5(12): 993–996, 2006.

David Patsouris, Michael Müller, Sander Kersten. *Peroxisome proliferator activated receptor ligands for the treatment of insulin resistance.* Current Opinion in Investigational Drugs, 5(10): 1045–1050, 2004.

Marjorie Peraza, Andrew Burdick, Holly Marin, Frank Gonzalez, Jeffrey Peters. *The toxicology of ligands for peroxisome proliferator-activated receptors (PPAR).* Toxicological Sciences, 90(2): 269–295, 2006.

Bernard Pirard. *Peroxisome proliferator-activated receptors target family landscape: A chemometrical approach to ligand selectivity based on protein binding site analysis.* Journal of Computer-Aided Molecular Design, 17(12): 785–796, 2003.

Ewgenij Proschak, Kerstin Sander, Heiko Zettl, Yusuf Tanrikulu, Oliver Rau, Petra Schneider, Manfred Schubert-Zsilavecz, Holger Stark, Gisbert Schneider. *From molecular shape to potent bioactive agents II: Fragment-based de novo design.* ChemMedChem, 4(1): 45–48, 2008.

Rodrigo Quintanilla, Youngnam Jin, Karen Fuenzalida, Miguel Bronfman, Gail Johnson. *Rosiglitazone treatment prevents mitochondrial dysfunction in mutant huntingtin-expressing cells. Possible role of peroxisome proliferator-activated receptor-γ (PPARγ) in the pathogenesis of Huntington disease.* Journal of Biological Chemistry, 283(37): 25 628–25 637, 2008.

Carl Rasmussen, Christopher Williams. *Gaussian Processes for Machine Learning.* MIT Press, Cambridge, 2006.

Oliver Rau. *Screening pflanzlicher Extrakte auf Aktivierung des humanen Peroxisomen Proliferator-aktivierten Rezeptors und seiner Subtypen.* Ph.D. thesis, Johann Wolfgang Goethe-University, Frankfurt am Main, Germany, 2007.

Oliver Rau, Mario Wurglics, Alexander Paulke, Jessica Zitzkowski, Nadine Meindl, Andreas Bock, Theodor Dingermann, Mona Abdel-Tawab, Manfred Schubert-Zsilavecz. *Carnosic acid and carnosol, phenolic diterpene compounds of the labiate herbs rosemary and sage, are activators of the human peroxisome proliferator-activated receptor gamma.* Planta Medica, 72(10): 881–887, 2006.

Oliver Rau, Heiko Zettl, Laura Popescu, Dieter Steinhilber, Manfred Schubert-Zsilavecz. *The treatment of dyslipidemia — what's left in the pipeline?* ChemMedChem, 3(2): 206–221, 2008.

Evan Rosen, Bruce Spiegelman. *Peroxisome proliferator-activated receptor γ ligands and atherosclerosis: Ending the heartache.* The Journal of Clinical Investigation, 106(5): 629–632, 2000.

Evan Rosen, Bruce Spiegelman. *PPARγ: A nuclear regulator of metabolism, differentiation, and cell growth.* Journal of Biological Chemistry, 276(41): 37 731–37 734, 2001.

Anne Rubenstrunk, Rémy Hanf, Dean Hum, Jean-Charles Fruchart, Bart Staels. *Safety issues and prospects for future generations of PPAR modulators.* Biochimica et Biophysica Acta - Molecular and Cell Biology of Lipids, 1771(8): 1065–1081, 2007.

Christoph Rücker, Marco Scarsi, Markus Meringer. *2D QSAR of PPARγ agonist binding and transactivation.* Bioorganic & Medicinal Chemistry, 14(15): 5178–5195, 2006.

Joachim Rudolph, Libing Chen, Dyuti Majumdar, William Bullock, Michael Burns, Thomas Claus, Fernando Dela Cruz, Michelle Daly, Frederick Ehrgott, Jeffrey Johnson, James Livingston, Robert Schoenleber, Jeffrey Shapiro, Ling Yang, Manami Tsutsumi, Xin Ma. *Indanylacetic acid derivatives carrying 4-thiazolyl-phenoxy tail groups, a new class of potent PPAR α/γ/δ pan agonists: Synthesis, structure-activity relationship, and in vivo efficacy.* Journal of Medicinal Chemistry, 50(5): 984–1000, 2007.

Matthias Rupp, Petra Schneider, Gisbert Schneider. *Distance phenomena in high-dimensional chemical descriptor spaces: Consequences for similarity-based approaches.* Journal of Computational Chemistry, 30(14): 2285–2296, 2009.

Yojiro Sakiyama. *The use of machine learning and nonlinear statistical tools for ADME prediction.* Expert opinion on drug metabolism & toxicology, 5(2): 149–169, 2009.

Noeris Salam, Tom Huang, Bhavani Kota, Moon Kim, Yuhao Li, David Hibbs. *Novel PPAR-gamma agonists identified from a natural product library: A virtual screening, induced-fit docking and biological assay study.* Chemical Biology & Drug Design, 71(1): 57–70, 2008.

Jennifer Schlezinger, Gregory Howard, Christopher Hurst, Jessica Emberley, David Waxman, Thomas Webster, David Sherr. *Environmental and endogenous peroxisome proliferator-activated receptor gamma agonists induce bone marrow b cell growth arrest and apoptosis: Interactions between mono(2-ethylhexyl)phthalate, 9-cis-retinoic acid, and 15-deoxy-$\delta^{12,14}$-prostaglandin J_2.* The Journal of Immunology, 173(5): 3165–3177, 2004.

Gisbert Schneider, Werner Neidhart, Thomas Giller, Gerard Schmid. *"Scaffold-hopping" by topological pharmacophore search: A contribution to virtual screening.* Angewandte Chemie International Edition, 38(19): 2894–2896, 1999.

Kristina Schoonjans, Julia Peinado-Onsurbe, Anne-Marie Lefebvre, Rich Heyman, Mike Briggs, Samir Deeb, Bart Staels, Johan Auwerx. *PPARα and PPARγ activators direct a tissue-specific transcriptional response via a PPRE in the lipoprotein lipase gene.* The EMBO Journal, 15: 5336–5348, 1996.

Kristina Schoonjans, Mitsuhiro Watanabe, Hiroyuki Suzuki, Abderrahim Mahfoudi, Grigorios Krey, Walter Wahli, Paul Grimaldi, Bart Staels, Tokuo Yamamoto, Johan Auwerx. *Induction of the acyl-coenzyme A synthetase gene by fibrates and fatty acids is mediated by a peroxisome proliferator response element in the c promoter.* Journal of Biological Chemistry, 270(3): 19 269–19 276, 1995.

Francisco Schopfer, Yiming Lin, Paul Baker, Taixing Cui, Minerva Garcia-Barrio, Jifeng Zhang, Kai Chen, Yuqing Chen, Bruce Freeman. *Nitrolinoleic acid: An endogenous peroxisome proliferator-activated receptor γ ligand.* Proceedings of the National Academy of Sciences of the United States of America, 102(7): 2340–2345, 2005.

Timon Schroeter, Anton Schwaighofer, Sebastian Mika, Antonius Ter Laak, Detlev Suelzle, Ursula Ganzer, Nikolaus Heinrich, Klaus-Robert Müller. *Machine learning models for lipophilicity and their domain of applicability.* Molecular Pharmaceutics, 4(4): 524–538, 2007a.

Timon Schroeter, Anton Schwaighofer, Sebastian Mika, Antonius Ter Laak, Detlev Suelzle, Ursula Ganzer, Nikolaus Heinrich, Klaus-Robert Müller. *Predicting lipophilicity of drug-discovery molecules using Gaussian process models.* ChemMedChem, 2(9): 1265–1267, 2007b.

Timon Schroeter, Anton Schwaighofer, Sebastian Mika, Antonius Ter Laak, Detlev Sülzle, Ursula Ganzer, Nikolaus Heinrich, Klaus-Robert Müller. *Estimating the domain of applicability for machine learning QSAR models: A study on aqueous solubility of drug discovery molecules.* Journal of Computer-Aided Molecular Design, 21(9): 485–498, 2007c.

Anton Schwaighofer, Timon Schroeter, Sebastian Mika, Katja Hansen, Antonius ter Laak, Philip Lienau, Andreas Reichel, Nikolaus Heinrich, Klaus-Robert Müller. *A probabilistic approach to classifying metabolic stability.* Journal of Chemical Information and Modeling, 48(4): 785–796, 2008.

Anton Schwaighofer, Timon Schroeter, Sebastian Mika, Julian Laub, Antonius ter Laak, Detlev Sülzle, Ursula Ganzer, Nikolaus Heinrich, Klaus-Robert Müller. *Accurate solubility prediction with error bars for electrolytes: A machine learning approach.* Journal of Chemical Information and Modeling, 47(2): 407–424, 2007.

