Virtual Screening of Potential Bioactive Substances using the Support Vector Machine approach

Dissertation
zur Erlangung des Doktorgrades
der Naturwissenschaften

vorgelegt beim Fachbereich Chemische und Pharmazeutische Wissenschaften
der Johann Wolfgang Goethe-Universität
in Frankfurt am Main

von
Evgeny Byvatov
aus Moskau

Frankfurt am Main, 2005

Virtual Screening of Potential Bioactive Substances using the Support Vector Machine approach

Dissertation
zur Erlangung des Doktorgrades
der Naturwissenschaften

vorgelegt beim Fachbereich Chemische und Pharmazeutische Wissenschaften
der Johann Wolfgang Goethe-Universität
in Frankfurt am Main

von
Evgeny Byvatov
aus Moskau
Frankfurt am Main, 2005

Vom Fachbereich Chemische und Pharmazeutische Wissenschaften der Johann Wolfgang Goethe-Universität als Dissertation angenommen

Dekan:
Gutachter:
Datum der Disputation

Eidesstattliche Versicherung

Ich erkläre hiermit an Eides statt, dass ich die vorgelegte Dissertation über

Virtual Screening of Potential Bioactive Substances

selbstständig angefertigt und mich anderer Hilfsmittel als der in der in ihr angegebenen nicht bedient habe, insbesondere, dass aus Schriften Entlehnungen, soweit sie in der Dissertation nicht ausdrücklich als solche mit Angabe der betreffenden Schrift bezeichnet sind, nicht stattgefunden haben.

Frankfurt am Main, den ………………

Evgeny Byvatov

Der vorliegende Arbeit wurde in der Zeit von November 2002 bin Januar 2005 an der Johann Wolfgang Goethe-Universität Frankfurt am Main unter der Anleitung von Prof. Dr. Gisbert Schneider durchgeführt.

Table of contents

1 Acknowledgement 1
2 List of Abbreviations 2
3 Declaration 3
4 Introduction 6
4.1 The Drug Discovery process 6
4.2 From molecules to Descriptors 10
4.2.1 2D Molecular Descriptors 10
4.2.1.1 Atom Counts and Bond Counts 10
4.2.1.2 Physical Properties 12
4.2.1.3 Adjacency and Distance Matrix Descriptors 15
4.2.1.4 Connectivity (Kier&Hall) and Kappa Shape Indices 16
4.2.1.5 Descriptors of the Surface Areas 17
4.2.1.6 Pharmacophore Feature Descriptors 18
4.2.1.7 Partial Charge Descriptors 20
4.2.2 Topological Cross-correlation Pharmacophores 22
4.2.3 3D Molecular Descriptors 22
4.2.3.1 Potential Energy Descriptors 23
4.2.3.2 Conformation Dependent Charge Descriptors 24
4.2.3.3 Surface Area, Volume and Shape Descriptors 25
4.2.4 Three Point Pharmacophore 26
4.3 From Descriptors to the Predictive Model. Support Vector Machine... 27
4.3.1 Optimal hyperplane 30
4.3.1.1 The optimal hyperplane algorithm 31
4.3.2 The Soft Margin hyperplane 33
4.3.3 The Method of Convolution of the Dot-Product in Feature Space 36
4.3.4 General Features of Support Vector Machine 38
4.3.4.1 Decision Rules by SVM 38
4.3.4.2 The Support Vector Machine is a universal machine 39
4.3.4.3 Generalization Ability 39
4.3.5 Constructing Separating Hyperplanes 40
4.3.5.1 Optimal hyperplane algorithm 40
4.3.5.2 Soft margin hyperplane Algorithm 43
4.4 Selecting relevant descriptors 47
4.4.1 Feature selection methods 47
4.4.2 Relevance to the Concept: Weak and Strong Relevance 48
4.4.3 General Characteristics of Feature Selection methods 48
4.4.4 Categorization Scheme of Feature Selection Methods 49
4.4.5 Filter Selection Method 50
4.4.5.1 Kolmogorov-Smirnov Statistics. 50
4.4.6 Wrapper Selection Model 51
4.4.6.1 SVM-Based wrapper for feature Selection 52
4.4.7 Embedded Selection Model 54
5 Zusammenfassung 55
6 Lebenslauf 59
7 Publications 60
7.1 SVM applications in bioinformatics 61
7.2 Comparison of support vector machine and artificial neural network systems for drug/nondrug classification 85
7.3 SVM based Feature Selection for Characterization of Focused Compound Collections 94

7.4 SMILIB: Rapid assembly of combinatorial libraries in SMILES 102
7.5 From Virtual to Real Screening for Novel D_3 Dopamine Receptor Ligands 106
7.6 Virtual Screening Filter to Identify Cytochrome P450 2C9 (CYP2C9) Inhibitors based on SVM for Model Building and Feature Visualization 126
7.7 Extraction and visualization of pharmacophore models by SVM....... 162
7.8 Improvement of the efficiency of lead based drug design by active learning 175
8 References 191

1 Acknowkledgment

I thank Prof. Dr. Gisbert Schneider for his inspiring ideas and constant support in the exiting field of Chemoinformatics and for guiding me in my research process.

I thank Lutz, Steffen, Uli and Micha for their help in day-to-day work.

I thank Alireza and Norbert for the technical help regarding IT support of the projects.

I thank the Aventis Molecular Modelling team, especially Dr. Karl-Heinz Baringhaus, for supporting the collaboration with Aventis Pharma, Dr. Hans Matter for supervising the collaborative project, Dr. Tomas Klabunde for providing test results for a GPCR library, and Dr. A. Dudda and Dr. G.U. Kürzel for experimental data for this project.

I thank Prof. Dr. Stark, Prof. Dr. Steinhilber, Britta Sasse, Uli Fechner, Dr. Sadowski, Dr. Karl-Heinz Baringhaus, Dr. Hans Matter and Lutz Franke for co-athoring my scientific articles.

I thank the Beilstein Institut zur Förderung der Chemischen Wissenschaften (Frankfurt am Main) for the financial support of my research.

I thank my parents for the constant support during all my life.

2 List of Abbreviations

2D	Two dimensional
3D	Three dimentional
ADME	Absorbtion, distribution, metabolism and exertion
ANN	Artificial neural network
ASA	Accessible surface area
CATS	Chemically Advanced Template Search
COX	Cyclooxygenase
CYP	Cytochrome P450
CYP2C9	Cytochrome P450 subtype 2C9
EDC	Early development candidate
HTS	High throughput screening
KS	Kolmogorov Smirnov
IC_{50}	Concentration that leads to 50% inhibition of activity
MOE	Molecular Operating Environment
PDE	Probability density estimation
QSAR	Quantitative Structure Activity Relationship
SMILES	Simplified molecular input line entry system
SPR	Statistical pattern recognition
SVM	Support Vector Machine
VC	Vapnik Cherenkov
VS	Virtual screening

3 Declaration

This dissertation is submitted in a cumulative fashion, meaning that results are presented as a collecion of research articles. These include five papers published in peer-reviewed journals, one paper submitted for the publication and two papers in preparation.

Chapter 2 describes the framework of my research:

- Introduction to the drug discovery process
- Role of Machine Learning in the virtual screening for potensial drug candidates
- Molecule representation for virtual screening
- Description of the Support Vector Machine in relation to other Machine Learning techniques
- Consept of the Feature Selection methodology

Chapter 3 contains full articles that constitute the research thesis. The articles are presented in the following thematical order:

- Review of the applications of the Support Vector Machine in Chemo- and Bioinformatics (ref. 1)
- Comparison of the SVM with a Neural Network for drug-likeness prediction (ref. 2)
- Feature selection by SVM (ref. 3)
- Construction of virtual combinatorial libraries (ref. 4)
- Applications of virtual screening for the discovery of new biologically active compounds (ref 5,6,7)
- Developing of a method for improvement of virtual screeining. (ref. 8)

The full references to the article and statements about contribution of the author to each work are presented here:

1. **SVM applications in bioinformatics**
   Byvatov E., Schneider G.
   Appl. Bioinformatics. 2003;2(2):67-77.

   Evgeny Byvatov contributed to this review by providing a comprehensive discussion of the available applications of the Support Vector Machine in Chemo- and Bioinformatics.

2. **Comparison of support vector machine and artificial neural network systems for drug/nondrug classification**
   Byvatov E., Fechner U., Sadowski J., Schneider G.
   J. Chem. Inf. Comput. Sci. 2003; 43(6):1882-9

   Evgeny Byvatov contributed to this paper by providing an extensive comparison of the different neural network training techniques with Support Vector Machine for the task of drug versus non-drug classification.

3. **SVM based Feature Selection for Characterization of Focused Compound Collections**
   Byvatov E., Schneider G.
   J. Chem. Inf. Comput. Sci. 2004; 44(3):993-9

   Evgeny Byvatov contributed to this paper by developing an algorithm for the selection of the features that are relevant for classification of chemical compounds as potential ligands for certain classes the pharmaceutical targets.

4. **SMILIB: Rapid assembly of combinatorial libraries in SMILES notation**
   Schüller A., Schneider G., **Byvatov E.**
   QSAR Comb. Sci. 2003; 22:719-721

   Evgeny Byvatov contributed to this work by coming up with the idea of a fast construction of a combinatorial library using SMILES notation. He has also supervised a student who performed the implementation of the algorithm.

5. **From Virtual to Real Screening for Novel D_3 Dopamine Receptor Ligands**
   Byvatov E., Sasse B.C., Stark H., Schneider G.
   ChemBioChem. *in press*

   Evgeny Byvatov contributed to this work by performing virtual screening of commercially available databases for selective ligands to the dopamine D_3 receptor.

6. **Virtual Screening Filter to Identify Cytochrome P450 2C9 (CYP2C9) Inhibitors based on SVM for Model Building and Feature Visualization**
   Byvatov E., Matter H., Baringhaus K.H., Schneider G.
   J. Med. Chem*., submitted*

   The contribution of Evgeny Byvatov to this work was the construction of the SVM regression and classification model for early identification of molecules with high affinity to CYP 2C9 cytochrome.

7. **Extraction and visualization of pharmacophore models by SVM**
   Byvatov E., Franke L., Werz O., Steinhilber D., Schneider G.
   J Med Chem *submitted*

   Evgeny Byvatov contributed to this work by performing virtual screening of commercially available compounds for ligands to the COX-2

8. **Improvement of the efficiency of lead based drug design by active learning**

Byvatov E., Schneider G.
in preparation

Evgeny Byvatov contributes to this work by developing an innovative approach for optimization of virtual screening by application of active learning methodology.

4 Introduction

4.1 The Drug Discovery process

Drug discovery and development is a creative, complex and highly regulated process. On average, it takes approximately 15 years to navigate a medication through the progression from the researcher laboratory to the patient. Although the development of new technologies has provided opportunities to significantly shorten that timeline, the process remains scientifically complex, and must be designed to take advantage of serendipity. [1]

Generating leads refers to the process of designing and synthesizing novel compounds with desired properties. Traditionally, it begins in a research laboratory by selecting and examining specific biological targets. These targets are disease-relevant, which means they play an important role in the progression of the disease or its symptoms. The desired targets should also be drugable, which refers to the ability to design or find small molecules that act on the target. [2] These chemical compounds, often referred to as “small molecules”, undergo extensive laboratory testing to determine their activity on the target. Those with biologically active structures, or structures that allow interaction with the biological target, become potential candidates for further study. Multiple compounds are usually identified and tested to determine which have the desired profile. Lead candidates are those with promising characteristics that are selected for further studies. Such molecules “lead the way” to develop new drugs.

The lead molecule is defined as a chemical entity that (a) already shows some of the desired properties, and (b) is small enough to be the core structure for the various chemical variations and additions to build analogs. [3]

In the past, researchers were limited by the number of leads they had access to and the speed that leads could be assessed.[4] With the advent of high-throughput technologies, the number of compounds and speed of assessment has significantly increased.

“Optimizing” the lead refers to the process used to manipulate the compound to improve its biological or therapeutic properties. Compounds are transformed into chemical structures that can be produced as the dosage forms for use in preclinical studies, which confirm the compound biological activity, safety, toxicology and pharmacokinetic profile. [5] Chemical leads are molecules with known structures which possess many, but not all of the properties described in the target product profile. The process of optimizing leads concludes with the selection of an Early Development Candidate (EDC), which is the drug candidate selected for more intense study. If a compound passes this step, applications are made to governmental regulatory authorities to request permission for human clinical testing. [6]

During clinical trials, test medication is administered to healthy volunteers and patients. (Phase I) This step is highly regulated by government agencies. Instructions for the conduct of clinical trials are specifically outlined in official documents, such as the Code of Federal Regulations in the US. [7]

The clinical trial process is separated into different phases, each with a specific objective. Initial human tests, called phase I studies, usually involve a small number (20 to 80) of healthy volunteers, and are conducted to determine dosing levels and assess the safety, tolerability, dose response and metabolic properties of the compound in humans.

In phase II studies, the drug is administered to a larger number (50 to 500) of subjects. Unlike phase I trials which involve healthy volunteers, phase II studies confirm the drug safety profile in patients diagnosed with the disease being studied. Phase II studies can be divided into two categories: phase IIa studies usually examine a variety of doses to identify the initial dosing regimen. Larger phase II studies, often referred to as phase IIb, confirm the safety in a larger patient population, and define the optimal dosing regimen. Because they are often double blinded, they may also provide preliminary data on the drug efficacy.

Phase III studies are much larger in scale, and gather additional information about the drug safety and effectiveness in the intended patient population. Depending on the therapeutic area, thousands of patients may be enrolled in studies that compare the drug being tested to one or more currently available therapies. The objective is to show statistical superiority via either improved efficacy or safety over current treatments. This critical endpoint is needed to obtain regulatory approval to market the drug.

In spite of the fact that phase III trials are one of the last important steps before requesting marketing approval, drug candidates are discontinued if the study results are negative. Discontinuing drug development at such a late stage in the process contributes to the enormous cost of bringing drugs to market. [7]

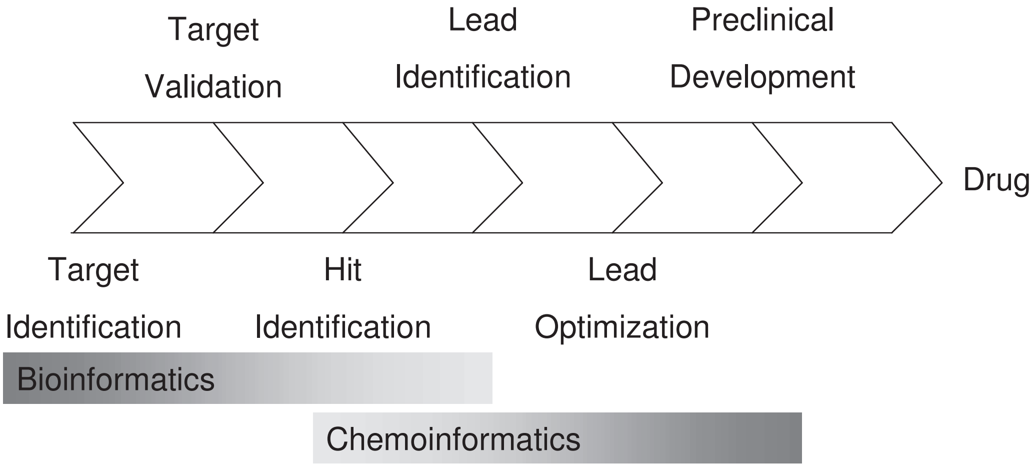


Figure 1. Overview of the early drug discovery process

We have described above overview of the drug development process, focusing on the Lead Identification and Preclinical Development. These are the main areas relevant for our research. **Figure 1** shows an overview of the early drug discovery process.

Computers have become much more powerful and cheaper over the last years, thereby allowing *in silico* screening using larger databases and more sophisticated algorithms. Using appropriate virtual screening techniques, it might become possible to predict properties like affinity to the target, absorption, distribution, metabolism, excretion or toxicity (ADME/Tox) at an earlier stage of the research pipeline, reducing expenditure per successful compound. Only the most promising compounds will then be synthesized and screened, potentially yielding a higher fraction of active structures in the selected subset and higher survival rates. Virtual screeining can allow reevaluation of the already existing databases. Some of these compounds might already possess the desired properties, which could be detected by, for instance, similarity searching. [8]

In this thesis I present the successful application of Machine Learning to the early virtual screening. Machine Learning existed as a separate field for many years. The first connection between this mathematical area and Life Science was done recently by Bioinformatics. The similar connection between Chemistry and Machine Learning was done after the first attempts to describe molecules with the help of descriptors.

In the beginning of this thesis I gave a short introduction to the drug discovery process. Virtual Screening can be applied in almost every pharmaceutical project: from the choice of the hits till ADME (Absorption, Distribution, Metabolism and Toxicity) filter. With virtual screening it is possible to significantly speed up these projects.

There is an introduction to the techniques that are used for describing molecules as a descriptor vector in Chapter 4.2. Descriptor vectors are the typical input for the Learning Machines like a Neural Network or the Support Vector Machine. We focus on the representations applied in our research. We have mostly calculated descriptors based on the 2D molecular graph. The typical examples are physicochemical properties, atom and bond counts, etc. (Chapter 4.2.1) as well as CATS descriptors (Chapter 4.2.2). The other types of descriptors are 3D based descriptors. Their success strongly depends on the conformer that is used during calculation of the 3D descriptors.

We have also applied molecular fingerprints as a method to describe molecules. Normally fingerprints are a poor choice, when a Neural Network is to be trained. The fingerprint vector is simply too long. In our case it was an exception. We applied fingerprints together with SVM. (Support Vector Machine). The advantage of the SVM in comparison to other methods is: it can be trained in a very high-dimensional space. We were the first who introduced this consecutive application of fingerprints and SVM in Chemoinformatics. (Chapter 7.7)

In Chapter 4.3 I focus on the on the SVM itself. Our predictions were almost always based on the trained SVM. SVM was our main tool. Due to the space limitation we usually made only brief introduction to SVM in all published articles. In order to compensate that we are giving here (in Chapter 4.3) the complete description of the SVM: SVM theory, optimal hyperplane, soft-margin hyperplane, quadratic programming as a technique to find the optimal hyperplane. We also include a discussion of the kernel functions that influence exact form of the separating surfaces.

In Chapter 4.4 we introduced Feature Selection methods that we used in our research. In this section we emphasise the difference between Filter and

Wrapper based approaches to the Feature Selection. In our article (Chapter 7.3) we compared advantages and disadvantages of the both methods in application to the Virtual Screening.

Chapter 7 contains publications that constitute the basis of this thesis.

Our first publication was a review article. (Chapter 7.1) In this article we give an overview of the applications of the SVM in Bio- and Chemoinfromatics. We performed a detailed comparison of SVM and Neural Networks in the next publication in order to justify usage of this method in our research. (Chapter 7.2)

Development of the new methods is described in Chapters 7.3 and 7.8. The first article is SVM-based feature selection technique. We selected features that are relevant to the certain biological activity of the molecule. The comparison of our methods with other standard methods demonstrated its superior performance. In the second article (Chapter 7.8) we applied Active Learning to the Virtual Screening.

In publications from the Chapters 7.5, 7.6 and 7.7 we applied virtual screening to the practical drug design. In publications from Chapter 7.6 we constructed an ADME filter. This filter was applied by Aventis Pharma to the early recognition of the potential CYP 2C9 ligands.