Mathieu Seimandi, Géraldine Lemaire, Arnaud Pillon, Agnés Perrin, Isabelle Carlavan, Johannes Voegel, Françoise Vignon, Jean-Claude Nicolas, Patrick Balaguer. *Differential responses of PPARα, PPARδ, and PPARγ reporter cell lines to selective PPAR synthetic ligands.* Analytical Biochemistry, 344(1): 8–15, 2005.

Barry Shearer, Hari Patel, Andrew Billin, James Way, Deborah Winegar, Millard Lambert, Robert Xu, Lisa Leesnitzer, Raymond Merrihew, Stephane Huet, Timothy Willson. *Discovery of a novel class of PPARδ partial agonists.* Bioorganic & Medicinal Chemistry Letters, 18(18): 5018–5022, 2008a.

Barry Shearer, David Steger, James Way, Thomas Stanley, David Lobe, Didier Grillot, Marie Iannone, Mitchell Lazar, Timothy Willson, Andrew Billin. *Identification and characterization of a selective peroxisome proliferator-activated receptor β/δ (NR1C2) antagonist.* Molecular Endocrinology, 22(2): 523–529, 2008b.

Talia Sher, Hua-Fang Yi, Wesley McBride, Frank Gonzalez. *cDNA cloning, chromosomal mapping, and functional characterization of the human peroxisome proliferator activated receptor.* Biochemistry, 32(21): 5598–5604, 1993.

Masako Shimura, Akiko Hasumi, Toshiko Minato, Mayu Hosono, Yutaka Miura, Satoru Mizutani, Keiji Kondo, Shinichi Oikawa, Aruto Yoshida. *Isohumulones modulate blood lipid status through the activation of PPARα.* Biochimica et Biophysica Acta - Molecular and Cell Biology of Lipids, 1736(1): 51–60, 2005.

Takuma Shiraki, Narutoshi Kamiya, Sayaka Shiki, Takashi Kodama, Akira Kakizuka, Hisato Jingami. *α,β-unsaturated ketone is a core moiety of natural ligands for covalent binding to peroxisome proliferator-activated receptor γ.* Journal of Biological Chemistry, 280(14): 14 145–14 153, 2005.

Michael Sierra, Véronique Beneton, Anne-Bénédict Boullay, Thierry Boyer, Andrew Brewster, Frédéric Donche, Marie-Claire Forest, Marie-Hélène Fouchet, Françoise Gellibert, Didier Grillot, Millard Lambert, Alain Laroze, Christelle Le Grumelec, Jean Michel Linget, Valerie Montana, Van-Loc Nguyen, Edwige Nicodème, Vipul Patel, Annie Penfornis, Olivier Pineau, Danig Pohin, Florent Potvain, Géraldine Poulain, Cécile Bertho Ruault, Michael Saunders, Jérôme Toum, Eric Xu, Robert Xu, Pascal Pianetti. *Substituted 2-[(4-aminomethyl)phenoxy]-2-methylpropionic acid PPARα agonists. 1. Discovery of a novel series of potent HDLc raising agents.* Journal of Medicinal Chemistry, 50(4): 685–695, 2007.

Jai Pal Singh, Raymond Kauffman, William Bensch, Guoming Wang, Pam McClelland, James Bean, Chahrzad Montrose, Nathan Mantlo, Asavari Wagle. *Identification of a novel selective peroxisome proliferator-activated receptor α agonist, 2-methyl-2-(4-{3-[1-(4-methylbenzyl)-5-oxo-4,5-dihydro-1H-1,2,4-triazol-3-yl]propyl}phenoxy)propanoic acid (LY518674), that produces marked changes in serum lipids and apolipoprotein A-1 expression.* Molecular Pharmacology, 68(3): 763–769, 2005.

Santosh Kumar Singh, Achyut Narayan Kesari, Rajesh Kumar Gupta, Dolly Jaiswal, Geeta Watal. *Assessment of antidiabetic potential of Cynodon dactylon extract in streptozotocin diabetic rats.* Journal of Ethnopharmacology, 114(2): 174–179, 2007.

Charles Slichter. *Principles of Magnetic Resonance.* Springer, New York, third edition, 1990.

Sören Sonnenburg, Gunnar Rätsch, Christin Schäfer, Bernhard Schölkopf. *Large scale multiple kernel learning.* Journal of Machine Learning Research, 7(7): 1531–1565, 2006.

Michael Sporn, Nanjoo Suh, David Mangelsdorf. *Prospects for prevention and treatment of cancer with selective PPARγ modulators (SPARMs).* Trends in Molecular Medicine, 7(9): 395–400, 2001.

Attila Szanto, Laszlo Nagy. *The many faces of PPARγ: Anti-inflammatory by any means?* Immunobiology, 213(9–10): 789–803, 2008.

Makoto Takamura, Mitsuya Sakurai, Eriko Yamada, Sachie Fujita, Makoto Yachi, Toshiyuki Takagi, Aya Isobe, Yuka Hagisawa, Toshihiko Fujiwara, Hiroaki Yanagisawa. *Synthesis and biological activity of novel α-substituted β-phenylpropionic acids having pyridin-2-ylphenyl moiety as antihyperglycemic agents.* Bioorganic & Medicinal Chemistry, 12(9): 2419–2439, 2004.

Nicholas Taleb. *Fooled by Randomness: The Hidden Role of Chance in Life and in the Markets.* Random House, second edition, 2005.

Feifei Tian, Li Yang, Fenglin Lv, Peng Zhou. *Predicting liquid chromatographic retention times of peptides from the Drosophila melanogaster proteome by machine learning approaches.* Analytica Chimica Acta, 644(1–2): 10–16, 2009.

Eric Tien, Joshua Gray, Jeffrey Peters, John Vanden Heuvel. *Comprehensive gene expression analysis of peroxisome proliferator-treated immortalized hepatocytes: Identification of peroxisome proliferator-activated receptor α-dependent growth regulatory genes.* Cancer Research, 63(18): 5767–5780, 2003.

Peter Tontonoz, Erding Hu, Jerry Devine, Elmus Beale, Bruce Spiegelman. *PPARγ2 regulates adipose expression of the phosphoenolpyruvate carboxykinase gene.* Molecular and Cellular Biology, 15(1): 351–357, 1995.

Peter Tontonoz, Erding Hu, Bruce Spiegelman. *Stimulation of adipogenesis in fibroblasts by PPARγ2, a lipid-activated transcription factor.* Cell, 79(7): 1147–1156, 1994.

Cicerone Tudor, Jérôme Feige, Harikishore Pingali, Vidya Bhushan Lohray, Walter Wahli, Béatrice Desvergne, Yves Engelborghs, Laurent Gelman. *Association with coregulators is the major determinant governing peroxisome proliferator-activated receptor mobility in living cells.* Journal of Biological Chemistry, 282(7): 4417–4426, 2007.

Sandra Ulrich, Stefan Loitsch, Oliver Rau, Andreas von Knethen, Bernhard Brüne, Manfred Schubert-Zsilavecz, Jürgen Stein. *Peroxisome proliferator-activated receptor γ as a molecular target of resveratrol-induced modulation of polyamine metabolism.* Cancer Research, 66(14): 7348–7354, 2006.

Vellarkad Viswanadhan, Arup Ghose, Ganapathi Revankar, Roland Robins. *Atomic physicochemical parameters for three dimensional structure directed quantitative structure-activity relationships. 4. Additional parameters for hydrophobic and dispersive interactions and their applications for an automated superposition of certain naturally occurring nucleoside antibiotics.* Journal of Chemical Information and Computer Sciences, 29(3): 163–172, 1989.

David Weininger. *SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules.* Journal of Chemical Information and Computer Sciences, 28(1): 31–36, 1988.

David Weininger, Arthur Weininger, Joseph Weininger. *SMILES. 2. Algorithm for generation of unique SMILES notation.* Journal of Chemical Information and Modeling, 29(2): 97–101, 1989.

Martin Weisel, Ewgenij Proschak, Gisbert Schneider. *Pocketpicker: Analysis of ligand binding-sites with shape descriptors.* Chemistry Central Journal, 1(7), 2007.

John Welch, Mercedes Ricote, Taro Akiyama, Frank Gonzalez, Christopher Glass. *PPARγ and PPARδ negatively regulate specific subsets of lipopolysaccharide and IFN-target genes in macrophages.* Proceedings of the National Academy of Sciences of the United States of America, 100(11): 6712–6717, 2003.

Yi-Min Wen, Bao-Liang Lu, Hai Zhao. *Equal clustering makes min-max modular support vector machine more efficient.* In *Proceedings of the 12th International Conference on Neural Information Processing (ICONIP 2005), Taipei, Taiwan, October 30–November 2*, 77–82. IEEE Computer Society, 2005.

Norbert Wiener. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series.* MIT Press, Cambridge, 1949.

Ashley Wilkinson, Gregory Monteith, Nicholas Shaw, Chun-Nam Lin, Michael Gidley, Sarah Roberts-Thomson. *Effects of the mango components mangiferin and quercetin and the putative mangiferin metabolite norathyriol on the transactivation of peroxisome proliferator-activated receptor isoforms.* Journal of Agricultural and Food Chemistry, 56(9): 3037–3042, 2008.

Timothy Willson, Peter Brown, Daniel Sternbach, Brad Henke. *The PPARs: From orphan receptors to drug discovery.* Journal of Medicinal Chemistry, 43(4): 527–550, 2000.

Timothy Willson, Millard Lambert, Steven Kliewer. *Peroxisome proliferator-activated receptor γ and metabolic disease.* Annual Review of Biochemistry, 70: 341–367, 2001.

Zhidan Wu, Yuhong Xie, Ron Morrison, Nancy Bucher, Stephen Farmer. *PPARγ induces the insulin-dependent glucose transporter GLUT4 in the absence of C/EBPα during the conversion of 3T3 fibroblasts into adipocytes.* The Journal of Clinical Investigation, 101(1): 22–32, 1998.

Eric Xu, Millard Lambert, Valerie Montana, Kelli Plunket, Linda Moore, Jon Collins, Jeffery Oplinger, Steven Kliewer, Robert Gampe Jr., David McKee, John Moore, Timothy Willson. *Structural determinants of ligand binding selectivity between the peroxisome proliferator-activated receptors.* Proceedings of the National Academy of Sciences of the United States of America, 98(24): 13 919–13 924, 2001.

Hiroaki Yajima, Emiko Ikeshima, Maho Shiraki, Tomoka Kanaya, Daisuke Fujiwara, Hideharu Odai, Nobuyo Tsuboyama-Kasaoka, Osamu Ezaki, Shinichi Oikawa, Keiji Kondo. *Isohumulones, bitter acids derived from hops, activate both peroxisome proliferator-activated receptor α and γ and reduce insulin resistance.* Journal of Biological Chemistry, 279(32): 33 456–33 462, 2004.

Takeo Yoshikawa, Zoran Brkanac, Barbara Dupont, Guo-Qiang Xing, Robin Leach, Sevilla Detera-Wadleigh. *Assignment of the human nuclear hormone receptor, NUC1 (PPARD), to chromosome 6p21.1–p21.2.* Genomics, 35(3): 637–638, 1996.

Jiangsheng Yu, Xue-Wen Chen. *Bayesian neural network approaches to ovarian cancer identification from high-resolution mass spectrometry data.* Bioinformatics, 21(Supplement 1): i487–i494, 2005.

Fernando Zapata-Gonzalez, Felix Rueda, Jordi Petriz, Pere Domingo, Francesc Villarroya, Julieta Diaz-Delfin, Maria de Madariaga, Joan Domingo. *Human dendritic cell activities are modulated by the omega-3 fatty acid, docosahexaenoic acid, mainly through PPARγ:RXR heterodimers: Comparison with other polyunsaturated fatty acids.* Journal of Leukocyte Biology, 84(4): 1172–1182, 2008.

Fang Zhang, Brian Lavan, Francine Gregoire. *Peroxisome proliferator-activated receptors as attractive antiobesity targets.* Drug News & Perspectives, 17(10): 661–669, 2004.

Peng Zhou, Feifei Tian, Xiang Chen, Zhicai Shang. *Modeling and prediction of binding affinities between the human amphiphysin SH3 domain and its peptide ligands using genetic algorithm-Gaussian processes.* Peptide Science, 90(6): 792–802, 2008.

Peng Zhou, Feifei Tian, Fenglin Lv, Zhicai Shang. *Comprehensive comparison of eight statistical modelling methods used in quantitative structure-retention relationship studies for liquid chromatographic retention times of peptides generated by protease digestion of the Escherichia coli proteome.* Journal of Chromatography A, 1216(15): 3107–3116, 2009.

Vincent Zoete, Aurelien Grosdidier, Olivier Michielin. *Peroxisome proliferator-activated receptor structures: Ligand specificity, molecular switch and interactions with regulators.* Biochimica et Biophysica Acta - Molecular and Cell Biology of Lipids, 1771(8): 915–925, 2007.

Anna Zomer, Bart van der Burg, Gerbert Jansen, Ronald Wanders, Bwee Tien Poll-The, Paul van der Saag. *Pristanic acid and phytanic acid: Naturally occurring ligands for the nuclear receptor peroxisome proliferator-activated receptor α.* Journal of Lipid Research, 41(11): 1801–1807, 2000.

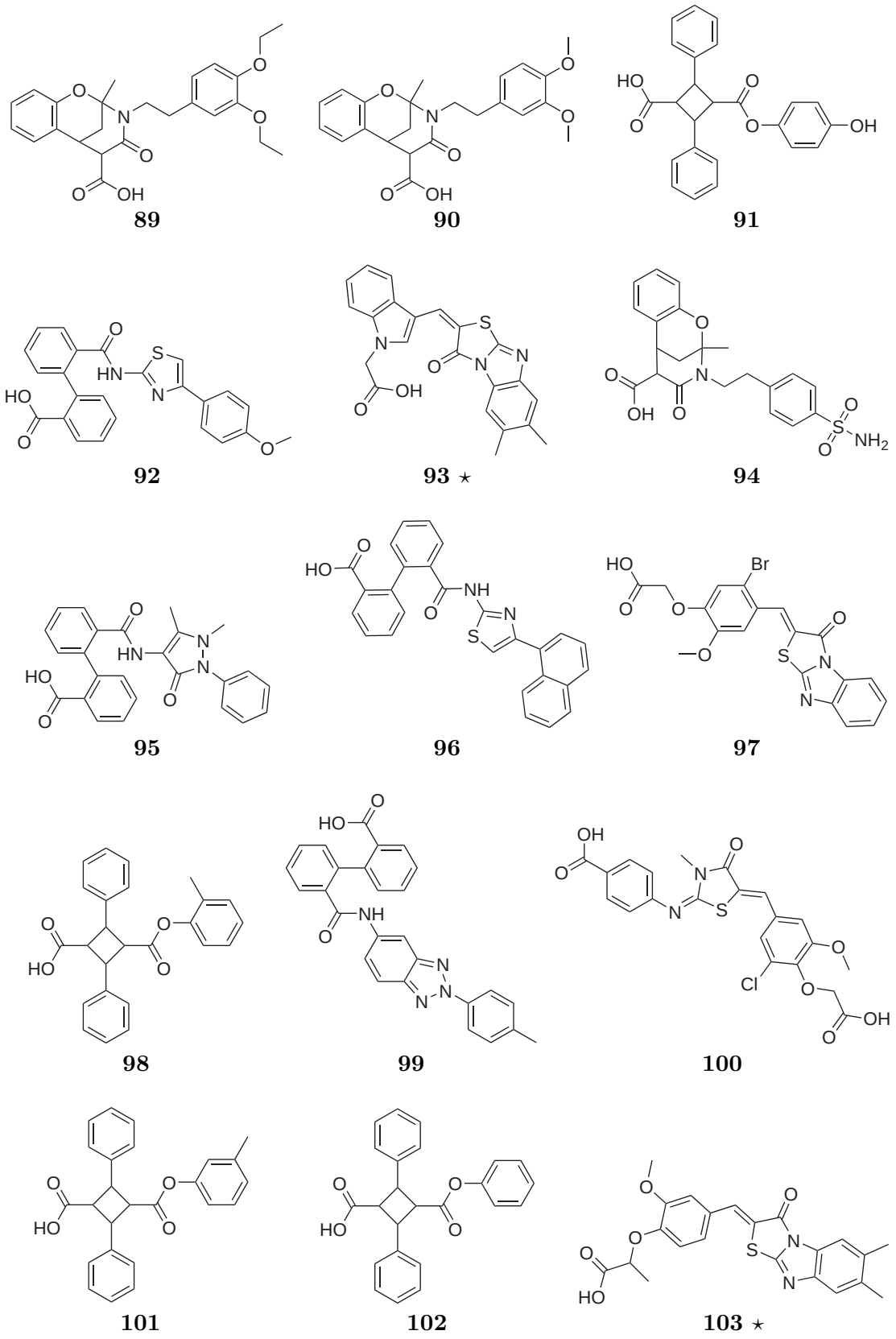
Appendix A

Supplementary data

The following data are provided supplementary to the main text.