This thesis is an example of successful applications of virtual screening in Drug Design using Support Vector Machine.

4.2 From molecules to Descriptors

In this chapter we will describe methods that are used for transformation of the chemical structures of molecules to a form suitable for the analysis by Machine Learning algorithms. A typical way to represent molecules for this analysis is by descriptors. Descriptors are vectors in a high-dimensional space. Chemical properties of the molecules are mapped to the individual components of these vectors. We will mainly distinguish two types of descriptors, 2D and 3D-based. Calculation of the first type of descriptors is based only on the molecular graph. The second type required estimation of the possible 3D conformation of the underlying molecule.

Space limitation in the articles usually did not allow us to introduce descriptors that we used in sufficient details. This is the reason why we include this introduction here. Calculation of almost all descriptors was done using a standard MOE implementation. [9] Descriptors codes and complementary information is provided in the tables below.

We will treat separately CATS descriptors [10] and fingerprints. [11] Both of these types of molecular descriptors, in our research, were 2D based.

4.2.1 2D Molecular Descriptors

Descriptors that can be derived more or less directly from the molecular graph are commonly used for the generation quantitative structre activity relationaships (QSAR). In general, two-dimentional descriptors are computationally inexpensive, which make them attractive for high throughput applications.

2D molecular descriptors are defined to be numerical properties that can be calculated from the connection table representation of a molecule (*e.g.*, elements, formal charges and bonds, but not atomic coordinates). 2D descriptors are, therefore, not dependent on the conformation of a molecule and are most suitable for large database studies due to the speed of calculation.

3D-based descriptors, in contrary to 2D, are computationally more expensive and should be used only when their application is well justified.

4.2.1.1 Atom Counts and Bond Counts

The atom count and bond count descriptors are functions of the counts of atoms and bonds (subdivided according to various criteria). We used the MOE implementation. [9] (see **Table 1**)

Table 1. Atom Counts and Bond Counts Descriptors.

MOE-Code	Description
a_aro	Number of aromatic atoms.

`a_count`	Number of atoms (including implicit hydrogens). This is calculated as the sum of (1+h_i) over all non-trivial atoms i.
`a_heavy`	Number of heavy atoms #{Z_i|Z_i>1}.
`a_ICM`	Atom information content (mean). This is the entropy of the element distribution in the molecule (including implicit hydrogens but not lone pair pseudo-atoms). Let n_i be the number of occurrences of atomic number i in the molecule. Let p_i=n_i/n where n is the sum of the n_i. The value of `a_ICM` is the negative of the sum over all i of p_i log p_i.
`a_IC`	Atom information content (total). This is calculated to be `a_ICM` times n.
`a_nH`	Number of hydrogen atoms (including implicit hydrogens). This is calculated as the sum of h_i over all non-trivial atoms i plus the number of non-trivial hydrogen atoms.
`a_nB`	Number of boron atoms: #{Z_i|Z_i = 5}.
`a_nC`	Number of carbon atoms: #{Z_i|Z_i=6}.
`a_nN`	Number of nitrogen atoms: #{Z_i|Z_i=7}.
`a_nO`	Number of oxygen atoms: #{Z_i|Z_i=8}.
`a_nF`	Number of fluorine atoms: #{Z_i|Z_i=9}.
`a_nP`	Number of phosphorus atoms: #{Z_i|Z_i=15}.
`a_nS`	Number of sulfur atoms: #{Z_i|Z_i=16}.
`a_nCl`	Number of chlorine atoms: #{Z_i|Z_i=17}.
`a_nBr`	Number of bromine atoms: #{Z_i|Z_i=35}.
`a_nI`	Number of iodine atoms: #{Z_i|Z_i=53}.
`b_1rotN`	Number of rotatable single bonds. A bond is rotatable if it is not in a ring, and neither atom of the bond is such that (d_i+h_i)<2.
`b_1rotR`	Fraction of rotatable single bonds: `b_1rotN` divided by `b_count`.
`b_ar`	Number of aromatic bonds.
`b_count`	Number of bonds (including implicit hydrogens). This is calculated as the sum of (d_i/2 + h_i) over all non-trivial atoms i.
`b_double`	Number of double bonds. Aromatic bonds are not considered

	to be double bonds.
`b_heavy`	Number of bonds between heavy atoms.
`b_rotN`	Number of rotatable bonds. A bond is rotatable if it is not in a ring, and neither atom of the bond is such that $(d_i+h_i)<2$.
`b_rotR`	Fraction of rotatable bonds: `b_rotN` divided by `b_count`.
`b_single`	Number of single bonds (including implicit hydrogens). Aromatic bonds are not considered to be single bonds.
`b_triple`	Number of triple bonds. Aromatic bonds are not considered to be triple bonds.
`VAdjMa`	Vertex adjacency information (magnitude): $1 + \log_2 m$ where m is the number of heavy-heavy bonds. If m is zero, then zero is returned.
`VAdjEq`	Vertex adjacency information (equality): $-(1-f)\log_2(1-f) - f \log_2 f$ where $f = (n^2 - m) / n^2$, n is the number of heavy atoms and m is the number of heavy-heavy bonds. If f is not in the open interval (0,1), then 0 is returned.

4.2.1.2 Physical Properties

The physical properties descriptors capture electronic, lipophilic and steric characteristics of a molecule. [12] Very well known examples of such properties are molecular weight, octanol-water partitioning coefficient (LogP), the total energy, the ionization potential, the charge, the molecular refractability and hundreds of others. [13]

Many physical property descriptors (**Table 2**) can be easily measured, but for most of them good and fast computer algorithms exist that can to a certain extent replace the measurement. The following physical properties were used in this research (Chapters 7.2,7.3 and 7.6). They can be calculated by MOE [9] from the connection table (i.e without dependence on conformation) of a molecule.

Table 2. Physical Properties descriptors

MOE-Code	**Description**
`AM1_dipole`	The dipole moment calculated using the AM1 Hamiltonian [14].
`AM1_E`	The total energy (kcal/mol) calculated using the AM1 Hamiltonian [14].
`AM1_Eele`	The electronic energy (kcal/mol) calculated using the AM1 Hamiltonian [14].
`AM1_HF`	The heat of formation (kcal/mol) calculated using the AM1

	Hamiltonian [14].
AM1_IP	The ionization potential (kcal/mol) calculated using the AM1 Hamiltonian [14].
AM1_HF	The energy(eV) of the Lowest Unoccupied Molecular Orbital calculated using the AM1 Hamiltonian [14].
AM1_HOMO	The energy (eV) of the Highest Occupied Molecular Orbital calculated using the MOPAC AM1 Hamiltonian [14].
apol	Sum of the atomic polarizabilities (including implicit hydrogens) with polarizabilities taken from [15].
bpol	Sum of the absolute value of the difference between atomic polarizabilities of all bonded atoms in the molecule (including implicit hydrogens) with polarizabilities taken from [15].
density	Molecular mass density: Weight divided by vdw_vol.
FCharge	Total charge of the molecule (sum of formal charges).
MNDO_dipole	The dipole moment calculated using the MNDO Hamiltonian [14].
MNDO_E	The total energy (kcal/mol) calculated using the MNDO Hamiltonian [14].
MNDO_Eele	The electronic energy (kcal/mol) calculated using the MNDO Hamiltonian [14].
MNDO_HF	The heat of formation (kcal/mol) calculated using the MNDO Hamiltonian [14].
MNDO_IP	The ionization potential (kcal/mol) calculated using the MNDO Hamiltonian [14].
mr	Molecular refractivity (including implicit hydrogens). This property is calculated from an 11 descriptor linear model [16] with $r^2 = 0.997$, RMSE = 0.168 on 1,947 small molecules.
PM3_dipole	The dipole moment calculated using the PM3 Hamiltonian [14].
PM3_E	The total energy (kcal/mol) calculated using the PM3 Hamiltonian [14].
PM3_Eele	The electronic energy (kcal/mol) calculated using the PM3 Hamiltonian [14].

PM3_HF	The heat of formation (kcal/mol) calculated using the PM3 Hamiltonian [14].
PM3_IP	The ionization potential (kcal/mol) calculated using the PM3 Hamiltonian [14].
PM3_HF	The energy(eV) of the Lowest Unoccupied Molecular Orbital calculated using the PM3 Hamiltonian [14].
SMR	Molecular refractivity (including implicit hydrogens). This property is an atomic contribution model [17] that assumes the correct protonation state (washed structures). The model was trained on ~7000 structures and results may vary from the mr descriptor.
Weight	Molecular weight (including implicit hydrogens) with atomic weights taken from [15].
logP(o/w)	Log of the octanol/water partition coefficient (including implicit hydrogens). This property is calculated from a linear atom type model [18] with $r^2 = 0.931$, RMSE=0.393 on 1,827 molecules.
Reactive	Indicator of the presence of reactive groups. A non-zero value indicates that the molecule contains a reactive group. The table of reactive groups is based on the Oprea set [19] and includes metals, phospho-, N/O/S-N/O/S single bonds, thiols, acyl halides, Michael Acceptors, azides, esters, etc.
SlogP	Log of the octanol/water partition coefficient (including implicit hydrogens). This property is an atomic contribution model [17] that calculates logP from the given structure; i.e., the correct protonation state (washed structures). Results may vary from the logP(o/w) descriptor. The training set for SlogP was ~7000 structures.
TPSA	Polar surface area calculated using group contributions to approximate the polar surface area from connection table information only. The parameterization is that of Ertl *et al.* [20].
vdw_vol	van der Waals volume calculated using a connection table approximation.
vdw_area	Area of van der Waals surface calculated using a connection table approximation.

4.2.1.3 Adjacency and Distance Matrix Descriptors

Descriptors of this class are based on either the adjacency matrix or the topological distance matrix. The adjacency matrix *M* of a molecular graph consists of entries a_{ij} = 1 for adjacent vertices, and a_{ij} = a_{ji} = 0 otherwise. The entries d of the topological distance matrix *D* hold the minimal number of edges between vertex *i* and vertex *j*.

The adjacency matrix of $CH_3CH{=}O$ is displayed on the left of **Figure 2** and its distance matrix is displayed on the right (**Figure 2**):

$$\begin{array}{lccccccclccccccc}
\mathrm{C1} & 0 & 1 & 1 & 1 & 1 & 0 & 0 & \quad & 0 & 1 & 1 & 1 & 1 & 2 & 2 \\
\mathrm{H2} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \quad & 1 & 0 & 2 & 2 & 2 & 3 & 3 \\
\mathrm{H3} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \quad & 1 & 2 & 0 & 2 & 2 & 3 & 3 \\
\mathrm{H4} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \quad & 1 & 2 & 2 & 0 & 2 & 3 & 3 \\
\mathrm{C5} & 1 & 0 & 0 & 0 & 0 & 1 & 1 & \quad & 1 & 2 & 2 & 2 & 0 & 1 & 1 \\
\mathrm{H6} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \quad & 2 & 3 & 3 & 3 & 1 & 0 & 2 \\
\mathrm{O7} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \quad & 2 & 3 & 3 & 3 & 1 & 2 & 0
\end{array}$$

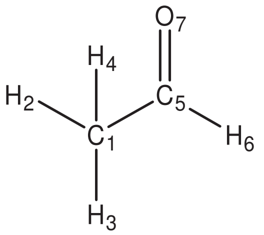


Figure 2. Calculation of the adjacent matrix using molecular graph representation.

Petitjean [21] defines the “eccentricity” of a vertex to be the longest path from that vertex to any other vertex in the graph. The graph “radius**”** is the smallest vertex eccentricity in the graph and the graph “diameter” as the largest vertex eccentricity. These values are calculated using the distance matrix and are used for several descriptors described below.

The following descriptors were calculated by MOE from the distance and adjacency matrices of the heavy atoms [9] used in the research projects described in chapters 7.2,7.3 and 7.6. (see **Table 3**)

Table 3. Adjacency and Distance Matrix Descriptors.

MOE-Code	Description
`balabanJ`	Balaban's connectivity topological index [22].
`Diameter`	Largest value in the distance matrix [21].
`petitjean`	Value of (`diameter` - `radius`) / `diameter`.
`petitjeanSC`	Petitjean graph Shape Coeffecient as defined in [21]: (`diameter` - `radius`) / `radius`.
`Radius`	If r_i is the largest matrix entry in row *i* of the distance matrix *D*, then the radius is defined as the smallest of the r_i [21].
`VDistEq`	If *m* is the sum of the distance matrix entries then `VdistEq` is defined to be the sum of $\log_2 m$ - $p_i \log_2 p_i$ / *m* where p_i is the number of distance matrix entries equal to *i*.
`VDistMa`	If *m* is the sum of the distance matrix entries then `VDistMa` is

	defined to be the sum of $\log_2 m - D_{ij} \log_2 D_{ij} / m$ over all i and j.
`weinerPath`	Wiener path number: half the sum of all the distance matrix entries as defined in [23] and [24].
`weinerPol`	Wiener polarity number: half the sum of all the distance matrix entries with a value of 3 as defined in [23].

4.2.1.4 Connectivity (Kier&Hall) and Kappa Shape Indices

Connectivity and Kappa Shape Indices were used in our following research projects: Chapters 7.2,7.3 and 7.6.

The introduction to this type of descriptors is included here, for their calculation the MOE package was applied. [9] For a heavy atom i let $v_i = (p_i - h_i) / (Z_i - p_i - 1)$ where p_i is the number of σ and π valence electrons of atom i. The Kier and Hall chi connectivity indices are calculated from the d_i and v_i values.Here h_i, Z_i and d_i are defined as follows:

1) Z denotes the *atomic number* of an atom. *Heavy atoms* are atoms that have an atomic number strictly greater than 1.
2) The *hydrogen count*, h, of an atom is the number of hydrogens to which it is (or should be) attached. This count includes all hydrogen atoms that are necessary to fill valence.
3) The *heavy degree, d*, of an atom is the number of heavy atoms to which it is bonded. That is, d is the number of bonded neighbors of the atom in the hydrogen suppressed graph.

The Kier and Hall kappa molecular shape indices [25] compare the molecular graph with minimal and maximal molecular graphs, and are intended to capture different aspects of molecular shape. In the following description (**Table 4**), n denotes the number of atoms in the hydrogen suppressed graph, m is the number of bonds in the hydrogen suppressed graph and a is the sum of $(r_i/r_c - 1)$ where r_i is the covalent radius of atom i, and r_c is the covalent radius of a carbon atom.

Table 4. Connectivity and Kappa Shape Indices.

MOE-Code	Description
`chi0`	Atomic connectivity index (order 0) from [25] and [26]. This is calculated as the sum of $1/\text{sqrt}(d_i)$ over all heavy atoms i with $d_i > 0$.
`chi0_C`	Carbon connectivity index (order 0). This is calculated as the sum of $1/\text{sqrt}(d_i)$ over all carbon atoms i with $d_i > 0$.
`chi1`	Atomic connectivity index (order 1) from [25] and [26]. This

	is calculated as the sum of 1/sqrt(d_id_j) over all bonds between heavy atoms *i* and *j* where $i < j$.
`chi1_C`	Carbon connectivity index (order 1). This is calculated as the sum of 1/sqrt(d_id_j) over all bonds between carbon atoms *i* and *j* where i < *j*.
`chi0v`	Atomic valence connectivity index (order 0) from [25] and [26]. This is calculated as the sum of 1/sqrt(v_i) over all heavy atoms *i* with $v_i > 0$.
`chi0v_C`	Carbon valence connectivity index (order 0). This is calculated as the sum of 1/sqrt(v_i) over all carbon atoms *i* with $v_i > 0$.
`chi1v`	Atomic valence connectivity index (order 1) from [25] and [26]. This is calculated as the sum of 1/sqrt(v_iv_j) over all bonds between heavy atoms *i* and *j* where $i < j$.
`chi1v_C`	Carbon valence connectivity index (order 1). This is calculated as the sum of 1/sqrt(v_iv_j) over all bonds between carbon atoms *i* and *j* where $i < j$.
`Kier1`	First kappa shape index: $(n\text{-}1)^2 / m^2$ [25].
`Kier2`	Second kappa shape index: $(n\text{-}1)^2 / m^2$ [25].
`Kier3`	Third kappa shape index: $(n\text{-}1)\ (n\text{-}3)^2 / p_3^{\ 2}$ for odd *n*, and $(n\text{-}3)\ (n\text{-}2)^2 / p_3^{\ 2}$ for even *n* [25].
`KierA1`	First alpha modified shape index: $s\ (s\text{-}1)^2 / m^2$ where $s = n + a$ [25].
`KierA2`	Second alpha modified shape index: $s\ (s\text{-}1)^2 / m^2$ where $s = n + a$ [25].
`KierA3`	Third alpha modified shape index: $(n\text{-}1)\ (n\text{-}3)^2 / p_3^{\ 2}$ for odd *n*, and $(n\text{-}3)\ (n\text{-}2)^2 / p_3^{\ 2}$ for even *n* where $s = n + a$ [25].
`KierFlex`	Kier molecular flexibility index: (`KierA1`) (`KierA2`) / *n* [25].
`Zagreb`	Zagreb index: the sum of d_i^2 over all heavy atoms *i*.

4.2.1.5 Descriptors of the Surface Areas

The Surface Area descriptors (**Table 5**) were calculated based on the estimation of the van der Waals surface area. For each atom the van der Waals surface area v_i was calculated together with the other atomic property, p_i. The v_i were calculated using a connection table approximation. Each descriptor in a series is defined to be the sum of the v_i over all atoms *i* such that p_i is in a specified range (*a*,*b*].

As p_i we used contribution to the logP(o/w) and Molecular Refractivity. In the descriptions to follow, L_i denotes the contribution to logP(o/w) for atom i as calculated in the `SlogP` descriptor [17]. R_i denotes the contribution to Molar Refractivity for atom i as calculated in the `SMR` descriptor [17]. The ranges were determined by percentile subdivision over a large collection of compounds as implemented in MOE. [9]

Table 5. Surface Area Descriptors.

MOE-Code	Description
`SlogP_VSA0`	Sum of v_i such that L_i <= -0.4.
`SlogP_VSA1`	Sum of v_i such that L_i is in (-0.4,-0.2].
`SlogP_VSA2`	Sum of v_i such that L_i is in (-0.2,0].
`SlogP_VSA3`	Sum of v_i such that L_i is in (0,0.1].
`SlogP_VSA4`	Sum of v_i such that L_i is in (0.1,0.15].
`SlogP_VSA5`	Sum of v_i such that L_i is in (0.15,0.20].
`SlogP_VSA6`	Sum of v_i such that L_i is in (0.20,0.25].
`SlogP_VSA7`	Sum of v_i such that L_i is in (0.25,0.30].
`SlogP_VSA8`	Sum of v_i such that L_i is in (0.30,0.40].
`SlogP_VSA9`	Sum of v_i such that L_i > 0.40.
`SMR_VSA0`	Sum of v_i such that R_i is in [0,0.11].
`SMR_VSA1`	Sum of v_i such that R_i is in (0.11,0.26].
`SMR_VSA2`	Sum of v_i such that R_i is in (0.26,0.35].
`SMR_VSA3`	Sum of v_i such that R_i is in (0.35,0.39].
`SMR_VSA4`	Sum of v_i such that R_i is in (0.39,0.44].
`SMR_VSA5`	Sum of v_i such that R_i is in (0.44,0.485].
`SMR_VSA6`	Sum of v_i such that R_i is in (0.485,0.56].
`SMR_VSA7`	Sum of v_i such that R_i > 0.56.

4.2.1.6 Pharmacophore Feature Descriptors

For our reseach we usually used chemical structures without alternation: we assumed that it is in the correct protonated form and other structurally relevant provided information is correct. This is true for the compounds from COBRA database [27] (Chapter 7.3,7.7,7.8) and some commercially available libraries

(Chapters 7.2,7.7,7.8). During the search for the ligands for dopamine D_3 receptor we manually checked the correct protonation form of the compounds. (Chapter 7.5)

This preprocessing of compounds is particularly relevant during calculation of the Pharmacophore Atom Types. Pharmacophore Atom Type descriptors (**Table 6**) consider only the heavy atoms of a molecule and assign a type to each atom. That is, hydrogens are suppressed during the calculation. The atom typing mechanism is a rule-based system for assigning pharmacophore features to atoms. The feature set was Donor, Acceptor, Polar (both Donor and Acceptor), Positive (base), Negative (acid), Hydrophobe and Other. Assignments may take into account implied protonation, deprotonation, keto/enol considerations and tautomerism. [9]

Table 6. Pharmacophore Features Descriptors.

MOE-Code	Description
a_acc	Number of hydrogen bond acceptor atoms (not counting acidic atoms but counting atoms that are both hydrogen bond donors and acceptors such as -OH).
a_acid	Number of acidic atoms.
a_base	Number of basic atoms.
a_don	Number of hydrogen bond donor atoms (not counting basic atoms but counting atoms that are both hydrogen bond donors and acceptors such as -OH).
a_hyd	Number of hydrophobic atoms.
vsa_acc	Approximation to the sum of VDW surface areas of pure hydrogen bond acceptors (not counting acidic atoms and atoms that are both hydrogen bond donors and acceptors such as -OH).
vsa_acid	Approximation to the sum of VDW surface areas of acidic atoms.
vsa_base	Approximation to the sum of VDW surface areas of basic atoms.
vsa_don	Approximation to the sum of VDW surface areas of pure hydrogen bond donors (not counting basic atoms and atoms that are both hydrogen bond donors and acceptors such as -OH).
vsa_hyd	Approximation to the sum of VDW surface areas of hydrophobic atoms.
vsa_other	Approximation to the sum of VDW surface areas of atoms

	typed as "other".
`vsa_pol`	Approximation to the sum of VDW surface areas of polar atoms (atoms that are both hydrogen bond donors and acceptors), such as -OH.

4.2.1.7 Partial Charge Descriptors

Descriptors that depend on the partial charge of each atom of a chemical structure require calculation of those partial charges. An unfortunate complication is the fact that there are numerous methods of calculating partial charges. The main difference between these variants is the source of the partial charges.

We used standard MOE implementation of the partial charge descriptors.[9] Let q_i denote the partial charge of atom i as defined above. Let v_i be the van der Waals surface area of atom i (as calculated by a connection table approximation). The following descriptors were calculated by MOE [9]. (see **Table 7**)

Table 7. Partial Charge Descriptors.

MOE-Code	Description
`PEOE_PC+`	Total positive partial charge: the sum of the positive q_i.
`PEOE_PC-`	Total negative partial charge: the sum of the negative q_i.
`PEOE_RPC+`	Relative positive partial charge: the largest positive q_i divided by the sum of the positive q_i.
`PEOE_RPC-`	Relative negative partial charge: the smallest negative q_i divided by the sum of the negative q_i.
`PEOE_VSA_POS`	Total positive van der Waals surface area. This is the sum of the v_i such that q_i is non-negative. The v_i are calculated using a connection table approximation.
`PEOE_VSA_NEG`	Total negative van der Waals surface area. This is the sum of the v_i such that q_i is negative. The v_i are calculated using a connection table approximation.
`PEOE_VSA_PPOS`	Total positive polar van der Waals surface area. This is the sum of the v_i such that q_i is greater than 0.2. The v_i are calculated using a connection table approximation.
`PEOE_VSA_PNEG`	Total negative polar van der Waals surface area. This is the sum of the v_i such that q_i is less than -0.2. The v_i are calculated using a connection table approximation.
`PEOE_VSA_HYD`	Total hydrophobic van der Waals surface area. This is the sum of the v_i such that $\|q_i\|$ is less than or equal to 0.2. The v_i are calculated using a connection table approximation.

`PEOE_VSA_POL`	Total polar van der Waals surface area. This is the sum of the v_i such that $\|q_i\|$ is greater than 0.2. The v_i are calculated using a connection table approximation.
`PEOE_VSA_FPOS`	Fractional positive van der Waals surface area. This is the sum of the v_i such that q_i is non-negative divided by the total surface area. The v_i are calculated using a connection table approximation.
`PEOE_VSA_FNEG`	Fractional negative van der Waals surface area. This is the sum of the v_i such that q_i is negative divided by the total surface area. The v_i are calculated using a connection table approximation.
`PEOE_VSA_FPPOS`	Fractional positive polar van der Waals surface area. This is the sum of the v_i such that q_i is greater than 0.2 divided by the total surface area. The v_i are calculated using a connection table approximation.
`PEOE_VSA_FPNEG`	Fractional negative polar van der Waals surface area. This is the sum of the v_i such that q_i is less than -0.2 divided by the total surface area. The v_i are calculated using a connection table approximation.
`PEOE_VSA_FHYD`	Fractional hydrophobic van der Waals surface area. This is the sum of the v_i such that $\|q_i\|$ is less than or equal to 0.2 divided by the total surface area. The v_i are calculated using a connection table approximation.
`PEOE_VSA_FPOL`	Fractional polar van der Waals surface area. This is the sum of the v_i such that $\|q_i\|$ is greater than 0.2 divided by the total surface area. The v_i are calculated using a connection table approximation.
`PEOE_VSA+6`	Sum of v_i where q_i is greater than 0.3.
`PEOE_VSA+5`	Sum of v_i where q_i is in the range [0.25,0.30).
`PEOE_VSA+4`	Sum of v_i where q_i is in the range [0.20,0.25).
`PEOE_VSA+3`	Sum of v_i where q_i is in the range [0.15,0.20).
`PEOE_VSA+2`	Sum of v_i where q_i is in the range [0.10,0.15).
`PEOE_VSA+1`	Sum of v_i where q_i is in the range [0.05,0.10).
`PEOE_VSA+0`	Sum of v_i where q_i is in the range [0.00,0.05).
`PEOE_VSA-0`	Sum of v_i where q_i is in the range [-0.05,0.00).
`PEOE_VSA-1`	Sum of v_i where q_i is in the range [-0.10,-0.05).

PEOE_VSA-2	Sum of v_i where q_i is in the range [-0.15,-0.10).
PEOE_VSA-3	Sum of v_i where q_i is in the range [-0.20,-0.15).
PEOE_VSA-4	Sum of v_i where q_i is in the range [-0.25,-0.20).
PEOE_VSA-5	Sum of v_i where q_i is in the range [-0.30,-0.25).
PEOE_VSA-6	Sum of v_i where q_i is less than -0.30.

4.2.2 Topological Cross-correlation Pharmacophores

Topological cross-correlation of generalized atom types is a simple molecular descriptor that leads to a molecular size independent description of potential pharmacophores. [10] The general idea of this representation scheme is to count the distances between atom pairs and then to regard the histogram of counts as a simplifying but exhaustive pharmacophore fingerprint of the molecule. Distances are expressed as the number of bonds along the shortest path connecting two nodes (non-hydrogen atoms) in the molecular graph. Each node is checked as to whether it can be assigned one of the following generalized atom types: hydrogen- bond donor (D), hydrogen-bond acceptor (A), positively charged (P), negatively charged (N), or lipophilic (L). Atom types were defined using SMILES [28] as follows: lipophilic (C(C)(C)(C)(C), Cl); positive ([+], NH2); negative ([-], COOH, SOOH, POOH); hydrogen-bond donor (OH, NH, NH2); hydrogen-bond acceptor: (O, N[!H]).

All possible node pairs are then checked: the numbers of all 15 possible pairs of generalized atom types (DD, DA, DP, DN, DL, AA, AP, AN, AL, PP, PN, PL, NN, NL, LL) are determined, and the resulting histogram counts are divided by the total number of non-hydrogen atoms to obtain scaled vectors. Distances of up to ten bonds were considered in the present studies, which led to a 150 (15x10) dimensional vector representation of a molecular compound.

4.2.3 3D Molecular Descriptors

Three-dimensional descriptors require the generation of molecular conformers prior to any computation that is related directly to the descriptor. This makes the handling of three-dimensional descriptors more complicated since the output is always dependent on the conformers.

Despite this fact we have used 3D molecular descriptors in our research. Sometimes information required for the correct prediction of the ligand activity may be extracted only from the 3D structure of the molecule, for instance, when biological activity of a molecule depends on the exact locations of the certain functional groups.

4.2.3.1 Potential Energy Descriptors

The energy descriptors use a potential energy model to calculate energetic quantities from stored 3D conformations.(**Table 8**) Most of the energy descriptors belong to the orientation independent class; that is, they depend on internal coordinates alone and not on an external reference frame. Descriptors that rely on an external reference frame are clearly indicated in the **Table 8.**

Table 8. Potential Energy Descriptors.

MOE-Code	Description
`E`	Value of the potential energy.
`E_ang`	Angle bend potential energy.
`E_ele`	Electrostatic component of the potential energy.
`E_nb`	Value of the potential energy with all bonded terms disabled.
`E_oop`	Out-of-plane potential energy.
`E_sol`	Solvation energy.
`E_stb`	Bond stretch-bend cross-term potential energy.
`E_str`	Bond stretch potential energy.
`E_strain`	Local strain energy: the current energy minus the value of the energy at a near local minimum. The current energy is calculated as for the `E` descriptor. The local minimum energy is the value of the `E` descriptor after first performing an energy minimization. Current chirality is preserved and charges are left undisturbed during minimization.
`E_tor`	Torsion (proper and improper) potential energy.
`E_vdw`	van der Waals component of the potential energy.
`E_rele`	Electrostatic interaction energy between the ligand and "receptor"
`E_rnb`	Non-bonded interaction energy between the molecule and a "receptor". It is similar to the other E_r* calls in that it is an interaction energy term.
`E_rsol`	Solvation free energy difference. Let *L* be the free energy of solvation of the molecule (ligand), *R* be the free energy of solvation of the atoms current (receptor), and *G* be the free energy of solvation of the *RL* complex. Consequently, the returned value is *G – L - R*.

E_rvdw	van der Waals interaction energy between the molecule and the atoms currently loaded.

4.2.3.2 Conformation Dependent Charge Descriptors

The following descriptors (**Table 9**) depend upon the partial charges of the molecules and their conformations. Accessible surface area (ASA) refers to the water accessible surface area using a probe radius of 1.4 Å. Let q_i denote the partial charge of atom *i*.

Table 9. Conformation Dependent Charge Descriptors.

MOE-Code	Description
ASA+	Water accessible surface area of all atoms with positive partial charge (strictly greater than 0).
ASA-	Water accessible surface area of all atoms with negative partial charge (strictly less than 0).
ASA_H	Water accessible surface area of all hydrophobic ($\|q_i\|<0.2$) atoms.
ASA_P	Water accessible surface area of all polar ($\|q_i\|>=0.2$) atoms.
DASA	Absolute value of the difference between ASA+ and ASA-.
CASA+	Positive charge weighted surface area, ASA+ times max { $q_i > 0$ } [29].
CASA-	Negative charge weighted surface area, ASA- times max { $q_i < 0$ } [29].
DCASA	Absolute value of the difference between CASA+ and CASA- [29].
dipole	Dipole moment calculated from the partial charges of the molecule.
dipoleX	The *x* component of the dipole moment (external coordinates).
dipoleY	The *y* component of the dipole moment (external coordinates).
dipoleZ	The *z* component of the dipole moment (external coordinates).
FASA+	Fractional ASA+ calculated as ASA+ / ASA.
FASA-	Fractional ASA- calculated as ASA- / ASA.

FCASA+	Fractional CASA+ calculated as CASA+ / ASA.
FCASA-	Fractional CASA- calculated as CASA- / ASA.
FASA_H	Fractional ASA_H calculated as ASA_H / ASA.
FASA_P	Fractional ASA_P calculated as ASA_P / ASA.

4.2.3.3 Surface Area, Volume and Shape Descriptors

The following descriptors depend on the structure connectivity and conformation. (**Table 10**)

Table 10. Surface Area, Volume and Shape Descriptors.

MOE-Code	Description
ASA	Water accessible surface area calculated using a radius of 1.4 A for the water molecule. A polyhedral representation is used for each atom in calculating the surface area.
dens	Mass density: molecular weight divided by van der Waals volume as calculated in the vol descriptor.
glob	Globularity, or inverse condition number (smallest eigenvalue divided by the largest eigenvalue) of the covariance matrix of atomic coordinates. A value of 1 indicates a perfect sphere while a value of 0 indicates a two- or one-dimensional object.
pmi	Principal moment of inertia.
pmiX	*x* component of the principal moment of inertia (external coordinates).
pmiY	*y* component of the principal moment of inertia (external coordinates).
pmiZ	*z* component of the principal moment of inertia (external coordinates).
rgyr	Radius of gyration.
std_dim1	Standard dimension 1: the square root of the largest eigenvalue of the covariance matrix of the atomic coordinates. A standard dimension is equivalent to the standard deviation along a principal component axis.
std_dim2	Standard dimension 2: the square root of the second largest eigenvalue of the covariance matrix of the atomic coordinates. A standard dimension is equivalent to the

	standard deviation along a principal component axis.
`std_dim3`	Standard dimension 3: the square root of the third largest eigenvalue of the covariance matrix of the atomic coordinates. A standard dimension is equivalent to the standard deviation along a principal component axis.
`vol`	van der Waals volume calculated using a grid approximation (spacing 0.75 A).
`VSA`	van der Waals surface area. A polyhedral representation is used for each atom in calculating the surface area.

4.2.4 Three Point Pharmacophore

The concept of a pharmacophore key was introduced by Sheridan and co-workers [30] as a means to account for the potential for intermolecular interactions in a 3D database search. Pharmacophore keys are 3D structural keys whose features include perspective macromolecular recognition sites. These sites include hydrogen bond donors, hydrogen bond acceptors, positively charged centers, aromatic ring centers, and hydrophobic centers. The pharmacophore itself is defined as a set of three centers forming a triangle. To generate the key, the pharmacophores exhibited by a particular conformation or ensemble of conformations are mapped onto appropriate bits in the binary set. This process is illustrated in **Figure 3**.

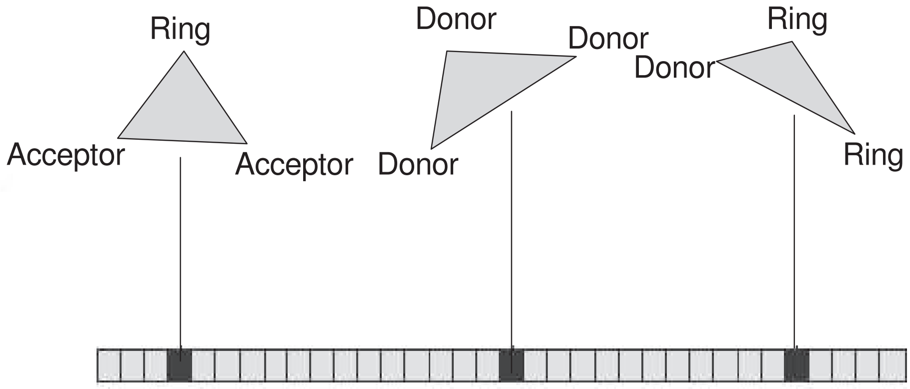


Figure 3. Three-point pharmacophore key generation. Each pharmacophore pattern present in the reference molecule is ‘projected’ onto a particular bit position, determined by the three ‘atom’ types and their mutual distances.

4.3 From Descriptors to the Predictive Model. Support Vector Machine

This section will describe Machine Learning technique that we used for constructing models for prediction of the biological activity of potential compound. As we have shown in the previous section chemical formula of the molecule is first transformed to the descriptor vector. Using this description we will then train the classifier with the compounds with measured biological activity. Typically we will mark compounds as active (class) or inactive (non-class). In this case we have a binary classifier. For the classification we used Support Vector Machine, which is described below.

More than 60 years ago R.A. Fisher [31] suggested the first algorithm for pattern recognition. He considered a model of two normal distributed populations, $N(\boldsymbol{m_1},\boldsymbol{\Sigma_1})$ and $N(\boldsymbol{m_2},\boldsymbol{\Sigma_2})$ of n-dimensional vectors $\boldsymbol{x}$ with mean vectors $\boldsymbol{m_1}$ and $\boldsymbol{m_2}$ and co-variance matrices $\boldsymbol{\Sigma_1}$ and $\boldsymbol{\Sigma_2}$, and showed that the optimal (Bayesian) solution is a quadratic decision function:

$$F_{sq} = sign\left[\frac{1}{2}(\mathbf{x}-\mathbf{m_1})^T \boldsymbol{\Sigma}_1^{-1}(\mathbf{x}-\mathbf{m}_1) - \frac{1}{2}(\mathbf{x}-\mathbf{m}_2)^T \boldsymbol{\Sigma}_2^{-1}(\mathbf{x}-\mathbf{m_2}) + \ln\frac{|\Sigma_2|}{|\Sigma_1|}\right] \tag{1}$$

In the case where $\boldsymbol{\Sigma_1} = \boldsymbol{\Sigma_2} = \boldsymbol{\Sigma}$ the quadratic decision function (1) degenerates to a linear function:

$$F_{sq} = sign\left[(\mathbf{m_1}-\mathbf{m_2})^T \boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}(\mathbf{m}_1^T\boldsymbol{\Sigma}^{-1}\mathbf{m}_1 - \mathbf{m}_2^T\boldsymbol{\Sigma}^{-1}\mathbf{m}_2)\right] \tag{2}$$

To estimate the quadratic decision function one has to determine (n(n+3))/2 free parameters. To estimate the linear function only n free parameters have to be determined. In the case where the number of observations is small (say less than $10n^2$) estimating $o(n^2)$ parameters is not reliable. Fisher therefore recommended, even in the case of $\boldsymbol{\Sigma_1} \neq \boldsymbol{\Sigma_2}$, to use the linear discriminator function (2) with $\boldsymbol{\Sigma}$ of the form:

$$\boldsymbol{\Sigma} = \tau\boldsymbol{\Sigma}_1 + (1-\tau)\boldsymbol{\Sigma}_2 \tag{3}$$

where τ is some constant. The optimal coefficient for τ was found in [32]. Fisher also recommended a linear decision function for the case where the two distributions are not normal. Algorithms for pattern recognition were therefore from the very beginning associated with the construction of linear decision surfaces.

In 1962 Rosenblatt [33] explored a different kind of learning machines: perceptrons or neural networks. The perceptron consists of connected neurons, where each neuron implements a separating hyperplane, so the perceptron as a whole implements a piecewise linear separating surface.