- *Top predictions:* Schemes A.1, A.2, and A.3 show the 30 top-ranked compounds of models 7, 14, and 15, respectively, defined in Chapter 4 on p. 162.
- *Nuclear magnetic resonance spectra:* ^{1}H-nuclear magnetic resonance spectrum (^{1}H-NMR, Figure A.1), nuclear Overhauser effect differential spectra (NOE, Figures A.2 and A.3), and rotating frame nuclear Overhauser effect spectrum (ROESY, Figure A.4), used in identification of Compound MR16's configuration (p. 169).

  Experiments were performed on a Bruker NMR spectrometer (Bruker optics, `www.brukeroptics.com`; AV300 MHz for ^{1}H-NMR, NOE; AV400 MHz for ROESY). All spectra were recorded in dimethylsulfoxide-d_6 using standard pulse sequences.



89 **90** **91**

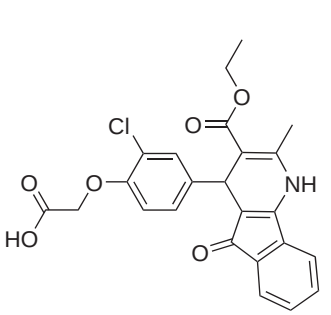
92 **93** ⋆ **94**

95 **96** **97**

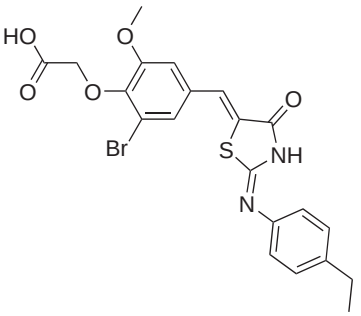
98 **99** **100**

101 **102** **103** ⋆

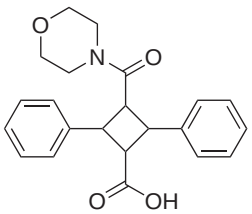
Scheme A.1 30 top-ranked compounds of model 7 (CATS2D/RBF+RQ). ⋆ = selected for assay tests. *Continued on next page...*



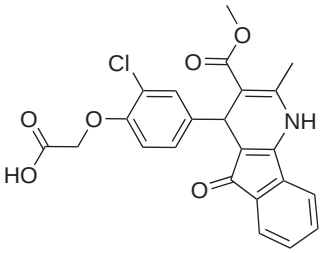
104



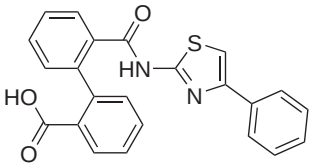
105



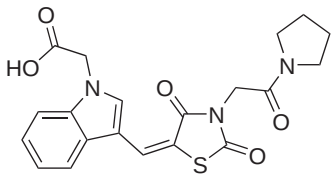
106



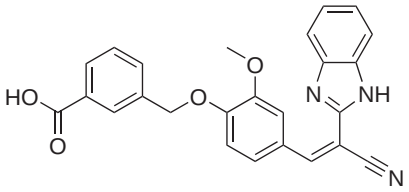
107



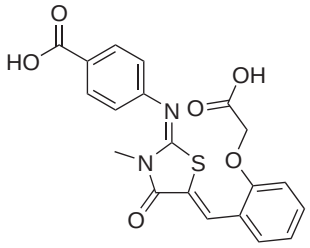
108 ⋆



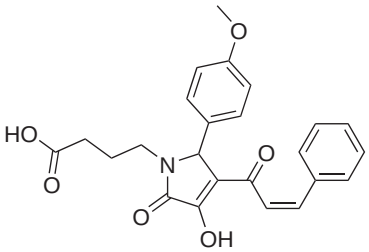
109 ⋆



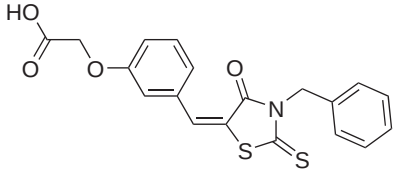
110



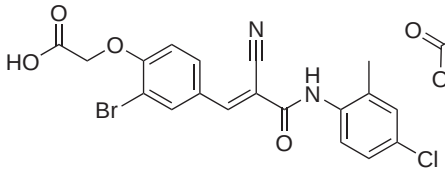
111



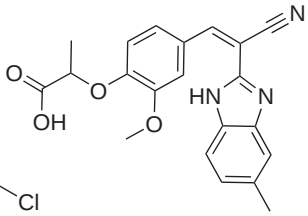
112



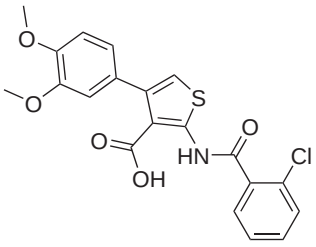
113 ⋆



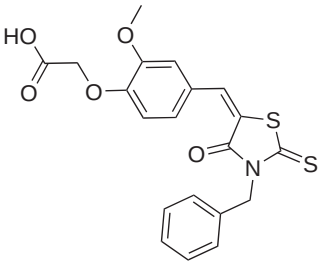
114



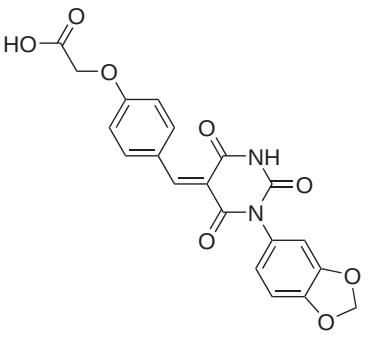
115



116

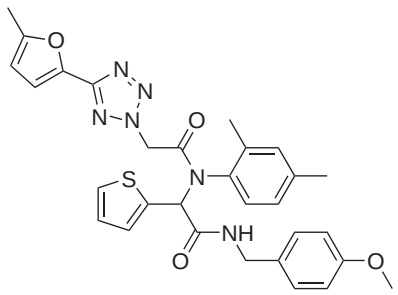


117

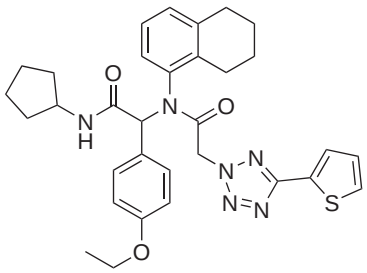


118

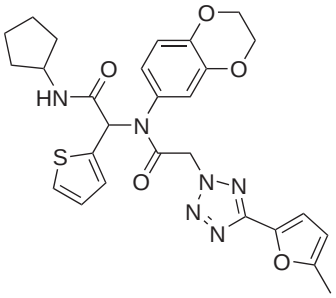
Scheme A.1 ...*continued from previous page.* ⋆ = selected for assay tests.