No algorithm that minimizes the error on a set of vectors by adjusting all the weights of the network was found in Rosenblatt's time, and Rosenblatt suggested a scheme where only the weights of the output unit were adaptive. According to the fixed setting of the other weights the input vectors are non-linearly transformed into the feature space, Z, of the last layer of units. In this space a linear decision function is constructed:

$$I(x) = sign\left(\sum \alpha_i \mathbf{z}_i(x)\right) \tag{4}$$

by adjusting the weights α_i from the ith hidden unit to the output unit so as to minimize some error measure over the training data. As a result of Rosenblatt's approach, construction of decision rules was again associated with the construction of linear hyperplanes in some space.

An algorithm that allows for all weights of the neural network to adapt in order locally to minimize the error on a set of vectors belonging to a pattern recognition problem was found in 1986 [34-36] when the back-propagation algorithm was discovered. The solution involves a slight modification of the mathematical model of neurons. Therefore, neural networks implement "piece-wise linear-type" decision functions.

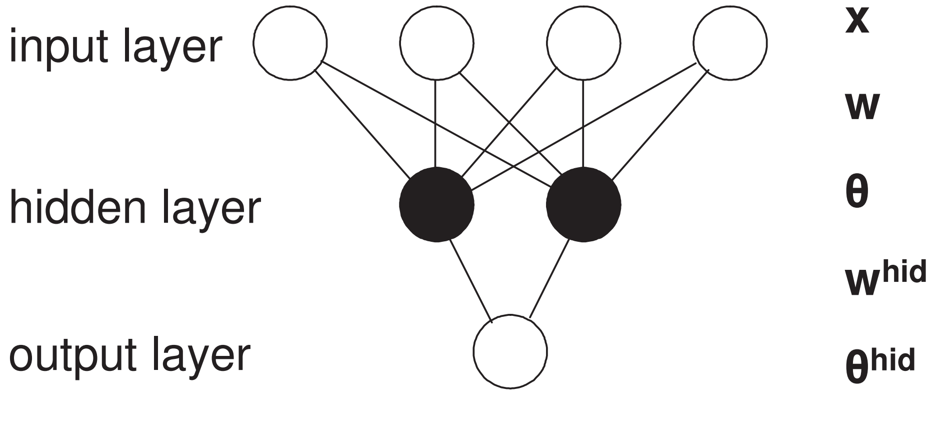


Figure 4. Three-layered Neural Network.

This was the invention of the artificial neural networks (ANN). ANN means three-layered, fully connected, feed-forward networks that can be trained to approximate any continious, non-linear function. **Figure 4** shows example of a simple ANN with four input neurons, two hidden neurons and one output neurons.

The input neuron simply represents the input vector ***x***, i.e. the descriptor variable of the compound that is to be predicted. All input neurons are connected to all hidden neurons of the second layer. Each connection has an assigned weight w, and each neuron of the hidden layer has an assigned ‘threshold’ *θ*. Analogously, the third layer is connected to the hidden layer vie the weigths w_h^{hid} and the output threshold θ^{hid}. The output *y* of an ANN can be expressed by

$$y = \sum_{h=1}^{H} w_h^{hid} T\left(\sum_{i=1}^{N} w_i x_i + \theta_h \right) + \theta^{hid}$$

where *H* is the number of hidden neurons and *N* is the number of input neurons, *T* is so-called ‘transfer function’. Diffeent transfer functions can be applied. [37] The typical training algorithm is usually variations of the back propagation methodology.

In contrast to ANN the Support Vector Machine (SVM) implements the following idea: it maps the input vectors into some high dimensional feature space *Z* through some non-linear mapping chosen a priori. In this space a linear decision surface is constructed with special properties that ensure high generalization ability of the machine.

The conceptual part of this problem was solved in 1965 [38] for the case of optimal hyperplanes for separable classes. An optimal hyperplane is here defined as the linear decision function with maximal margin between the vectors of the two classes. It was observed that to construct such optimal hyperplanes one only has to take into account a small amount of the training data, the so called “support vectors”, which determine this margin. It was shown that if the training vectors are separated without errors by an optimal hyperplane the expectation value of the probability of committing an error on a test example, *E*[Pr(*error*)] is bounded by the ratio between the expectation value of the number of support vectors and the number of training vectors [39]:

$$E[\Pr(error)] \leq \frac{E[\text{number of support vectors}]}{E[\text{number of training vectors}]} \tag{5}$$

Note that this bound does not explicitly contain the dimensionality of the space of separation. It follows from this bound, that if the optimal hyperplane can be constructed from a small number of support vectors relative to the training set size the generalization ability will be high—even in an infinite dimensional space. In Section 5 we will demonstrate that the ratio (5) for a real life problems can be as low as 0.03 and the optimal hyperplane generalizes well in a billion dimensional feature space.

Let $\mathbf{w} \bullet \mathbf{z} + b = 0$ be the optimal hyperplane in feature space. We will show, that the weights W_0 for the optimal hyperplane in the feature space can be written as some linear combination of support vectors

$$\mathbf{w} = \sum_{\substack{\text{support} \\ \text{vectors}}} \alpha_{\mathrm{i}} \mathbf{z}_i \ .$$

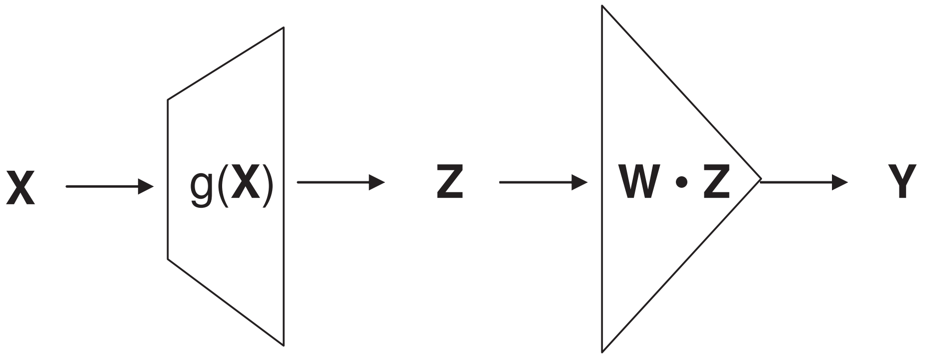


Figure 5. Illustration of the principles of SVM. First data $\boldsymbol{x}$ is mapped to a very high-dimentional space via $g(\boldsymbol{x})$: $z = g(\boldsymbol{x})$. Then the decision surface is $\boldsymbol{y} = \boldsymbol{w} \bullet z$ constructed in this very high-dimentional space.

The linear decision function *I (z)* in the feature space will accordingly be of the form:

$$I(z) = sign\left(\sum_{\substack{\text{support} \\ \text{vectors}}} \alpha_i \mathbf{z}_i \bullet \mathbf{z} + b_0 \right)$$

where $(z_i \bullet z)$ is the dot-product between support vectors z_i and vector z in feature space. The decision function can therefore be described as a two-layer network. However, even if the optimal hyperplane generalizes well the technical problem of how to treat the high-dimensional feature space remains. In 1992 it was shown [39], that the order of operations for constructing a decision function can be interchanged: instead of making a non-linear transformation of the input vectors followed by dot-products with support vectors in feature space, one can first compare two vectors in input space (by e.g. taking their dot-product or some distance measure), and then make a non-linear transformation of the value of the result (see **Figure 5**). This enables the construction of rich classes of decision surfaces, for example polynomial decision surfaces of arbitrary degree. This type of learning machine is called a support-vector machine.

The technique of Support Vector Machine was first developed for the restricted case of separating training data without errors.[40] Here we will describe two cases: separation without error is possible and separation without error is not possible.

4.3.1 Optimal hyperplane

In this section a review of the method of optimal hyperplanes [41] for separation of training data without errors will be given. In the next section the introduction of a notion of soft margins is described, that will allow for an analytic treatment of learning with errors on the training set.

4.3.1.1 The optimal hyperplane algorithm

The set of labeled training patterns

$$(y_1,\boldsymbol{x}_1),\,.\,.\,.\,.\,.\,.\,,(y_l,\boldsymbol{x}_l)$$

is said to be linearly separable if there exists a vector ***w*** and a scalar *b* such that the inequalities

$$\begin{aligned} (w\bullet\mathbf{x}_i)+w_0 &\geq +1 \qquad \text{if } y_i = +1\text{,} \\ (w\bullet\mathbf{x}_i)+w_0 &\leq -1 \qquad \text{if } y_i = \text{-}1 \end{aligned} \qquad (9)$$

are valid for all elements of the training set (8). Below we write the inequalities (9) in the form []:

$$y_i[(w\bullet x_i)+w_0]\geq 1 \qquad \text{i} = 1, \ldots, l \qquad (10)$$

The optimal hyperplane

$$(w\bullet x_i)+w_0=0 \qquad (11)$$

is the unique one which separates the training data with a maximal margin: it determines the direction ***w/|w|*** where the distance between the projections of the training vectors of two different classes is maximal. This distance *ρ (**w** , b)* is given by

$$\rho(\mathbf{w},b)=\min_{|x:y=1|}\frac{\mathbf{x}\bullet\mathbf{w}}{|\mathbf{w}|}=\max_{|x:y=-1|}\frac{\mathbf{x}\bullet\mathbf{w}}{|\mathbf{w}|}\,. \qquad (12)$$

The optimal hyperplane is the hyperplane with parameters *(**w**$_0$, b$_0$)* that maximize the distance (12). It follows from (12) and (10) that

$$\rho(\mathbf{w_0},b_0)=\frac{2}{|\mathbf{w_0}|}=\frac{2}{\sqrt{\mathbf{w_0}\bullet\mathbf{w_0}}} \qquad (13)$$

This means that the optimal hyperplane is the unique one that minimizes |***w**$_0$*| under the constraints (10). Constructing an optimal hyperplane is therefore a quadratic programming problem.

Vectors $\boldsymbol{x}_i$ for which $y_i(\boldsymbol{w} \bullet x_i + b) = 1$ will be termed support vectors. In section 4.3.5 we show that the vector $\boldsymbol{w}_0$ that determines the optimal hyperplane can be written as a linear combination of training vectors:

$$\mathbf{w_0}=\sum_{i=1}^{l} y_i\alpha_i^0\mathbf{x}_i \qquad (14)$$

where $\alpha_i^0 > 0$. Since $\alpha > 0$ only for support vectors (see Section 2.3.5), the expression (14) represents a compact form of writing $\boldsymbol{w_0}$. We also show that to find the vector of parameters α_i:

$$\Lambda_0^T = (\alpha_1^0 \ldots \alpha_l^0)$$

one has to solve the following quadratic programming problem:

$$W(\Lambda) = \Lambda^T 1 - \frac{1}{2}\Lambda^T \mathbf{D}\Lambda \qquad (15)$$

with respect to $\Lambda^T = (\alpha_1 \ldots \alpha_l)$, subject to the constraints:

$$\Lambda \geq 0 \qquad (16)$$

$$\Lambda^T \mathbf{Y} = 0 \qquad (17)$$

where $1^T = (1,\ldots,1)$ is a l-dimentional unit vector, $\mathbf{Y}^T = (y_1 \ldots y_l)$ is the l-dimentional vector of labels, and $\boldsymbol{D}$ is a symmetric $l \times l$ matrix with elements

$$D_{ij} = y_i y_j \mathbf{x_i} \bullet \mathbf{x_j}, \qquad i,j = 1,\ldots..,l. \qquad (18)$$

The inequality (16) describes the nonnegative quadrant. We therefore have to maximize the quadratic form (15) in the nonnegative quadrant, subject to the constraints (17).

When the training data (8) can be separated without errors we also show in Section 2.3.5. the following relationship between the maximum of the functional (15), the pair *(* Λ_0, b_0*)*, and the maximal margin ρ_0 from (13):

$$W(\Lambda_0) = \frac{2}{\rho_0^2} \qquad (19)$$

If some Λ_0 and large constant W the inequality

$$W(\Lambda_*) > W_0 \qquad (20)$$

is valid, one can accordingly assert that all hyperplanes that separate the training data have a margin

$$\rho < \sqrt{\frac{2}{W_0}}\,.$$

If the training set (8) cannot be separated by a hyperplane, the margin between patterns of the two classes becomes arbitrary small, resulting in the value of the functional *W(Λ)* turning arbitrary large. Maximizing the functional (15) under constraints (16) and (17) one therefore either reaches a maximum (in this case one has constructed the hyperplane with the maximal margin po), or one finds that the maximum exceeds some given (large) constant W_0 (in which case a separation of the training data with a margin larger then $\sqrt{1/W_0}$ is impossible).

The problem of maximizing functional (15) under constraints (16) and (17) can be solved very efficiently using the following scheme. Divide the training data into a number of portions with a reasonable small number of training vectors in each portion. Start out by solving the quadratic programming problem determined by the first portion of training data. For this problem there are two possible outcomes: either this portion of the data cannot be separated by a hyperplane (in which case the full set of data as well cannot be separated), or the optimal hyperplane for separating the first portion of the training data is found.

Let the vector that maximizes functional (15) in the case of separation of the first portion be Λ_1. Among the coordinates of vector Λ_1 some are equal to zero. They correspond to non-support training vectors of this portion. Make a new set of training data containing the support vectors from the first portion of training data and the vectors of the second portion that do not satisfy constraint (10), where **w** is determined by Λ_1. For this set a new functional $W_2(\Lambda)$ is constructed and maximized at Λ_2. Continuing this process of incrementally constructing a solution vector Λ_* covering all the portions of the training data one either finds that it is impossible to separate the training set without error, or one constructs the optimal separating hyperplane for the full data set, $\Lambda_* = \Lambda_0$. Note, that during this process the value of the functional $W(\Lambda)$ is monotonically increasing, since more and more training vectors are considered in the optimization, leading to a smaller and smaller separation between the two classes.

4.3.2 The Soft Margin hyperplane

Consider the case where the training data cannot be separated without error. In this case one may want to separate the training set with a minimal number of errors. To express this formally let us introduce some non-negative variables $\xi_i \geq 0$, $i = 1,...,l$. (**Figure 6**)

We can now minimize the functional

$$\Phi(\xi) = \sum_{i=1}^{l} \xi_i^{\sigma} \tag{21}$$

for small , subject to constrains

$$y_i(w \bullet x_i + b) \geq 1 - \xi_i \tag{22}$$

$$\xi_i \geq 0 \tag{23}$$

For sufficiently small σ the functional (21) describes the number of the training errors. [41]

Minimizing (21) one finds some minimal subset of training errors:

$$(y_{i1},x_{i1}),\ldots..,(y_{ik},x_{ik})$$

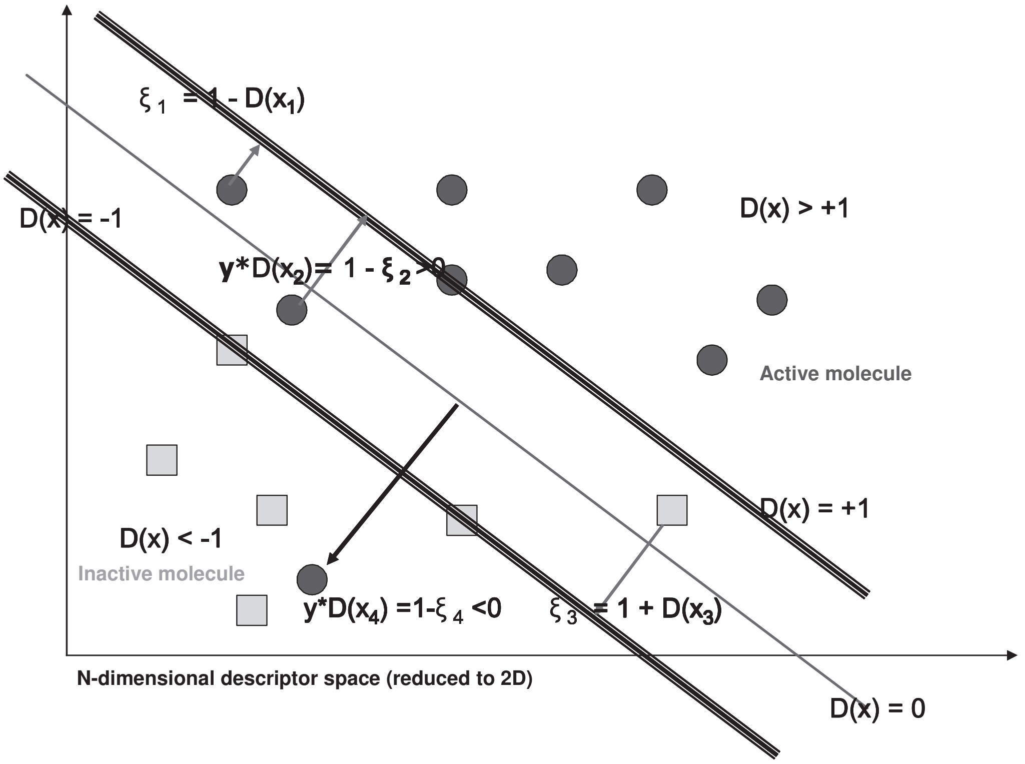


Figure 6. Construction of the optimal SVM plane. SVM identifies most optimal (linear or non-linear) hyperplane in n-dimensional descriptor space separating actives from inactives

If these data are excluded from the training set one can separate the remaining part of the training set without errors. To separate the remaining part of the training data one can construct an optimal separating hyperplane.

This idea can be expressed formally as: minimize the functional

$$\frac{1}{2}w^2 + CF\left(\sum_{i=1}^{l}\xi_i^{\sigma}\right) \tag{24}$$

subject to constraints (22) and (23), where *F(u)* is a monotonic convex function and *C* is a constant.

For sufficiently large *C* and sufficiently small a, the vector wo and constant b0, that minimize the functional (24) under constraints (22) and (23), determine the hyperplane that minimizes the number of errors on the training set and separate the rest of the elements with maximal margin.

Note, however, that the problem of constructing a hyperplane which minimizes the number of errors on the training set is in general NP-complete. To avoid NP-completeness of our problem we will consider the case of σ = 1 (the smallest value of σ for which the optimization problem (15) has a unique solution). In this case the functional (24) describes (for sufficiently large C) the problem of constructing a separating hyperplane which minimizes the sum of deviations, £, of training errors and maximizes the margin for the correctly classified vectors. If the training data can be separated without errors the constructed hyperplane coincides with the optimal margin hyperplane.

In contrast to the case with σ < 1 there exists an efficient method for finding the solution of (24) in the case of σ = 1. Let us call this solution the soft margin hyperplane. In Section 2.3.5 we consider the problem of minimizing the functional

$$\frac{1}{2}w^2 + CF\left(\sum_{i=1}^{l}\xi_i\right) \tag{25}$$

subject to the constraints (22) and (23), where F(u) is a monotonic convex function with F(0) = 0. To simplify the formulas we only describe the case of F(u) = u2 in this section. For this function the optimization problem remains a quadratic programming problem. In Section 2.3.5 we show that the vector **w**, as for the optimal hyperplane algorithm, can be written as a linear combination of support vectors ***x***$_i$:

$$\mathbf{w_0} = \sum_{i=1}^{l} y_i \alpha_i^0 \mathbf{x}_i$$

To find vector $\mathbf{\Lambda}^T = (\alpha_1, ..., \alpha_l)$ one has to solve the dual quadratic programming problem of maximizing

$$W(\Lambda, \delta) = \Lambda^T 1 - \frac{1}{2}\left(\Lambda^T \mathbf{D}\Lambda + \frac{\delta^2}{2}\right) \tag{26}$$

subject to constraints

$$\Lambda^T \mathbf{Y} = 0 \tag{27}$$

$$\delta \geq 0 \tag{28}$$

$$0 \leq \Lambda \leq \delta 1 \tag{29}$$

where are the same elements as used in the optimization problem for constructing an optimal hyperplane, δ is a scalar, and (29) describes coordinate-wise inequalities.