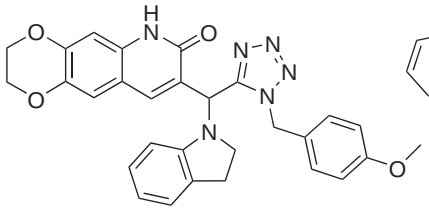
119



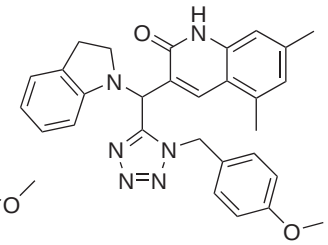
120



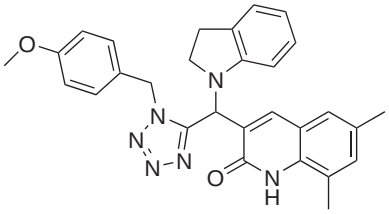
121



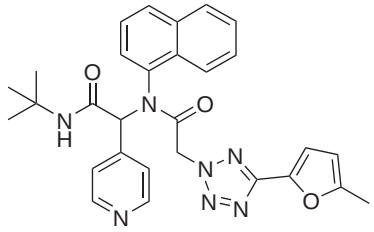
122



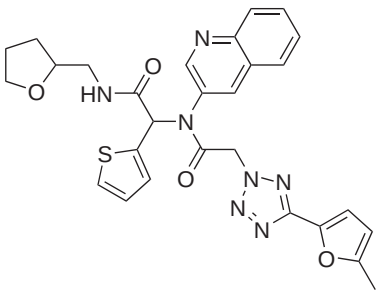
123



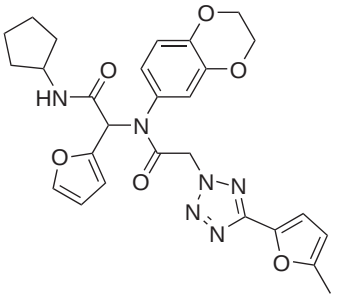
124



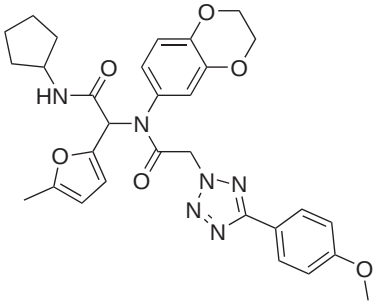
125



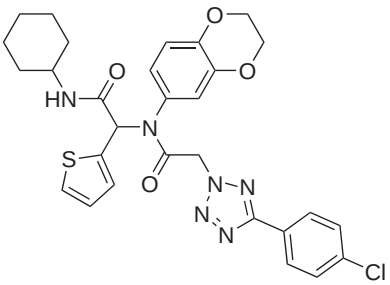
126



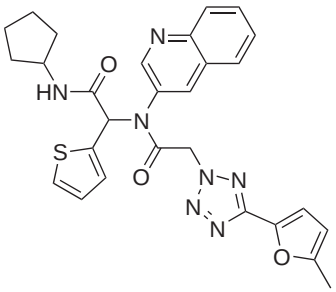
127



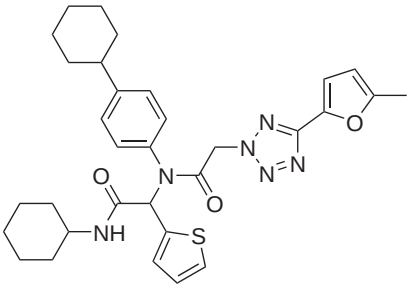
128



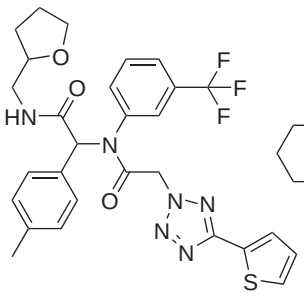
129



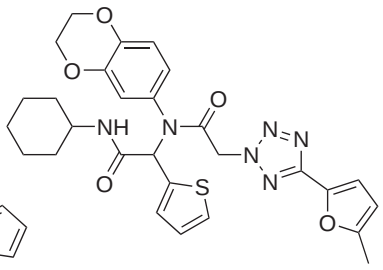
130



131

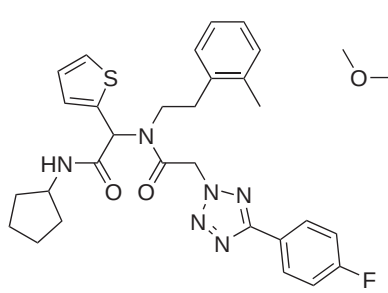


132

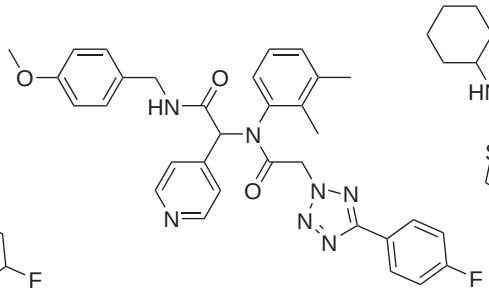


133

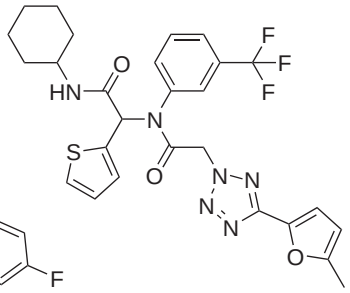
Scheme A.2 30 top-ranked compounds of model 14 (topo/ISOAK+all/RBF+all/RQ+W). ⋆ = selected for assay tests. *Continued on next page...*