Note that (29) implies that the smallest admissible value δ in functional (26) is

$$\delta = \alpha_{\max} = \max(\alpha_1, \ldots\ldots, \alpha_l)$$

Therefore to find a soft margin classifier one has to find a vector Λ that maximize

$$W(\Lambda) = \Lambda^T 1 - \frac{1}{2}\left(\Lambda^T \mathbf{D} \Lambda + \frac{\alpha_{\max}^2}{2} \right) \qquad (30)$$

under the constraints $\Lambda > 0$ and (27). This problem differs from the problem of constructing an optimal margin classifier only by the additional term with amax in the functional (30). Due to this term the solution to the problem of constructing the soft margin classifier is unique and exists for any data set.

The functional (30) is not quadratic because of the term with amax. Maximizing (30) subject to the constraints $\Lambda > 0$ and (27) belongs to the group of so-called convex programming problems. Therefore, to construct a soft margin classifier one can either solve the convex programming problem in the l-dimensional space of the parameters Λ, or one can solve the quadratic programming problem in the dual l + 1 space of the parameters Λ and δ. In our experiments we construct the soft margin hyperplanes by solving the dual quadratic programming problem.

4.3.3 The Method of Convolution of the Dot-Product in Feature Space

The algorithms described in the previous sections construct hyperplanes in the input space.To construct a hyperplane in a feature space one first has to transform the n-dimensional input vector **x** into an N-dimensional feature vector through a choice of an N-dimensional vector function:

$$\phi : \Re^n \to \Re^N$$

An N-dimentional linear separator **w** and a bias b is then constructed for the set of transformed vectors:

$$\phi(\mathbf{x_i}) = \phi_1(\mathbf{x_i}), \phi_2(\mathbf{x_i}), \ldots, \phi_N(\mathbf{x_i}) \qquad \mathrm{i} = 1, \ldots\ldots, l$$

Classification of an unknown vector x is done by first transforming the vector to the separating space $\mathbf{x} : \mathbf{x} \to \phi(\mathbf{x})$ and then taking the sign of the function

$$f(\mathbf{x}) = w \bullet \phi(\mathbf{x}) + b \qquad (31)$$

According to the properties of the soft margin classifier method the vector w can be written as a linear combination of support vectors (in the feature space). That means

$$\mathbf{w} = \sum_{i=1}^{l} y_i \alpha_i \mathbf{x}_i \tag{32}$$

The linearity of the dot-product implies, that the classification function *f* in (31) for an unknown vector **x** only depends on the dot-products:

$$f(\mathbf{x}) = \phi(\mathbf{x}) \bullet \mathbf{w} + b = \phi(\mathbf{x}) \bullet \sum_{i=1}^{l} y_i \alpha_i \mathbf{x}_i + b \tag{33}$$

The idea of constructing support-vector macine comes from considering general forms of the dot-product in a Hilbert space [32]:

$$\phi(\mathbf{u}) \bullet \phi(\mathbf{v}) \equiv K(\mathbf{u}, \mathbf{v}) \tag{34}$$

According to the Hilbert-Schmidt Theory [42] any symmetric function K(**u**, **v**), with K(**u**, **v**) $\in L_2$, can be expanded in the form

$$K(\mathbf{u}, \mathbf{v}) = \sum_{i=1}^{\infty} \lambda_i \phi_i(\mathbf{u}) \bullet \phi_i(\mathbf{v}) \tag{35}$$

where $\lambda_i \in L_2$ and ϕ_i are eigenvalues and eigenfunctions

$$\int K(\mathbf{u}, \mathbf{v}) \phi_i(\mathbf{u}) d\mathbf{u} = \lambda_i \phi_i(\mathbf{v})$$

of the integral operator defined by the kernel *K(**u**, **v**)*. A sufficient condition to ensure that (34) defines a dot-product in a feature space is that all the eigenvalues in the expansion (35) are positive. To guarantee that these coefficients are positive, it is necessary and sufficient (Mercer's Theorem) that the condition

$$\iint K(\mathbf{u}, \mathbf{v}) g(\mathbf{u}) g(\mathbf{v}) d\mathbf{u} d\mathbf{v} > 0$$

is satisfied for all *g* such that

$$\int g^2(\mathbf{u}) d\mathbf{u} < \infty$$

Functions that satisfy Mercer's theorem can therefore be used as dot-products. Aizerman, Braverman and Rozonoer [43] consider a convolution of the dot-product in the feature space given by function of the form

$$K(\mathbf{u}, \mathbf{v}) = \exp\left(-\frac{|\mathbf{u} - \mathbf{v}|}{\sigma} \right) \tag{36}$$

which they call Potential Functions.

However, the convolution of the dot-product in feature space can be given by any function satisfying Mercer's condition; in particular, to construct a polynomial classifier of degree *d* in n-dimensional input space one can use the following function

$$K(\mathbf{u},\mathbf{v}) = (\mathbf{u} \bullet \mathbf{v} + 1)^d \qquad (37)$$

Using different dot-products $K(\mathbf{u},\mathbf{v})$ one can construct different learning machines with arbitrary types of decision surfaces [39]. The decision surface of these machines has a form

$$f(\mathbf{x}) = \sum_{i=1}^{l} y_i \alpha_i K(\mathbf{x}, \mathbf{x_i})$$

where $\boldsymbol{x_i}$ is the image of a support vector in input space and α_i is the weight of a support vector in the feature space.

To find the vectors $\boldsymbol{x_i}$ and weights α_i one follows the same solution scheme as for the original optimal margin classifier or soft margin classifier. The only difference is that instead of matrix ***D*** (determined by (18)) one uses the matrix

$$D_{i,j} = y_i y_j K(\mathbf{x_i}, \mathbf{x_j}) \qquad \text{i,j} = 1,\ldots..,l$$

4.3.4 General Features of Support Vector Machine

4.3.4.1 Decision Rules by SVM

To construct a support-vector machine decision rule one has to solve a quadratic optimization problem:

$$W(\Lambda) = \Lambda^T 1 - \frac{1}{2}\left(\Lambda^T \mathbf{D} \Lambda + \frac{\delta^2}{2}\right),$$

under the simple constraints:

$$0 \le \Lambda \le \delta 1$$
$$\Lambda^T \mathbf{Y} = 0$$

where matrix

$$D_{i,j} = y_i y_j K(\mathbf{x_i}, \mathbf{x_j}) \qquad \text{i,j} = 1,\ldots..,l$$

is determined by the elements of the training set, and $K(\mathbf{u},\mathbf{v})$ is the function determining the convolution of the dot-products.

The solution to the optimization problem can be found efficiently by solving intermediate optimization problems determined by the training data that currently constitute the support vectors. This technique is described in Section 4.3.2. The obtained optimal decision function is unique.

Each optimization problem can be solved using any standard techniques.

4.3.4.2 The Support Vector Machine is a universal machine

By changing the function $K(\mathbf{u},\mathbf{v})$ for the convolution of the dot-product one can implement different learning machines.

In the next section we will consider support-vector machines that use polynomial decision surfaces. To specify polynomials of different order *d* one can use the following functions for convolution of the dot-product

$$K(\mathbf{u},\mathbf{v}) = (\mathbf{u} \bullet \mathbf{v} + 1)^d$$

Radial Basis Function machines with decision functions of the form

$$f(\mathbf{x}) = sign\left(\sum_{i=1}^{n} \alpha_i \exp\left(-\frac{|\mathbf{x}-\mathbf{x}_i|^2}{\sigma^2} \right) \right)$$

can be implemented by using convolutions of the type

$$K(\mathbf{u},\mathbf{v}) = \exp\left(-\frac{|\mathbf{u}-\mathbf{v}|^2}{\sigma^2} \right)$$

In this case the support-vector machine will construct both the centers $\boldsymbol{x}_i$ of the approximating function and the weights α_i.

One can also incorporate a priori knowledge of the problem at hand by constructing special convolution functions. Support-vector machines are therefore a rather general class of learning machines which changes its set of decision functions simply by changing the form of the dot-product.

4.3.4.3 Generalization Ability

To control the generalization ability of a learning machine one has to control two different factors: the error-rate on the training data and the capacity of the learning machine as measured by its VC-dimension (Vapnik, 1982). There exists a bound for the probability of errors on the test set of the following form: with probability 1 - η;

$$\text{Pr (test error)} < \text{Frequency (training error)} + \text{Confidence interval} \qquad (38)$$

the inequality is valid. In the bound (38) the confidence interval depends on the VC-dimension of the learning machine, the number of elements in the training set, and the value of η.

The two factors in (38) form a trade-off: the smaller the VC-dimension of the set of functions of the learning machine, the smaller the confidence interval, but the larger the value of the error frequency.

A general way for resolving this trade-off was proposed as the principle of structural risk minimization: for the given data set one has to find a solution that minimizes their sum. A particular case of structural risk minimization principle is the Occam-Razor principle: keep the first term equal to zero and minimize the second one.

It is known that the VC-dimension of the set of linear indicator functions

$$I(\mathbf{x}) = sign((\mathbf{w} \bullet \mathbf{x}) + b), \qquad |\mathbf{x}| \le C_x$$

with fixed threshold *b* is equal to the dimensionality of the input space. However, the VC-dimension of the subset

$$I(\mathbf{x}) = sign((\mathbf{w} \bullet \mathbf{x}) + b) \qquad |\mathbf{x}| \le C_x \qquad |\mathbf{w}| \le C_w$$

(the set of functions with bounded norm of the weights) can be less than the dimensionality of the input space and will depend on C_w.

From this point of view the optimal margin classifier method executes an Occam-Razor principle. It keeps the first term of (38) equal to zero (by satisfying the inequality (9)) and it minimizes the second term (by minimizing the functional **w** • **w**). This minimization prevents an over-fitting problem.

However, even in the case where the training data are separable one may obtain better generalization by minimizing the confidence term in (38) even further at the expense of errors on the training set. In the soft margin classifier method this can be done by choosing appropriate values of the parameter C. In the support-vector machine algorithm one can control the trade-off between complexity of decision rule and frequency of error by changing the parameter C, even in the more general case where there exists no solution with zero error on the training set. Therefore the support-vector machine can control both factors for generalization ability of the learning machine.

4.3.5 Constructing Separating Hyperplanes

In this section we derive both the method for constructing optimal hyperplanes and soft margin hyperplanes.

4.3.5.1 Optimal hyperplane algorithm

It was shown in Section 4.3.1.1, that to construct the optimal hyperplane

$$\mathbf{w}_0 \bullet \mathbf{x} + b_0 = 0 \tag{40}$$

which separates a set of training data

$$(y_1,\mathbf{x_1}),\ldots,(y_l,\mathbf{x}_l)$$

one has to minimize a functional

$$\Phi = \mathbf{w} \bullet \mathbf{w}$$

subject to the constraints

$$y_i(w \bullet x_i + b) \geq 1, \qquad \text{i=1,. . . . .}l \tag{41}$$

To do this we use a standard optimization technique. We construct a Lagrangian

$$L(\mathbf{w},\mathbf{\Lambda},b) = \frac{1}{2}\mathbf{w} \bullet \mathbf{w} - \sum_{i=1}^{l} \alpha_i [y_i(w \bullet x_i + b) - 1] \tag{42}$$

where $\mathbf{\Lambda}^T = (\alpha_1, \ldots\ldots, \alpha_l)$ is the vector of non-negative Lagrange multipliers corresponding to the constraints (41).

It is known that the solution to the optimization problem is determined by the saddle point of this Lagrangian in the 2*l* + 1-dimensional space of ***w***, $\mathbf{\Lambda}$, and *b*, where the minimum should be taken with respect to the parameters w and b, and the maximum should be taken with respect to the Lagrange multipliers $\mathbf{\Lambda}$.

At the point of the minimum (with respect to ***w*** and *b*) one obtains:

$$\left.\frac{\partial L(\mathbf{w},\mathbf{\Lambda},b)}{\partial \mathbf{w}}\right|_{\mathbf{w}=\mathbf{w}_0} = \left(\mathbf{w}_0 - \sum_{i=1}^{l} \alpha_i y_i \mathbf{x_i}\right) = 0 \tag{43}$$

$$\left.\frac{\partial L(\mathbf{w},\mathbf{\Lambda},b)}{\partial b}\right|_{b=b_0} = \sum_{\alpha_i} y_i \mathbf{x_i} = 0 \tag{44}$$

From equality (43) we derive

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x_i} \tag{45}$$

which expresses, that the optimal hyperplane solution can be written as a linear combination of training vectors. Note, that only training vectors x, with ai > 0 have an effective contribution to the sum (45).

Substituting (45) and (44) into (42) we obtain

$$W(\mathbf{\Lambda}) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\mathbf{w}_0 \bullet \mathbf{w}_0 \qquad (46)$$

$$= \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \bullet \mathbf{x}_j \qquad (47)$$

In vector notation this can be rewritten as

$$W(\mathbf{\Lambda}) = \mathbf{\Lambda}^T 1 - \frac{1}{2}\mathbf{\Lambda}^T \mathbf{D}\mathbf{\Lambda} \qquad (48)$$

where ***1*** is an *l*-dimensional unit vector, and D is a symmetric t x ^-matrix with elements

$$\mathbf{D}_{ij} = y_i y_j \mathbf{x_i} \bullet \mathbf{x_j}$$

To find the desired saddle point it remains to locate the maximum of (48) under the constraints (43)

$$\mathbf{\Lambda}^T \mathbf{Y} = 0$$

where $\mathbf{Y}^T = (y_1, \ldots\ldots, y_l)$, and

$$\mathbf{\Lambda} \geq 0$$

The Kuhn-Tucker theorem plays an important part in the theory of optimization. According to this theorem, at our saddle point in ***w₀***, *b₀*, $\mathbf{\Lambda}_0$, any Lagrange multiplier α_i^0 and its corresponding constraint are connected by an equality

$$\alpha_i [y_i (\mathbf{x_i} \bullet w_0 + b_0) - 1] = 0, \qquad i = 1, \ldots, l$$

From this equality comes that non-zero values ai are only achieved in the cases where

$$y_i (\mathbf{x_i} \bullet w_0 + b_0) - 1 = 0$$

In other words: $\alpha_i \neq 0$ only for cases were the inequality is met as an equality. We call vectors $\boldsymbol{x_i}$ for which

$$y_i (\mathbf{x_i} \bullet w_0 + b_0) = 1$$

for support-vectors. Note, that in this terminology the Eq. (45) states that the solution vector $\boldsymbol{w_0}$ can be expanded on support vectors.

Another observation, based on the Kuhn-Tucker Eqs. (44) and (45) for the optimal solution, is the relationship between the maximal value W($\mathbf{\Lambda}_0$) and the separation distance *ρ0*:

$$\mathbf{w}_0 \bullet \mathbf{w}_0 = \sum_{i=1}^{l} \alpha_i^0 y_i \mathbf{x_i} \bullet \mathbf{w_0} = \sum_{i=1}^{l} \alpha_i^0 (1 - y_i b_0) = \sum_{i=1}^{l} \alpha_i^0$$

Substituting this equality into the expression (46) for W($\mathbf{\Lambda}_0$) we obtain

$$W(\Lambda_0) = \sum_{i=1}^{l} \alpha_i^0 - \frac{1}{2} \mathbf{w}_0 \bullet \mathbf{w_0} = \frac{\mathbf{w}_0 \bullet \mathbf{w_0}}{2}$$

Taking into account the expression (13) from Section 2 we obtain

$$W(\Lambda_0) = \frac{2}{\rho_0^2}$$

where *ρ0* is the margin for the optimal hyperplane.

4.3.5.2 Soft margin hyperplane Algorithm

Below we first consider the case of *F(u) = u*^{k}. Then we describe the general result for a monotonic convex function *F(u).*

To construct a soft margin separating hyperplane we maximize the functional

$$\Phi = \frac{1}{2} \mathbf{w} \bullet \mathbf{w} + C \left(\sum_{i=1}^{l} \xi_i \right)^k, \qquad \text{k} > 1$$

under the constraints

$$y_i (w \bullet x_i + b) \geq 1 - \xi_i, \qquad i = 1, \ldots, l \tag{49}$$

$$\xi_i \geq 0 \qquad i = 1, \ldots, l \tag{50}$$

The Lagrange functional for this problem is

$$L(\mathbf{w}, \xi, b, \mathbf{\Lambda}, \mathbf{R}) = \frac{1}{2} \mathbf{w} \bullet \mathbf{w} + C \left(\sum_{i=1}^{l} \xi_i \right)^k - \sum_{i=1}^{l} \alpha_i [y_i (w \bullet x_i + b) - 1 + \xi_i] - \sum_{i=1}^{l} r_i \xi_i \tag{51}$$

where the non-negative multipliers $\mathbf{\Lambda}^T = (\alpha_1, \ldots\ldots, \alpha_l)$ arise from the constraint (49), and the multipliers $\mathbf{R}^T = (r_1, \ldots\ldots, r_l)$ enforce the constraint (50).

We have to find the saddle point of this functional (the minimum with respect to the variables ***w****ᵢ*, *b*, and *ξᵢ*-, and the maximum with respect to the variables *αᵢ*, and *rᵢ*).

Let us use the conditions for the minimum of this functional at the extremum point:

$$\left.\frac{\partial L}{\partial \mathbf{w}}\right|_{\mathbf{w}=\mathbf{w}_0} = \mathbf{w}_0 - \sum_{i=1}^{l} \alpha_i y_i \mathbf{x_i} = 0 \tag{52}$$

$$\left.\frac{\partial L}{\partial b}\right|_{b=b_0} = \sum_{\alpha_i} y_i \alpha_i = 0 \tag{53}$$

$$\left.\frac{\partial L}{\partial \xi_i}\right|_{\xi_i=\xi_i^0} = kC\left(\sum_{i=1}^{l} \xi_i^0\right)^{k-1} - \alpha_i - r_i \tag{54}$$

If we denote

$$\sum_{i=1}^{l} \xi_i^0 = \left(\frac{\delta}{Ck}\right)^{\frac{1}{k-1}} \tag{55}$$

we can rewrite Eq. (54) as

$$\delta - \alpha_i - r_i = 0 \tag{56}$$

From the equalities (52)-(55) we find

$$\mathbf{w}_0 = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i$$

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \tag{57}$$

$$\delta = \alpha_i + r_i \tag{58}$$

Substituting the expressions for ***w₀,*** *b₀*, and *δ* into the Lagrange functional (51) we obtain

$$W(\mathbf{\Lambda},\delta) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \bullet \mathbf{x}_j - \frac{\delta^{k/k-1}}{(kC)^{1/k-1}}\left(1 - \frac{1}{k}\right) \tag{59}$$

To find the soft margin hyperplane solution one has to maximize the form functional (59) under the constraints (57)-(58) with respect to the non-negative variables α_i, ***rᵢ*** with i = 1,...,*l*. In vector notation (59) can be rewritten as

$$W(\mathbf{\Lambda}) = \mathbf{\Lambda}^T\mathbf{1} - \left[\frac{1}{2}\mathbf{\Lambda}^T\mathbf{D}\mathbf{\Lambda} + \frac{\delta^{k/k-1}}{(kC)^{1/k-1}}\left(1-\frac{1}{k}\right)\right] \tag{60}$$

where $\mathbf{\Lambda}$ and $\mathbf{D}$ are as defined above. To find the desired saddle point one therefore has to find the maximum of (60) under the constraints

$$\mathbf{\Lambda}^{\mathbf{T}}\mathbf{Y} = 0 \tag{61}$$

$$\mathbf{\Lambda} + \mathbf{R} = \delta\mathbf{1} \tag{62}$$

$$\mathbf{\Lambda} \geq 0 \tag{63}$$

and

$$\mathbf{R} \geq 0 \tag{64}$$

From (62) and (64) one obtains that the vector $\mathbf{\Lambda}$ should satisfy the conditions

$$0 \leq \mathbf{\Lambda} \leq \delta\mathbf{1} \tag{65}$$

From conditions (62) and (64) one can also conclude that to maximize (60)

$$\delta = \alpha_{\max} = \max(\alpha_1, \ldots\ldots, \alpha_l)$$

Substituting this value of S into (60) we obtain

$$W(\mathbf{\Lambda}) = \mathbf{\Lambda}^T\mathbf{1} - \left[\frac{1}{2}\mathbf{\Lambda}^T\mathbf{D}\mathbf{\Lambda} + \frac{\alpha_{\max}^{k/k-1}}{(kC)^{1/k-1}}\left(1-\frac{1}{k}\right)\right] \tag{66}$$

To find the soft margin hyperplane one can therefore either find the maximum of the quadratic form (51) under the constraints (61) and (65), or one has to find the maximum of the convex function (60) under the constraints (61) and (56). For the experiments reported in this paper we used *k = 2* and solved the quadratic programming problem (51).

For the case of *F(u) = u* the same technique brings us to the problem of solving the following quadratic optimization problem: minimize the functional

$$W(\mathbf{\Lambda}) = \mathbf{\Lambda}^T\mathbf{1} - \frac{1}{2}\mathbf{\Lambda}^T\mathbf{D}\mathbf{\Lambda}$$

under the constraints

$$0 \leq \mathbf{\Lambda} \leq C\mathbf{1}$$

and

$$\mathbf{\Lambda^T Y} = 0$$

The general solution for the case of a monotone convex function *F(u)* can also be obtained from this technique. The soft margin hyperplane has a form

$$\mathbf{w}_0 = \sum_{i=1}^{l} \alpha_i^0 y_i \mathbf{x_i}$$

where $\mathbf{\Lambda}_0^T = (\alpha^0,......,\alpha_l^0)$ is the solution of the following dual convex programming problem: maximize the functional

$$W(\mathbf{\Lambda}_0) = \mathbf{\Lambda}^T \mathbf{1} - \left[\frac{1}{2} \mathbf{\Lambda}^T \mathbf{D\Lambda} + \left(\alpha_{\max} f^{-1}\left(\frac{\alpha_{\max}}{C} \right) \right) - CF\left(f^{-1}\left(\frac{\alpha_{\max}}{C} \right) \right) \right]$$

under the constraints

$$\mathbf{\Lambda}^T \mathbf{Y} = 0$$
$$\mathbf{\Lambda} \geq 0$$

where we denote

$$f(u) = F^{'}(u)$$

For convex monotone functions *F(u)* with *F(0) = 0* the following inequality is valid:

$$uF^{'}(u) > F(u)$$

Therefore the second term in square brackets is positive and goes to infinity when $\alpha_{\max}$ goes to infinity.

Finally, we can consider the hyperplane that minimizes the form

$$\frac{1}{2}\left(\mathbf{w} \bullet \mathbf{w} + \sum_{i=1}^{l} \xi_i^2 \right)$$

subject to the constraints (49)-(50), where the second term minimizes the least square value for the errors. This lead to the following quadratic programming problem: maximize the functional

$$W(\mathbf{\Lambda}) = \mathbf{\Lambda}^T \mathbf{1} - \frac{1}{2}\left(\mathbf{\Lambda}^T \mathbf{D\Lambda} + \frac{1}{C} \mathbf{\Lambda}^T \mathbf{\Lambda} \right) \qquad (67)$$

in the non-negative quadrant $\mathbf{\Lambda} \geq 0$ subject to the constraint $\mathbf{\Lambda}^T \mathbf{Y} = 0$.

4.4 Selecting relevant descriptors

Descriptor selection or Feature selection plays an important role in Virtual Screening. Calculation of the irrelevant descriptors that are unimportant for the correct classification of the compound is time consuming. It limits chemical space that can be analysed by virtual screening by overloading available computational resources. Feature Selection can also simplify interpretability of the model. It is significantly easier to analyse model with lower number of parameters.

In this section we will first review the methods of Feature Selection and then describe in more details concept and techniques used in our research.

4.4.1 Feature selection methods

Feature selection or attribute selection has been a traditional research topic dating back to at least as early as the 70's (e.g. [44]). It is a broad subject that spans to research disciplines such as statistics [45], [63], [46], pattern recognition [47], [48], [49], data mining [50], machine learning [51], neural networks [52], fractals [53], rough sets theory [54], mathematical programming [55] [56] and many others.

The advantages of feature selection are that it reduces the dimensionality of the feature space and removes the redundant, irrelevant or noisy data. The immediate effects for data analysis tasks are speeding up the running time of the learning algorithms, improving the data quality, increasing the accuracy of the resulting model.

What is a feature selection? Suppose **X** is the original feature space with a cardinality of q, and $\mathbf{X_s}$ is the selected feature space with a cardinality of q_s, $\mathbf{X}_s \subseteq \mathbf{X}$, J($\mathbf{X_s}$) is the selection criterion for selected feature space $\mathbf{X_s}$. Without loss of generality, we assume that a higher value of J indicates a better feature space. The goal is to maximize J($\mathbf{X_s}$). Formally, the problem of feature selection is to find a sub-space $\mathbf{X}_s \subseteq \mathbf{X}$ such that

$$J(\mathbf{X}_s) = \max_{\mathbf{Z}\subset\mathbf{X},|\mathbf{Z}|=q_s} J(\mathbf{Z})$$

If an exhaustive approach is performed, then we need to consider all $\left(\frac{q}{q_s}\right)$ possible combinations. The number of combinations grows exponentially, making the exhaustive search unfeasible for larger values of q. Even for moderate values of q, performing the exhaustive search is impractical. Finding the best feature subset is usually intractable [57], and many problems related to feature selection have been shown to be NP-hard [58]. There are three kinds of feature selection strategies: (i) The number of features, say q_s is already given, and the task of the search algorithms is to decide which q_s features constitute a (sub)optimal feature subset. (ii) The second strategy is to search the smallest feature dimensionality for which the discrimination performance exceeds a specified value. (iii) The third search strategy selects a (sub)optimal feature subset which has a trade-off between the class discriminability (e.g. classification error rate) and the subset size (e.g. the number of selected features).

4.4.2 Relevance to the Concept: Weak and Strong Relevance

Determining which of the features are relevant to the learning task is a central issue in machine learning, as the inclusion of irrelevant or redundant features can reduce the performance of different learning algorithms. In order to determine which of the features are relevant or not, we need to first know the concepts of weak relevance and strong relevance. There are a number of different definitions in the machine learning literature for what it means for features to be ”relevant”. John, Kohavi and Pfleger [59] [60] define two notations of relevance [61]:

Strong Relevance: An attribute x_i is strongly relevant if its removal yields a deterioration of the performance of the Bayes Optimum Classifier.

Weak Relevance: An attribute x_i is weakly relevant if not strongly relevant and there exists a subset of variables *V* such that the performance on $V \cup \{x_i\}$ is better than the performance on *V*.

Therefore features that are neither strongly relevant nor weakly relevant are irrelevant. Irrelevant features should be left out.

4.4.3 General Characteristics of Feature Selection methods

Feature selection aims to search the relevant features in the feature space. Researchers have studied various aspects of feature selection. From the point of view of heuristic search, Blum and Langley [62] argue that the following four issues, which affect the nature of the search, can characterize any feature selection method.

1. The starting point in the feature space. Depending on which point to start with, the search direction will vary. Search from no features and successively add others is called forward selection. In contrast, search from all features and successively remove features is called backward selection. A third method could be to combine forward and backward search.
2. The organization of the search procedure. Obviously, if the number of features is too large, the exhaustive search of all the feature subspace is prohibitive, as there are 2^N possible combinations for *N* features. For example, heuristic search is more realistic than exhaustive search, but it does not guarantee finding the optimal solutions.
3. The evaluation strategy. How feature subsets are evaluated is an important problem. As for classification, the ideal feature subset should have the best separation of the data. Data separation is usually computed by an inter-class distance measure [63]. We usually used classification accuracy for the evaluation of the feature subset. Classification accuracy is defined as the percentage of test examples correctly classified by some algorithm.

Many induction algorithms incorporate a criterion based on information theory, others directly measure accuracy on the training set.

4. The criterion for stopping the search. During the process of evaluation, we might want to stop the search, when observing that there is no improvement of the classification accuracy.

4.4.4 Categorization Scheme of Feature Selection Methods

There is plenty of effort to compare and evaluate different feature selection methods [44] [64], but there are very few attempts to categorize the feature selection methods in the literature. Siedlecki and Sklansky [65] discussed the evolution of feature selection methods and grouped the methods into past, present and future categories. Their main focus was the branch and bound method and its variants. Dash and Liu [66] divided 32 existing feature selection methods into different groups based on the major two characteristics of feature selection: generation procedure (complete, heuristic and random) and evaluation function (distance, information, consistency, classification error rate). A taxonomy of feature selection algorithms into broad categories was given by Jain and Zongker [67], where the methods were first divided into those based on statistical pattern recognition (SPR) classification techniques, and those using artificial neural networks. The SPR category was then further divided into sub-categories. The categorization can also be simply done according to the monotonicity of the selection evaluation criteria, that is, monotonic versus non-monotonic. Another categorization could be according to the time complexity of the feature selection algorithm, e.g. the time complexity of floating search methods [68] is $O(2^n)$, while that of the sequential backward and sequential forward selection methods is $\Theta(n^2)$, where Θ denotes a tight estimate of complexity, while O denotes an estimate of complexity for which only an upper bound is known. Moreover, the feature selection methods can be categorized into two general groups [69], that is, the classifier-specific selection methods where the goodness is evaluated by a given criterion (e.g. the error rate of a certain classifier, this is useful for cases where we know which classification will be performed after selection) and the classifierindependent selection methods where the goodness is evaluated by the methods' own criterion (e.g. measures based on the approximation of class-conditional probability density functions, this is useful for cases where we don't know which classification will be used). Other categorization schemes include simply dividing the feature selection methods into: optimal (e.g. exhaustive search) vs. non-optimal (suboptimal), from the point of view of the optimality of the resulting subset; backward elimination vs. forward selection, from the point of view of starting point in the feature search space; and many others.

On the other hand, feature selection can be generally regarded as an optimization problem. For a general optimization problem, one may use the optimization tree category [70], that divides optimization techniques into discrete optimization and continuous optimization, both of which are then further divided into other subcategories. For more references on optimization, the reader is referred to Optimization Online [71].

We describe three typical model approaches in the following, i.e. the Filter Selection Model, the Wrapper Selection Model, and the Embedded Selection Model.

4.4.5 Filter Selection Method

The filter selection model is the earliest approach to feature selection. It utilizes an independent search criterion to find the appropriate feature subset before a machine learning algorithm is performed, thus it was termed as filter method by John, Kohavi and Pfleger [59]: it filters out irrelevant attributes before induction occurs, that is, the search is done independently of an induction algorithm. The procedure of the filter model is shown in **Figure 7**. The advantage of the filter model is that it does not need to re-run the algorithm for every induction algorithm when choosing to run on a reduced feature dataset, as a consequence, the filter approach is generally computational efficient, and it is practical for data sets with very high dimensionality.

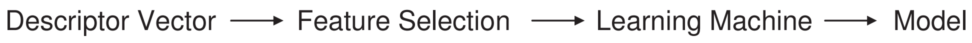


Figure 7. Filter-based feature selection.

There are a number of different representative filter algorithms in the literature. FOCUS, an algorithm designed by Almuallim and Dietterich [72] originally for 7the boolean domain, searches the feature space by looking at each feature in isolation, then turn to pairs of features, triples, and so on, and stops until it finds the minimal combination of features. The minimal feature subset divides the training data into pure classes, i.e. no instances have more than one class. The original training samples which are characterized by the resulting feature subset, are then passed to the decision tree induction algorithm ID3 [73].

Another representative work of the filter approach is the RELIEF algorithm due to Kira and Rendell [74]. The RELIEF algorithm follows the general and simple filter scheme, that is, it first evaluates the individual feature according to the evaluation criterion, and thereafter, the best n features are selected. However it uses a more complex evaluation function. The training samples, characterized by the selected features, are then passed to ID3. Two extensions were made to this algorithm by Kononenko [75], where more general data types can be treated. Although both FOCUS and RELIEF use the decision tree induction algorithm after feature selection, they are naturally not confined to decision tree algorithms, i.e. other induction algorithms can be used instead.

Since the filter approach does not take into account the learning bias introduced by the final induction algorithm, it may not be able to select the most suitable subset for the final induction algorithm. For this reason, the wrapper model was proposed.

4.4.5.1 Kolmogorov-Smirnov Statistics.

KS-based statistics represent a model-independent method for feature selection. It is routinely used for feature selection from different data sets and features. Its main advantage over other methods is the independence from the particular statistical model that generates the data, in contrast to other methods, that performwell only if the data adopts certain statistics. For instance, “correlation coefficient” [76] [77] based feature selection performs best if the data can be modeled by Gaussian mixtures,1 and its accuracy drops otherwise. Very often it is impossible to correctly guess statistical models of the data a priori, which results in only approximately correct models. If the underlying statistics is not known or a Gaussian mixture [78] model is not appropriate, KS statistics can be a method of choice.

In KS statistics each feature is first tested to have different statistics for class and nonclass samples. This is done by merging feature values for class and nonclass and building two separate cumulative fraction functions, one for class and one for nonclass. The cumulative fraction function represents the dependency of the percentage of samples whose feature values are below a certain threshold, on the position of the threshold value in the sorted list of feature values. An example of the cumulative function for the data set {0.08, 0.10, 0.15, 0.17, 0.24, 0.34, 0.38, 0.42, 0.49, 0.50, 0.70, 0.94, 0.95, 1.26, 1.37, 1.55, 1.75, 3.20, 6.98, 50.57} is given in **Figure 8**. The maximum difference D of two cumulative functions for class and nonclass is then used as a measure for the significance of a distinguishing feature. An example of this measure is given in **Figure 8**.

A KS statistics test is performed for all available features, which are then sorted with respect to the KS test results, and only the most relevant features are considered for further training.

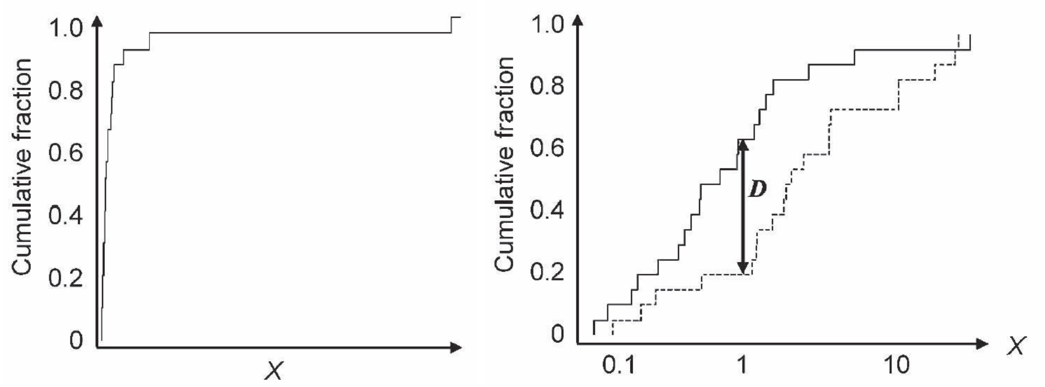


Figure 8. a) Example of the cumulative function b) illustration of the Kolmogorov-Smirnov Statistics.

4.4.6 Wrapper Selection Model

The strategy of the wrapper model is to use an induction algorithm to estimate the merit of the searched feature subset on the training data and using the estimated accuracy of the resulting classifier as its metric . The wrapper approaches often have better results than the filter approaches because they are tuned to the specific interaction between an induction algorithm and its training data. In this way, feature selection takes into account the biases from the final learning algorithm. The use of wrapper approaches was supported by the study of Aha and Banket [79], Doak [80] and John et al. [59].

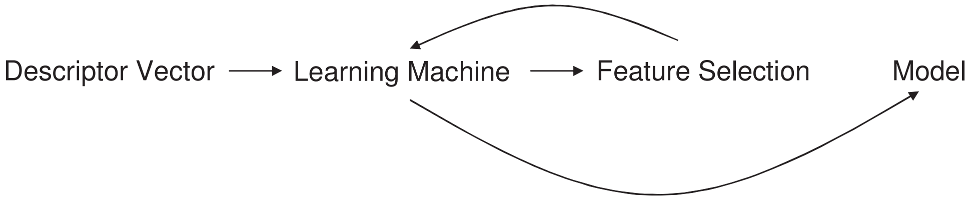


Figure 9. Wrapper-based approach to feature selection.

The wrapper selection procedure is illustrated in **Figure 9**.

The disadvantage of the wrapper model is that it is less tractable because of the prohibitive cost of running the classification algorithm many times when the dimensionality is considerably high.

4.4.6.1 SVM-Based wrapper for feature Selection

Usually feature selection algorithms are applied prior to the classifier training: A feature selection algorithm first selects a set of features and then a classifier is trained based on the features of this subset. Recently it was demonstrated that feature selection schemes, where the feature selection algorithm relies on the model that is created during training, produce better results. [57] Accordingly an alternative scheme for feature selection was suggested: The classifier is first trained using all available features. Then, the least important features are deleted. The drawback of this approach is that the trained classifier usually assumes a certain statistical model for the data, which might be only approximately correct. Current algorithms for nonlinear classifier training like artificial neural networks or SVM estimate a statistical model for the data sufficiently well to make this approach an alternative to modelindependent feature selection.