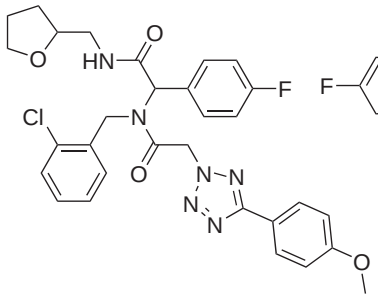
134



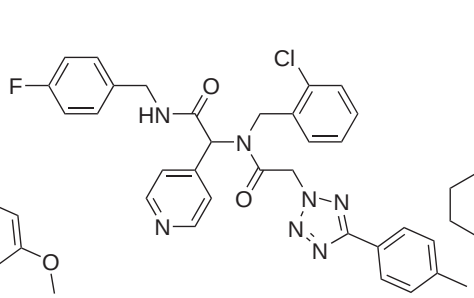
135



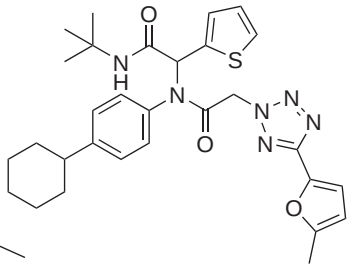
136



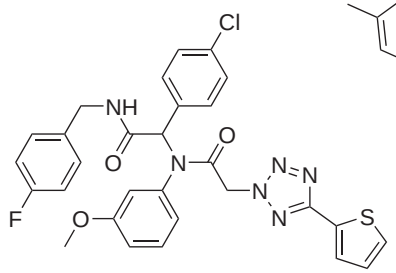
137



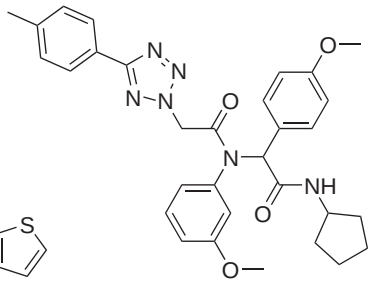
138



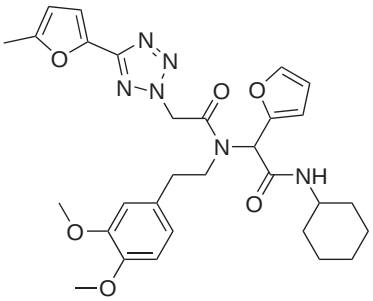
139



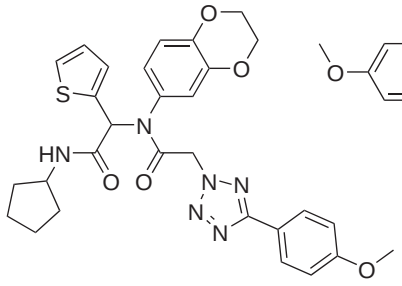
140



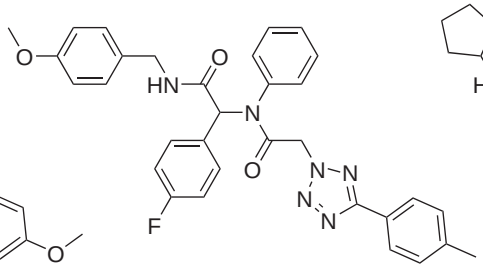
141



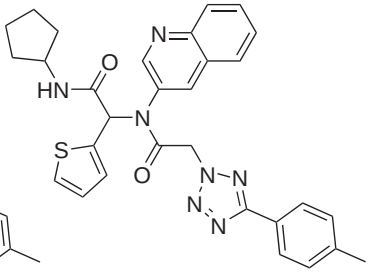
142



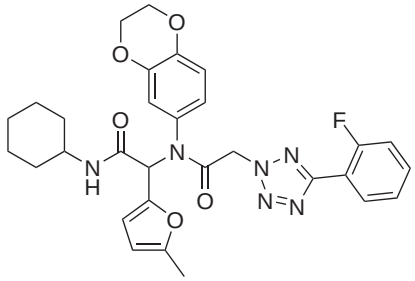
143



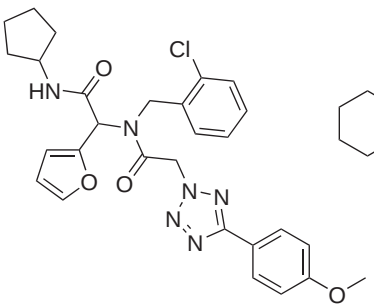
144



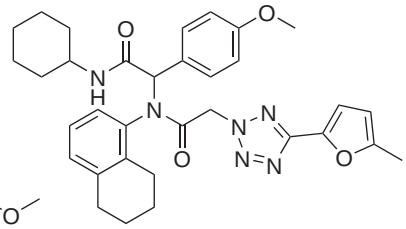
145



146

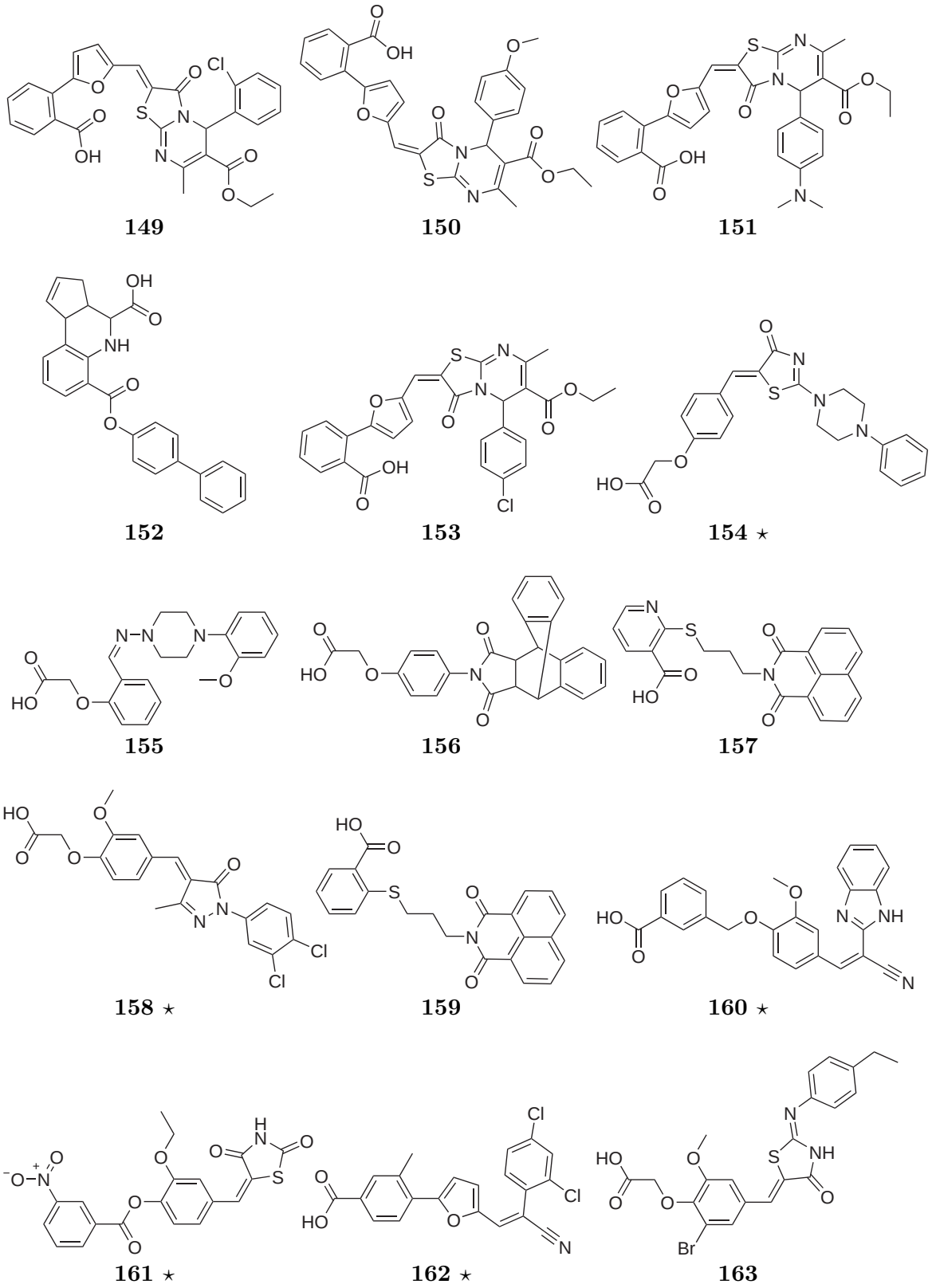


147



148

Scheme A.2 ...*continued from previous page.* ⋆ = selected for assay tests.



149 **150** **151**

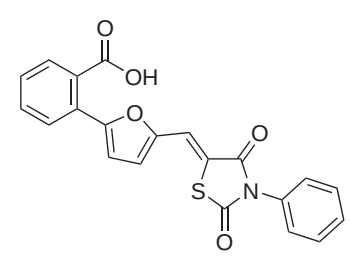
152 **153** **154** ⋆

155 **156** **157**

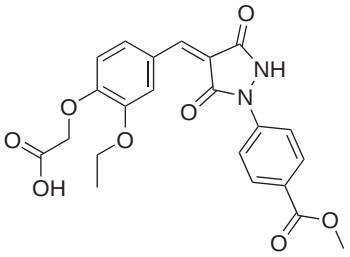
158 ⋆ **159** **160** ⋆

161 ⋆ **162** ⋆ **163**

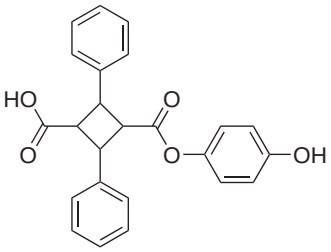
Scheme A.3 30 top-ranked compounds of model 15 (topo/ISOAK + allmkl/RBF + allmkl/RQ). ⋆ = selected for assay tests. *Continued on next page...*



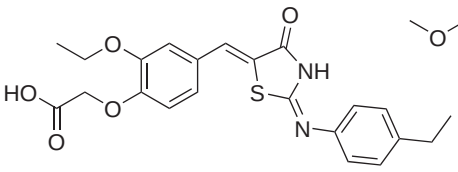
164 ⋆



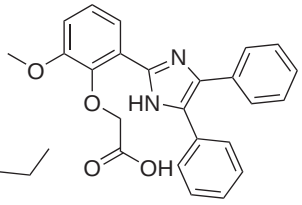
165



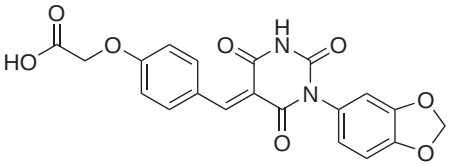
166



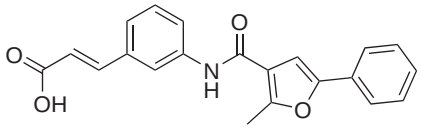
167



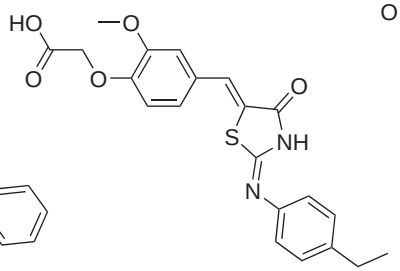
168



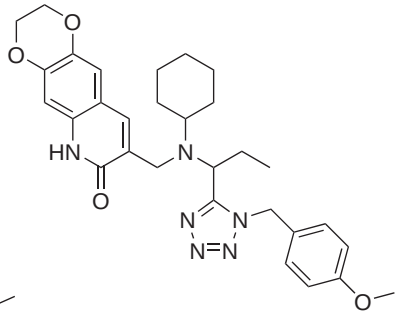
169



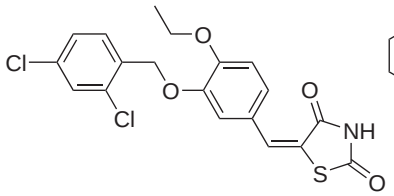
170 ⋆



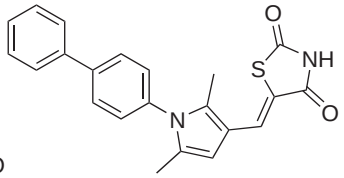
171



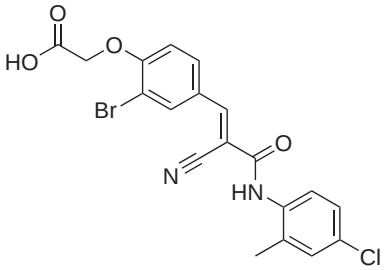
172



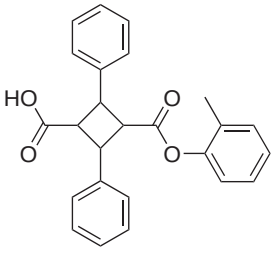
173



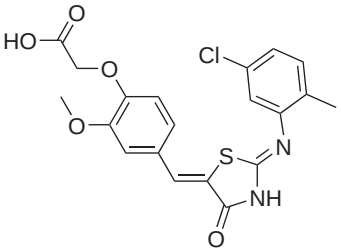
174 ⋆



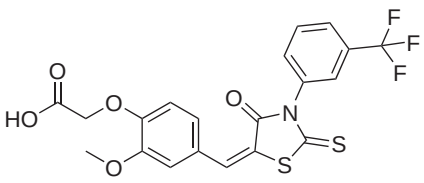
175



176 ⋆



177



178 ⋆

Scheme A.3 ...*continued from previous page.* ⋆ = selected for assay tests.