The separating surface generated by SVM is given by

$$f(\mathbf{x}) = \sum_i \alpha_i K(\mathbf{x}_i^{sv}, \mathbf{x}) + b$$

Here α_i, b, and $\boldsymbol{x}_i$ are parameters of the SVM, determined during training. $\mathbf{x}_i^{sv}$ are support vectors, which represent a subset of the training samples that determine the separating surface. This surface corresponds to the linear separation in a very high-dimensional space, where data points are mapped during SVM training. [81] This mapping is determined solely by the kernel function $K(\mathbf{x}, \mathbf{x}')$.[41] In this high-dimensional space the separating surface is given by

$$f(\mathbf{x}) = (\mathbf{w} \bullet \mathbf{x}) + b$$

where

$$w = \sum_i a_i \mathbf{x}_i^{sv}$$

is a normal vector of the separating hyperplane. To estimate the importance R_f of a feature to the accuracy of the SVM prediction we calculated a projection of the feature change in the mapped space to the normal of the SVM plane:

$$R_f = \frac{(\mathbf{w} \bullet \Delta\mathbf{x}_f)}{\Delta\mathbf{x}_f} = \frac{(\mathbf{w} \bullet \mathbf{x}_f^e) - (\mathbf{w} \bullet \mathbf{x}_f^b)}{\Delta\mathbf{x}_f} = \frac{\Delta f(\mathbf{x})}{\Delta\mathbf{x}_f} \to \frac{\partial f(\mathbf{x})}{\partial\mathbf{x}_f}$$

Calculating the derivative we obtain:

$$R_f(\mathbf{x}) = \frac{\partial f(\mathbf{x})}{\partial\mathbf{x}_f} = \sum_i a_i * \frac{\partial K(\mathbf{x}_i^{sv}, \mathbf{x})}{\partial\mathbf{x}_f} + b$$

For estimating the relevance of a feature to classification we should calculate R_f only in the vicinity of the separating hyperplane. To achieve it we will sum R_f only over support vectors, extending the principle of SVM that the position of the classifying hyperplane depends only on support vectors:

$$R_f = \sum_j R_f(\mathbf{x}_j^{sv}) = \sum_{i,j} a_i * \frac{\partial K(\mathbf{x}_i^{sv}, \mathbf{x}_j^{sv})}{\partial\mathbf{x}_f} + b$$

Empirically we observed that data normalization improved the performance in some cases; therefore, the final formula that we used to perform feature selection is

$$R_f = \sum_j R_f(\mathbf{x}_j^{sv}) = \sum_j \frac{\sum_i a_i * \frac{\partial K(\mathbf{x}_i^{sv}, \mathbf{x}_j^{sv})}{\partial\mathbf{x}_f} + b}{\sum_{i,k} a_i * \frac{\partial K(\mathbf{x}_i^{sv}, \mathbf{x}_j^{sv})}{\partial\mathbf{x}_k} + b}$$

Summarizing, R_f was calculated for all features, and those features with low R_f value were excluded from the features used for training. It is important to note that R_f depends only on the support vectors.

For constructing SVM models we used the SVM-light package. [82] A fifth-order polynomial kernel was used in SVM training: $K(\mathbf{x}, \mathbf{x}') = (s(\mathbf{x} \bullet \mathbf{x}') + 1)^5$. Training parameters s and C were optimized using a gradient decent-like algorithm to achieve maximum accuracy of prediction for the validation set. Parameter C is an internal parameter that is set prior to SVM training. It defines the tradeoff between the separating margin and the penalty for incorrect predictions. [81]

4.4.7 Embedded Selection Model

In contrast to the wrapper approach, which treats feature selection as a wrapper around the induction process, the embedded approach embeds the selection within the basic induction algorithm. Examples of this model are the decision tree algorithms ID3 and C4.56 7 by Quinlan [73] [83] and CART 8 by Breiman [84]. These decision tree algorithms use recursive partitioning methods for induction, and carry out a greedy search through the space of decision trees. At each stage they use an evaluation function to select the attribute that has the best ability to discriminate among the classes. They partition the training data based on this attribute and repeat the process on each subset, extending the tree downwards until no further discrimination is possible.

Besides these three approaches, another model called weighted model was also introduced [85], where feature weighting is considered.

5 Zusammenfassung

Die vorliegende Dissertation stellt eine kumulative Arbeit dar, die in insgesamt acht wissenschaftlichen Publikationen (fünf publiziert, zwei eingerichtet und eine in Vorbereitung) dargelegt ist. In diesem Forschungsprojekt wurden Anwendungen von maschinellem Lernen für das virtuelle Screening von Moleküldatenbanken durchgeführt. Das Ziel war primär die Einführung und Überprüfung des Support-Vector-Machine (SVM) Ansatzes für das virtuelle Screening nach potentiellen Wirkstoffkandidaten.

In der Einleitung der Arbeit ist die Rolle des virtuellen Screenings im Wirkstoffdesign beschrieben. Methoden des virtuellen Screenings können fast in jedem Bereich der gesamten pharmazeutischen Forschung angewendet werden. Maschinelles Lernen kann einen Einsatz finden von der Auswahl der ersten Moleküle, der Optimierung der Leitstrukturen bis hin zur Vorhersage von ADMET (Absorption, Distribution, Metabolism, Toxicity) Eigenschaften.

In Abschnitt 4.2 werden möglichen Verfahren dargestellt, die zur Beschreibung von chemischen Strukturen eingesetzt werden können, um diese Strukturen in ein Format zu bringen (Deskriptoren), das man als Eingabe für maschinelle Lernverfahren wie Neuronale Netze oder SVM nutzen kann. Der Fokus ist dabei auf diejenigen Verfahren gerichtet, die in der vorliegenden Arbeit verwendet wurden. Die meisten Methoden berechnen Deskriptoren, die nur auf der zweidimensionalen (2D) Struktur basieren. Standard-Beispiele hierfür sind physikochemische Eigenschaften, Atom- und Bindungsanzahl etc. (Abschnitt 4.2.1). CATS Deskriptoren, ein topologisches Pharmakophorkonzept, sind ebenfalls 2D-basiert (Abschnitt 4.2.2). Ein anderer Typ von Deskriptoren beschreibt Eigenschaften, die aus einem dreidimensionalen (3D) Molekülmodell abgeleitet werden. Der Erfolg dieser Beschreibung hangt sehr stark davon ab, wie repräsentativ die 3D-Konformation ist, die für die Berechnung des Deskriptors angewendet wurde.

Eine weitere Beschreibung, die wir in unserer Arbeit eingesetzt haben, waren Fingerprints. In unserem Fall waren die verwendeten Fingerprints ungeeignet zum Trainieren von Neuronale Netzen, da der Fingerprintvektor zu viele Dimensionen (~ 10^5) hatte. Im Gegensatz dazu hat das Training von SVM mit Fingerprints funktioniert. SVM hat den Vorteil im Vergleich zu anderen Methoden, dass sie in sehr hochdimensionalen Räumen gut klassifizieren kann. Dieser Zusammenhang zwischen SVM und Fingerprints war eine Neuheit, und wurde von uns erstmalig in die Chemieinformatik eingeführt.

In Abschnitt 4.3 fokussiere ich mich auf die SVM-Methode. Für fast alle Klassifikationsaufgaben in dieser Arbeit wurde der SVM-Ansatz verwendet. Ein Schwerpunkt der Dissertation lag auf der SVM-Methode. Wegen Platzbeschränkungen wurde in den beigefügten Veröffentlichungen auf eine detaillierte Beschreibung der SVM verzichtet. Aus diesem Grund wird in Abschnitt 4.3 eine vollständige Einführung in SVM gegeben. Darin enthalten ist eine vollständige Diskussion der SVM Theorie: optimale Hyperfläche, Soft-Margin-Hyperfläche, quadratische Programmierung als Technik, um diese optimale Hyperfläche zu finden. Abschnitt 4.3 enthält auch eine Diskussion von Kernel-Funktionen, welche die genaue Form der optimalen Hyperfläche bestimmen.

In Abschnitt 4.4 ist eine Einleitung in verschiede Methoden gegeben, die wir für die Auswahl von Deskriptoren genutzt haben. In diesem Abschnitt wird der Unterschied zwischen einer „Filter“- und der „Wrapper“-basierten Auswahl von Deskriptoren herausgearbeitet. In Veröffentlichung 3 (Abschnitt 7.3) haben wir die Vorteile und Nachteile von Filter- und Wrapper-basierten Methoden im virtuellen Screening vergleichend dargestellt.

Abschnitt 7 besteht aus den Publikationen, die unsere Forschungsergebnisse enthalten.

Unsere erste Publikation (Veröffentlichung 1) war ein Übersichtsartikel (Abschnitt 7.1). In diesem Artikel haben wir einen Gesamtüberblick der Anwendungen von SVM in der Bio- und Chemieinformatik gegeben. Wir diskutieren Anwendungen von SVM für die Gen-Chip-Analyse, die DNA-Sequenzanalyse und die Vorhersage von Proteinstrukturen und Proteininteraktionen. Wir haben auch Beispiele beschrieben, wo SVM für die Vorhersage der Lokalisation von Proteinen in der Zelle genutzt wurden. Es wird dabei deutlich, dass SVM im Bereich des virtuellen Screenings noch nicht verbreitet war.

Um den Einsatz von SVM als Hauptmethode unserer Forschung zu begründen, haben wir in unserer nächsten Publikation (Veröffentlichung 2) (Abschnitt 7.2) einen detaillierten Vergleich zwischen SVM und verschiedenen neuronalen Netzen, die sich als eine Standardmethode im virtuellen Screening etabliert haben, durchgeführt. Verglichen wurde die Trennung von wirstoffartigen und nicht-wirkstoffartigen Molekülen („Druglikeness“-Vorhersage). Die SVM konnte 82% aller Moleküle richtig klassifizieren. Die Klassifizierung war zudem robuster als mit dreilagigen feedforward-ANN bei der Verwendung verschiedener Anzahlen an Hidden-Neuronen. In diesem Projekt haben wir verschiedene Deskriptoren zur Beschreibung der Moleküle berechnet: Ghose-Crippen Fragmentdeskriptoren [86], physikochemische Eigenschaften [9] und topologische Pharmacophore (CATS) [10].

Die Entwicklung von weiteren Verfahren, die auf dem SVM-Konzept aufbauen, haben wir in den Publikationen in den Abschnitten 7.3 und 7.8 beschrieben. Veröffentlichung 3 stellt die Entwicklung einer neuen SVM-basierten Methode zur Auswahl von relevanten Deskriptoren für eine bestimmte Aktivität dar. Eingesetzt wurden die gleichen Deskriptoren wie in dem oben beschriebenen Projekt. Als charakteristische Molekülgruppen haben wir verschiedene Untermengen der COBRA Datenbank ausgewählt: 195 Thrombin Inhibitoren, 226 Kinase Inhibitoren und 227 Faktor Xa Inhibitoren. Es ist uns gelungen, die Anzahl der Deskriptoren von ursprünglich 407 auf ungefähr 50 zu verringern ohne signifikant an Klassifizierungsgenauigkeit zu verlieren. Unsere Methode haben wir mit einer Standardmethode für diese Anwendung verglichen, der Kolmogorov-Smirnov Statistik. Die SVM-basierte Methode erwies sich hierbei in jedem betrachteten Fall als besser als die Vergleichsmethoden hinsichtlich der Vorhersagegenauigkeit bei der gleichen Anzahl an Deskriptoren.

Eine ausführliche Beschreibung ist in Abschnitt 4.4 gegeben. Dort sind auch verschiedene „Wrapper“ für die Deskriptoren-Auswahl beschrieben.

Veröffentlichung 8 beschreibt die Anwendung von aktivem Lernen mit SVM. Die Idee des aktiven Lernens liegt in der Auswahl von Molekülen für das Lernverfahren aus dem Bereich an der Grenze der verschiedenen zu unterscheidenden Molekülklassen. Auf diese Weise kann die lokale Klassifikation verbessert werden. Die folgenden Gruppen von Moleküle wurden genutzt: ACE (Angiotensin converting enzyme), COX2 (Cyclooxygenase 2), CRF (Corticotropin releasing factor) Antagonisten, DPP (Dipeptidylpeptidase) IV, HIV (Human immunodeficiency virus) protease, Nuclear Receptors, NK (Neurokinin receptors), PPAR (peroxisome proliferator-activated receptor), Thrombin, GPCR und Matrix Metalloproteinasen. Aktives Lernen konnte die Leistungsfähigkeit des virtuellen Screenings verbessern, wie sich in dieser retrospektiven Studie zeigte. Es bleibt abzuwarten, ob sich das Verfahren durchsetzen wird, denn trotzt des Gewinns an Vorhersagegenauigkeit ist es aufgrund des mehrfachen SVM-Trainings aufwändig.

Die Publikationen aus den Abschnitten 7.5, 7.6 und 7.7 (Veröffentlichungen 5-7) zeigen praktische Anwendungen unserer SVM-Methoden im Wirkstoffdesign in Kombination mit anderen Verfahren, wie der Ähnlichkeitssuche und neuronalen Netzen zur Eigenschaftsvorhersage. In zwei Fällen haben wir mit dem Verfahren neuartige Liganden für COX-2 (cyclooxygenase 2) und dopamine D_3/D_2 Rezeptoren gefunden. Wir konnten somit klar zeigen, dass SVM-Methoden für das virtuelle Screening von Substanzdatensammlungen sinnvoll eingesetzt werden können.

Es wurde im Rahmen der Arbeit auch ein schnelles Verfahren zur Erzeugung großer kombinatorischer Molekülbibliotheken entwickelt, welches auf der SMILES Notation aufbaut. Im frühen Stadium des Wirstoffdesigns ist es wichtig, eine möglichst „diverse“ Gruppe von Molekülen zu testen. Es gibt verschiedene etablierte Methoden, die eine solche Untermenge auswählen können. Wir haben eine neue Methode entwickelt, die genauer als die bekannte MaxMin-Methode sein sollte. Als erster Schritt wurde die „Probability Density Estimation“ (PDE) für die verfügbaren Moleküle berechnet. [78] Dafür haben wir jedes Molekül mit Deskriptoren beschrieben und die PDE im *N*-dimensionalen Deskriptorraum berechnet. Die Moleküle wurde mit dem Metropolis Algorithmus ausgewählt. [87] Die Idee liegt darin, wenige Moleküle aus den Bereichen mit hoher Dichte auszuwählen und mehr Moleküle aus den Bereichen mit niedriger Dichte. Die erhaltenen Ergebnisse wiesen jedoch auf zwei Nachteile hin. Erstens wurden Moleküle mit unrealistischen Deskriptorwerten ausgewählt und zweitens war unser Algorithmus zu langsam. Dieser Aspekt der Arbeit wurde daher nicht weiter verfolgt.

In Veröffentlichung 6 (Abschnitt 7.6) haben wir in Zusammenarbeit mit der Molecular-Modeling Gruppe von Aventis-Pharma Deutschland (Frankfurt) einen SVM-basierten ADME Filter zur Früherkennung von CYP 2C9 Liganden entwickelt. Dieser nichtlineare SVM-Filter erreichte eine signifikant höhere Vorhersagegenauigkeit (q^2 = 0.48) als ein auf den gleichen Daten entwickelten PLS-Modell (q^2 = 0.34). Es wurden hierbei Dreipunkt-Pharmakophordeskriptoren eingesetzt, die auf einem dreidimensionalen Molekülmodell aufbauen. Eines der wichtigen Probleme im computerbasierten Wirkstoffdesign ist die Auswahl einer geeigneten Konformation für ein Molekül. Wir haben versucht, SVM auf dieses Problem anzuwenden. Der Trainingdatensatz wurde dazu mit jeweils mehreren Konformationen pro Molekül angereichert und ein SVM Modell gerechnet. Es wurden anschließend die Konformationen mit den am schlechtesten vorhergesagten IC_{50} Wert aussortiert. Die verbliebenen gemäß dem SVM-Modell bevorzugten Konformationen waren jedoch unrealistisch. Dieses Ergebnis zeigt Grenzen des SVM-Ansatzes auf. Wir glauben jedoch, dass weitere Forschung auf diesem Gebiet zu besseren Ergebnissen führen kann.

6 Lebenslauf

Evgeny Byvatov
Große Friedberger 21
Frankfurt am Main 60313 Germany

Geboren in Moskau am 17 April 1977,
Familienstand: ledig

9/1983-9/1991 Schule №779, Moskau

9/1991-9/1993 Gymnasium №2, Moskau

9/1993-10/1997 B. Sc. in theoretischer Physik und Mathematik. Moskauer Universität für Physik und Technologie (Russland)
Mittelwert der Noten: 4.8 von maximal 5.0 (*Summa Cum Laude - ausgezeichnet*)
Zusätzliches Studium der Mathematik an der Freien Universität Moskau (1993-1994)

10/1997-12/1999 M. Sc. in Life Science. The Weizmann Institute of Science (Israel)
Mittelwert der Noten: 90 von maximal 100

1/2000-11/2002 Weizmann Institute of Science (Israel). Department of Computer Science and Applied Mathematics.

11/2002-12/2004 Doktorarbeit an der Johann Wolfgang Goethe Universität Frankfurt
Betreuer: Prof. Dr. Gisbert Schneider
Stipendiat des Beilstein-Instituts zur Förderung der Chemischen Wissenschaften

Frankfurt, im Januar 2005

7 Publications

1. **SVM applications in bioinformatics**
   Byvatov E., Schneider G.
   Appl. Bioinformatics. 2003;2(2):67-77.

2. **Comparison of support vector machine and artificial neural network systems for drug/nondrug classification**
   Byvatov E., Fechner U., Sadowski J., Schneider G.
   J. Chem. Inf. Comput. Sci. 2003; 43(6):1882-9

3. **SVM based Feature Selection for Characterization of Focused Compound Collections**
   Byvatov E., Schneider G.
   J. Chem. Inf. Comput. Sci. 2004; 44(3):993-9

4. **SMILIB: Rapid assembly of combinatorial libraries in SMILES notation**
   Schüller A., Schneider G., **Byvatov E.**
   QSAR Comb. Sci. 2003; 22:719-721

5. **From Virtual to Real Screening for Novel D_3 Dopamine Receptor Ligands**
   Byvatov E., Sasse B.C., Stark H., Schneider G.
   ChemBioChem. *in press*

6. **Virtual Screening Filter to Identify Cytochrome P450 2C9 (CYP2C9) Inhibitors based on SVM for Model Building and Feature Visualization**
   Byvatov E., Matter H., Baringhaus K.H., Schneider G.
   J. Med. Chem.*, submitted*

7. **Extraction and visualization of pharmacophore models by SVM**
   Byvatov E., Franke L., Werz O., Steinhilber D., Schneider G.
   J Med Chem., *submitted*

8. **Improvement of the efficiency of lead based drug design by active learning**
   Byvatov E., Schneider G.
   in preparation

7.1 SVM applications in bioinformatics

Byvatov E., Schneider G.
Appl. Bioinformatics. 2003;2(2):67-77.

Support Vector Machine Applications in Bioinformatics

Evgeny Byvatov and Gisbert Schneider*

Johann Wolfgang Goethe-Universität
Institut für Organische Chemie und Chemische Biologie
Marie-Curie-Str. 11
D-60439 Frankfurt, Germany

Tel: +49 (0)69 79829821
Fax: +49 (0)69 79829826
Email: gisbert.schneider@modlab.de

* corresponding author

Abstract

The support vector machine (SVM) approach represents a data-driven method for solving classification tasks. It has been shown to produce lower prediction error compared to classifiers based on other methods like artificial neural networks, especially when large numbers of features are considered for sample description. In this review the theory and main principles of SVM are outlined, and successful applications in traditional areas of bioinformatics research are described. Current developments in SVM-related techniques are reviewed which might become relevant for future functional genomics and chemogenomics projects. In a comparative study, neural network and SVM models were developed for identification of small organic molecules potentially modulating the function of G-protein coupled receptors. The SVM system was able to correctly classify approximately 90% of the compounds in a cross-validation study yielding a Matthews correlation coefficient of 0.78. This classifier can be used for fast filtering of compound libraries in virtual screening applications.