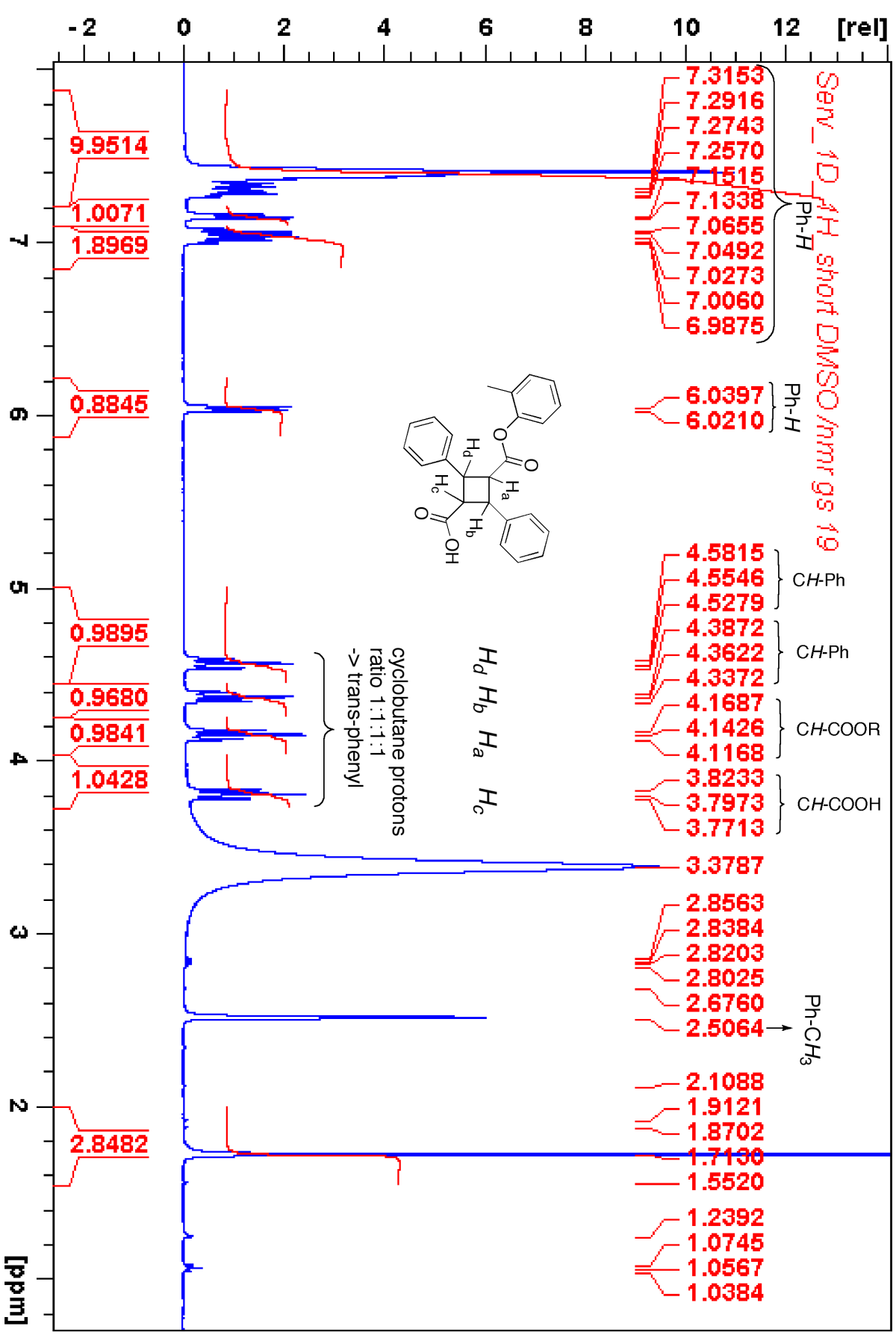


Figure A.1 ^{1}H-NMR spectrum of Compound MR16 in DMSO-d_6 at 300.13 MHz. Proton assignments were verified by HSQC and HMBC experiments. NMR = nuclear magnetic resonance, DMSO = dimethylsulfoxide, HSQC = heteronuclear single quantum coherence, HMBC = heteronuclear multiple bond correlation.

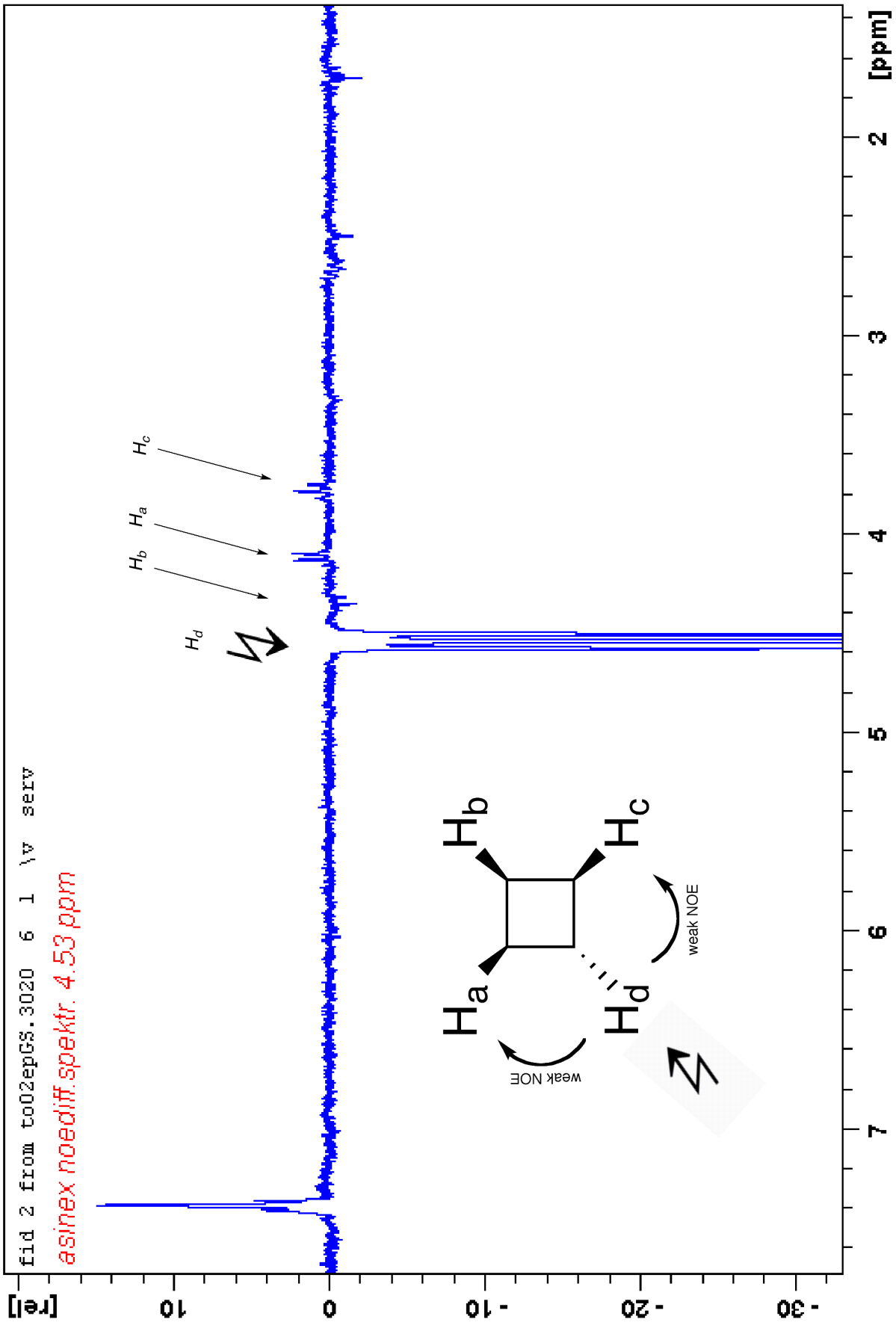


Figure A.2 Nuclear Overhauser effect differential spectrum in DMSO-d_6 at 4.53 ppm and 300.13 MHz, with an irradiation time of 7 s. The cyclobutane proton which stands alone on one side of the ring (H_d) gives weak signals to its trans-neighboring cyclobutane protons (H_a, H_c), and no signal to the other trans-neighboring proton (H_b). DMSO = dimethylsulfoxide.

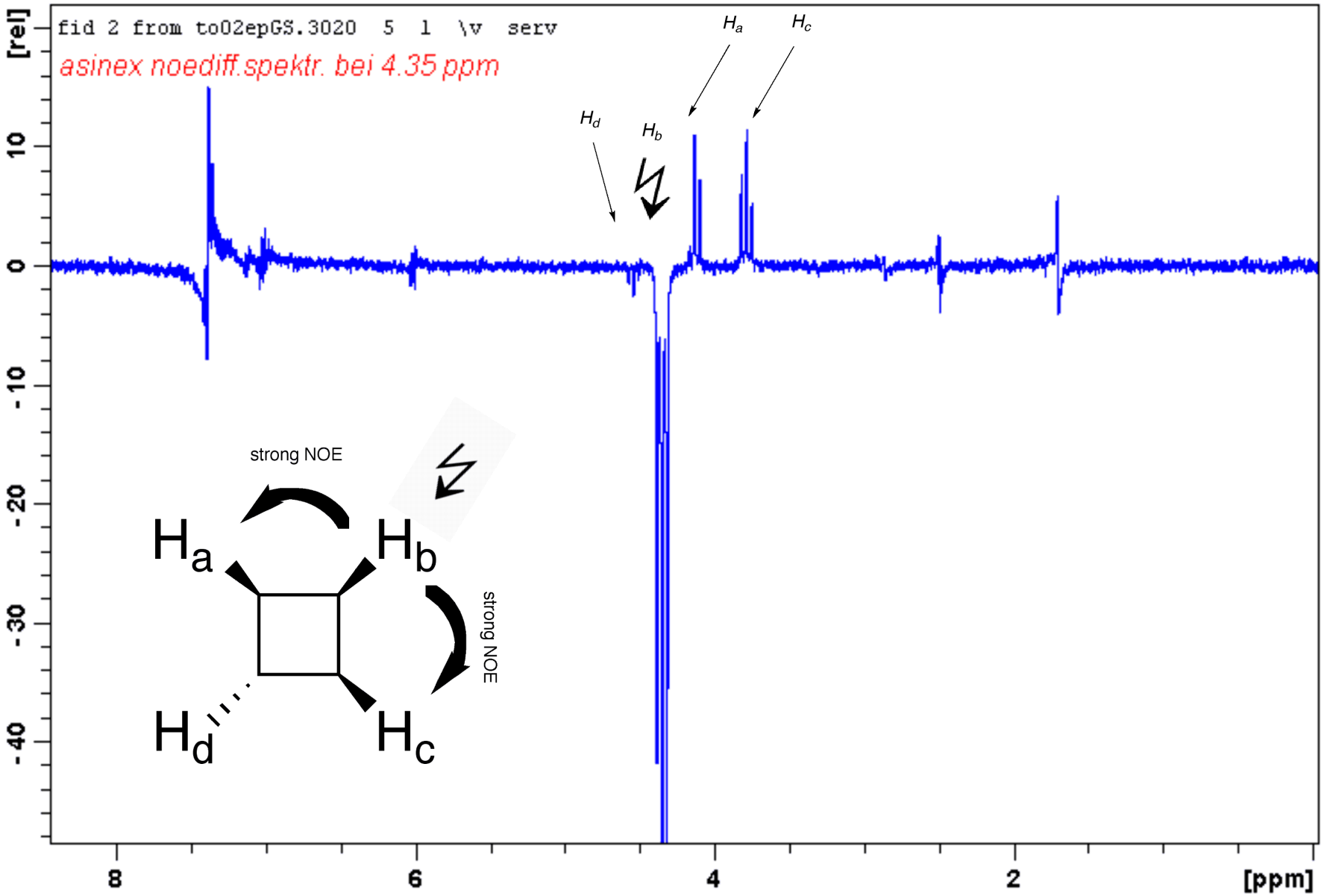


Figure A.3 Nuclear Overhauser effect differential spectrum in DMSO-d_6 at 4.35 ppm and 300.13 MHz, with an irradiation time of 7 s. In contrast to Figure A.2, the cyclobutane proton (H_b) with two cis-neighboring protons (H_a, H_c) gives strong signals to these, and no signal to the trans-standing proton (H_d). DMSO = dimethylsulfoxide.

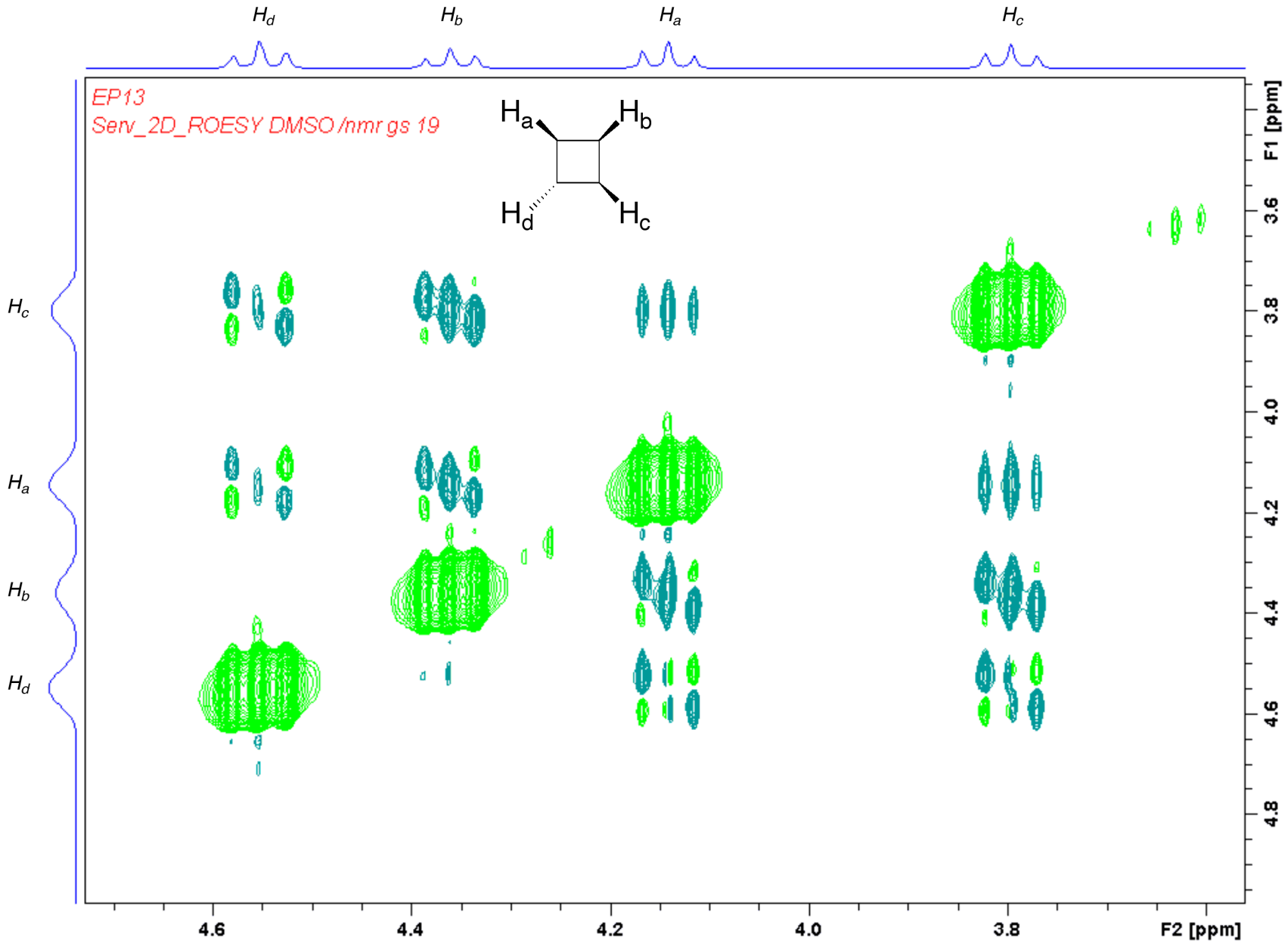


Figure A.4 Rotating frame nuclear Overhauser effect spectrum in DMSO-d_6 at 400.13 MHz with a spin-lock time of 300 ms. The spectrum shows dipolar couplings between all cyclobutane protons except the trans-standing protons H_b and H_d, corroborating the results of the nuclear Overhauser effect differential spectra (Figures A.2, A.3). DMSO = dimethylsulfoxide.