Key words

Classification / GPCR / Neural network / Prediction / SVM / Virtual screening

Introduction to the theory of the Support Vector Machine

The support vector machine (SVM) approach for solving classification tasks was introduced by Vapnik (Cortes and Vapnik, 1995; Vapnik 1995). It has been successfully applied in various areas of research ever since. Currently we witness its growing popularity in the bioinformatics field. In this review, we describe basic principles of SVM and then give examples of successful applications, together with a new application, namely the prediction of G-protein coupled receptor (GPCR) ligands.

Several standard learning techniques are routinely used in bioinformatics. A basic task is binary classification. Typically, data are represented as labeled vectors in a high-dimensional space. This representation is often chosen to preserve as much information as possible about features that are responsible for correct classification of samples. The choice of a particular type of label depends on the classification task. For example, if the task is to separate membrane proteins from cytoplasmic proteins, labels might be chosen to be +1 for membrane proteins (“class”) and -1 for cytoplasmic proteins (“nonclass”). In this case, the features might be various descriptors that map properties of the molecules to real numbers. The classifier separating “class” from “nonclass” members may be conceptualized as a surface in this high-dimensional data space, structuring it into two parts, one for “class” and one for “nonclass”. Contrary to other approaches like, e.g., standard feed-forward neural networks, SVM does not construct a classifying surface directly in the given data space; instead, the sample points are projected to a significantly higher-dimensional space, where the separating surface can be found as a linear hyperplane. The corresponding surface in the original space is then presented as a result of SVM training. Generally speaking, SVM classifiers are generated by a two-step procedure: First, the data vectors are mapped (“projected”) to a high-dimensional space. The dimension of this space is significantly larger than dimension of the original data space. The algorithm finds a class-separating linear hyperplane in this high-dimensional space (Figure 1), and then this hyperplane is mapped back to the original data space.

We first start with the description of the algorithm for constructing the separating hyperplane in a very high-dimensional “mapped” space and then review the procedure for identifying the corresponding classifying surface in the original space.

[Figure 1]

The separating hyperplane is defined as

$$D(\mathbf{x}) = (\mathbf{w} \bullet \mathbf{x}) + w_0 \ ,$$

where $\boldsymbol{x}$ is a sample vector mapped to a high-dimensional space and $\boldsymbol{w}$ and w_0 are parameters of the hyperplane that SVM will estimate. The width of the margin can be expressed as a minimal τ for which for all vectors holds:

$$\frac{y_k D(\mathbf{x_k})}{\|\mathbf{w}\|} \geq \tau \ .$$

Here y_k are the class labels (+1 for class and -1 for nonclass membership of the sample x_k). Without loss of generality we can apply a constraint $\tau\|\mathbf{w}\| = 1$ to ***w***. In this case, maximizing τ is equivalent to minimizing $\|\mathbf{w}\|$ and SVM training is becoming the problem of finding the minimum of a function with the following constraints:

$$\textit{minimize} \qquad \eta(\mathbf{w}) = \frac{1}{2}(\mathbf{w} \bullet \mathbf{w}) \tag{1}$$

$$\textit{subject to constraints} \quad y_i[(\mathbf{w} \bullet \mathbf{x_i}) + w_0] \geq 1$$

This problem is solved by introducing Lagrange multipliers and minimizing the following function:

$$Q(\mathbf{w}, w_0, \alpha) = \frac{1}{2}(\mathbf{w} \bullet \mathbf{w}) - \sum_{i=1}^{n} \alpha_i \{y_i[(\mathbf{w} \bullet \mathbf{x_i}) + w_0] - 1\} \ .$$

Here a_i are Lagrange multipliers. Differentiating over ***w*** and w_i and substituting we obtain:

$$\max \quad Q(a) = \sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i,j=1}^{n} a_i a_j y_i y_j (\mathbf{x_i} \bullet \mathbf{x_j})$$

$$\textit{subject to constraints} \quad \sum_{i=1}^{n} y_i \alpha_i = 0 \, , \qquad \alpha_i \geq 0, i = 1,...,n$$

When perfect separation is not possible slack variables are introduced for sample vectors violating the edges of the margin, and the optimization problem can be reformulated:

$$\textit{minimize} \qquad \eta(w) = \frac{1}{2}(\mathbf{w} \bullet \mathbf{w}) + C\sum_i \xi_i \ . \tag{2}$$

$$\textit{subject to constraints} \quad y_i[(\mathbf{w} \bullet \mathbf{x_i}) + w_0] \geq 1 - \xi_{\text{i}} \ .$$

Here ξ_i are slack variables. *C* is a tradeoff between maximizing the width of the margin and minimizing slack variables. These variables are not equal to zero only for those vectors which are within the margin. Introducing Lagrange multipliers again we finally obtain:

$$\max \quad Q(a) = \sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i,j=1}^{n} a_i a_j y_i y_j (\mathbf{x_i} \bullet \mathbf{x_j})$$

subject to constraints $\sum_{i=1}^{n} y_i \alpha_i = 0\,,\quad C \geq \alpha_i \geq 0, i = 1,...,n\,.$

This is a quadratic programming (QP) problem and several efficient standard methods are known to solve it (Coleman and Li 1996). Due to the very high dimensionality of the QP problem, which typically arises during SVM training, an extension of the algorithm for solving QP is used in SVM applications (Joachims 1999).

A geometric illustration of the meaning of slack variables and Lagrange multipliers is given in Figure 1. The main goal of any classifier is to construct a hyperplane that correctly predict the class membership of a data vector. It is reasonable to assume that vectors which are lying close to the border separating two classes should play a more important role in determining the exact position of the separating hyperplane. The main advantage of the SVM-classifier versus others is that only vectors that are lying sufficiently close to the class border determine the exact position of the classifying hyperplane. These vectors are called support vectors. Their name assumes that only they "support" the constructed hyperplane. The exact distance to the separating hyperplane, which is used as a criterion to find the support vectors, is a parameter to maximize during SVM training. The larger this distance is, the more pronounced is the class separation by the hyperplane.

All other vectors are non-support vectors, where $y_i * D(\mathbf{x_i}) > 1$. These samples are correctly classified by the hyperplane and are located outside the separating margin (Figure 1). Slack variables and Lagrange multipliers for them are equal to zero, which reflects the fact that their positions do not influence location of the separating hyperplane. Parameters of the hyperplane do not depend on them, and even if their positions are changed the separating hyperplane and margin will remain unchanged, provided that these points will stay outside the margin.

Support vectors can be tentatively divided into two groups: vectors lying on the border of separating hyperplane, and vectors lying within the separating hyperplane. The origin of this tentative division is historical. Two types of SVM exist, "hard-margin" (Equation 1) and "soft-margin" (Equation 2) systems. In hard-margin SVM classifiers no support vectors are allowed within the separating margin and the SVM is trained to maximize the separating margin. In soft-margin SVM classifier, which is our main focus here, a trade-off is introduced between having a large separating margin and a minimal number of misclassifications. This means that some of the vectors are tolerated within the separating margin for the benefit of having a larger margin. This trade-off is introduced by parameter *C*, which should be optimized to achieve maximum classification accuracy. Sometimes it is not possible to find a separating hyperplane even in a very high-dimensional space. In this case a tradeoff is introduced between the size of the separating margin and penalties for every vector which is within the margin (Cortes and Vapnik 1995).

As illustrated in Figure 1, for all support vectors the absolute values of the slack variables are equal to the distances from these points to the edge of the separating margin. For support vectors lying on the border of separating hyperplane slack variables are equal to zero. For other support vectors these distances determine to which extent they violate the margin. They are defined in units of half of the width of the separating margin. For correctly classified vectors within the separating margin, slack variable values are between zero and one. For misclassified vectors within the margin the values of the slack variables are between one and two. For other misclassified points they are greater than two.

For vectors lying on the edge of margin, Lagrange multipliers are between zero and *C*, slack variables for them are still equal to zero. For all support vectors, for which the values of slack variables are larger than zero, Lagrange multipliers are equal to *C*.

An important feature of SVM is the usage of Kernel functions rendering explicit mapping of the data to a very high-dimensional space unnecessary. In other words, scalar products of each vector pair are calculated in the original data space by introducing a kernel function $(\mathbf{x_i} \bullet \mathbf{x_j}) = K(\mathbf{x_i}, \mathbf{x_j})$. The kernel function defines the scalar product in a certain high-dimensional space if it satisfies the following conditions (Cristianini and Shawe-Taylor 2000):

1. K(**x**,**x'**) takes its maximum value when **x'** = **x**.

2. |K(**x**,**x'**)| decreases with |**x**-**x'**|.

Here **x** and **x'** are vectors in the original space for which a kernel function is defined that corresponds to a scalar product in the mapped high-dimensional space.

Various kernels may be applied (Burges 1998). For many applications a polynomial kernel functions has been shown to be sufficient, e.g. a fifth-order polynomial-based kernel:

$$K(\mathbf{x}, \mathbf{x'}) = ((\mathbf{x} \bullet \mathbf{x'}) * s + r)^5$$

where *s* and *r* are kernel parameters. This kernel corresponds to the decision function:

$$f(\mathbf{x}) = sign\left(\sum_i a_i * K(\mathbf{x}_\mathbf{i}^{\mathbf{sv}}, \mathbf{x}) + b \right),$$

Here a_i are Lagrange multipliers determined during SVM training. The sum is only over support vectors x^{sv}. Lagrange multipliers for all other points are equal to zero. Parameter *b* determines the shift of the hyperplane, which is also found during SVM training.

Kernel parameter and the error trade-off *C* (*vide supra*) should be tuned, e.g. by multiple cross-validation of training data. Basically, the following procedure is applied. The data set is divided into two parts, training and test set. The test subset is put aside and is used only for estimation of the performance of the trained classifier. Training data are divided into *k* non-overlapping subsets. First, the parameters to be determined are set to initial reasonable values. Then, the SVM is trained on the whole training data excluding the *k* subset and the performance of the obtained SVM classifier is estimated with the excluded *k* subset. This procedure is repeated for each subset, and so an average performance of the SVM classifier will be obtained. This optimization can be performed, e.g. by simple heuristics based on gradient descent methods (Bishop 1995). The optimized values are employed for final training with the complete training data. The kernel parameters as well as the trade-off between the size of the separating margin and errors in predictions, *C*, are usually predefined prior to final training. A recommended starting value for *C* is

$$\frac{1}{< x^2 >} \text{ (Vapnik 1995).}$$

Currently available software packages allow to use SVM as a “black box”, although some basic knowledge of SVM theory will help to obtain optimal results.

If more than a binary classification is required, i.e. a multi-class problem, *N* different subclasses will be predicted by SVM, and a multi-step binary classification is conducted: the data are divided into class one versus all other classes, class two versus all other etc. Then SVM models are obtained for each such pair. For *N* classes *N* different SVMs are trained. For final prediction a jury is formed by all *N* SVMs.

SVM algorithms are capable to solve large-scale multidimensional problems. Data containing up to several tens of thousands of samples can be efficiently analyzed by SVM. It has been estimated that the computational time for SVM training relates approximately with n^2 to the number of training samples (Joachims 1999).

Applications of the Support Vector Machine in bioinformatics

SVM for DNA chip analysis

SVM was successfully employed for analysis of gene expression using microarray data. Research in this area has two primary goals.

First, given the gene expression profile predict if it corresponds to certain physiological conditions, like classification of cancer versus non-cancer tissues (Furey et al. 2000). Using SVM in a combination with a feature selection scheme it was possible to identify small groups of genes, which could serve as markers to classify tissue as cancerous (Guyon et al. 2002) .

The second main direction is to create an alternative to conventional clustering methods by grouping genes with similar functions. When applied to gene expression data, SVM training begins with a set of genes that have a known common function (“positive class”). A separate set of genes that are known not to be members of this functional class is also specified (“negative class”). These two sets of genes are combined to form a training set, where genes are labeled positively if they are in the functional class and negatively if they are not in the functional class. SVM for functional characterization of genes was shown to produce more accurate predictions than other mostly unsupervised learning methods like self-organizing maps and *k*-means clustering (Brown et al. 2000). Recently this approach was further extended by adding phylogenic information about genes and applying different kernel functions (Pavlidis et al. 2002).

SVM has several mathematical features that that make it an attractive technique for gene expression analysis, including its flexibility in choosing the classifier function (SVM kernel), sparseness of the solution when dealing with large data sets, and explicit identification of outliers. It also provides an excellent classification performance compared to other methods like Parzen windows, decision trees, and Fisher linear discriminant (Brown et al. 2000).

SVM for protein structure prediction

Recently, SVM was applied for protein fold prediction. Given the amino acid sequence of a protein SVM was trained to predict the overall protein fold (Cai et al. 2002; Cai et al. 2002; Ding and Dubchak 2001). Proteins were described by features representing amino acid composition, hydrophobicity, normalized van der Waals volume, polarity, polarizability, and other properties. Performance of the classification was evaluated on classes of protein folds from the SCOP database (Lo Conte et al. 2000). Large classes of fold types like all-α, all-β, α/β or α+β were predicted with a high accuracy yielding 80-95% correct identification (Cai et al. 2002). Smaller classes of protein folds from SCOP were classified with an accuracy approaching 50% correct prediction, which still is significantly more accurate than random class label assignment which yields about 4%. Compared to artificial neural network classifiers, SVM was shown to produce more accurate predictions. It should be noted that small classes of protein folds contain only a small number of samples - about 20 per class -, and still it was possible to generate a useful, though not perfect, SVM classifier.

SVM for prediction of protein-protein interaction

A major post-genomic scientific and technological pursuit is to describe the functions of proteins encoded in a genome. One strategy is to first identify protein–protein interactions in a proteome, then determine signaling pathways, and finally to statistically infer functional roles of individual proteins. Currently huge amounts of genomic data are available, and protein-protein interaction assays are being developed to meet high-throughput standards (Rudert et al. 2000; Mahlknecht et al. 2001). Data-driven methods analyzing these experimental data can play a pivotal role for inference of protein function.

SVM was used as one of the methods for predicting protein-protein interaction from the knowledge of the amino acid sequences only (Bock and Gough 2001). Features representing local physico-chemical properties of proteins were evaluated. For each amino acid sequence of a protein–protein complex, feature vectors were assembled from encoded representations of tabulated residue properties including charge, hydrophobicity, and surface tension. This feature set was motivated by the previous demonstration of sequential hydrophilicity profiles as sensitive descriptors of local interaction sites (Hopp and Woods 1983). This concept was further extended to integrate charge and surface tension, as water molecules influence atomic packing for shape complementarity, and mediate polar interactions at protein–protein recognition sites (Lo Conte et al. 1999; Böhm and Schneider 2003). The postulate was that since sequentially-proximal protein secondary structure elements are often co-located in three-dimensional conformation (Levitt and Chothia 1976), the sequential profile of these additional features (charge, surface tension) must similarly 'co-locate' upon folding. Samples were selected as pairs of proteins and these pairs were labeled as interacting or non-interacting, forming the positive and negative sets required for SVM training. The method results in approximately 80% correctly predicted interacting pairs, which means that four out of five potential protein interactions were correctly estimated by the system (Bock and Gough 2001).

Further extension of this technique aimed at predicting the energy of ligand-receptor interactions (Bock and Gough 2002). The results obtained were comparable to those of other methods, including molecular dynamics (Böhm 1998; Head et al. 1996; Wanga et al. 2002). SVM regression was used to map the feature vector of a receptor-ligand pair to the value of their interaction energy. The main conclusion of this research is that it might become possible to predict binding free energy without direct information about the three-dimensional structure of receptor and the ligand with an accuracy that is comparable to other computationally more expensive methods.

Combination of HMM and SVM for sequence analysis

A core problem in statistical biological sequence analysis is the annotation of new protein sequences with structural and functional features. To a certain degree, this can be achieved by relating new sequences to proteins for which such structural properties are already known, i.e. by detection of protein homologies. Several statistical, sequence-based tools have been developed for this purpose, including the heuristic alignment methods BLAST (Altschul et al. 1990) and FASTA (Pearson and Lipman 1988), or Hidden Markov Models (HMMs) (Krogh et al. 1994). It was shown that methods such as PSI-BLAST (Altschul et al. 1997) and HMMs, which can be used to construct a statistical model from multiple sequence alignments, perform better than simple pair-wise comparison methods, but all sequence-based methods actually miss many important remote homologues (Park et al. 1998).

A new methodology was developed by Haussler and co-workers combining a sequence-specific algorithm with the general discriminative statistical method SVM (Jaakkola et al. 1999). In their approach, HMMs were employed to extract features from protein sequences, and these features were subsequently used to train an SVM classifier. The main idea of HMM is: given a sequence of a new protein predict its probability to be generated by a certain statistical model called "HMM model". In this case $P(X \mid H)$ can be estimated, where X is a protein sequence and H is the HMM model. The naïve Bayes estimation of a log-likelihood classifier can be used to predict whether a new sequence belongs to a family of proteins:

$$\zeta(X) = \log\left(\frac{P(X \mid H_1)P(H_1)}{P(X \mid H_0)P(H_0)}\right) = \sum_{i:X_i \subset H_1} \lambda_i K(X, X_i) - \sum_{i:H_i \subset H_0} \lambda_i K(X, X_i).$$

Here $\zeta(X)$ is a log-likelihood of the ratio of probabilities of sequence X to be generated by HMM H_1 versus H_0: If $\zeta(X)$ is larger than zero then sequence X is more likely to be generated by HMM H_1 than H_0 and vice versa. Here $P(H)$ is the probability to observe sequence generated by model H and $P(X \mid H)$ is the probability to generate sequence X by HMM H. H_1 represents a model of the sequences under investigation, and H_0 is a random model. In this method $\zeta(X)$ is interpreted as a decision function and approximated in accordance to SVM theory, which means that λ_i are interpreted as Lagrange multipliers for the sample vectors X_i.

To be more specific, for predicting protein homology with SVM, $\zeta(X)$ was adopted to represent a discriminative surface like it exists in the SVM implementation. In order to do so, a new kernel function, the Fisher kernel, was introduced (Jaakkola et al. 2000). The Fisher kernel was defined as

$$K(X,X') = e^{-\frac{1}{2\sigma^2}(U_x - U_{x'})^T (U_x - U_{x'}))},$$

by introducing gradients over HMM parameters:

$$U_X = \nabla_\theta P(X \mid H_1, \theta).$$

Here $P(X \mid H_1, \theta)$ is the same as $P(X \mid H_1)$, but HMM parameters are explicitly represented by variable θ. Gradients are calculated with respect to these parameters. The magnitude of the components of the gradients specifies the extent to which each parameter contributes to generating the query sequence. A detailed description of parameter calculation in the context of HMMs along with their relation to sufficient statistics is described in more detail elsewhere (Jaakkola et al. 1999). The original work on a family of GPCR indicates that a combination of SVM and HMM leads to an improvement of the characterization of protein families from SCOP. A similar approach combining HMM and SVM was also applied for characterization of promoter regions (Pavlidis et al. 2001), and characterization of GPCR superfamilies (Karchin et al. 2002).

SVM for feature selection

Feature selection plays a pivotal role in molecular informatics. Current screening techniques provide us with a large amount of information, which allows formation of various features for analyzing biological properties. Often only a surprisingly small number of features is responsible for a certain biological effect, and techniques that will allow to find such features are very useful. An application of SVM to feature selection was described for identifying splice sites in mRNA precursors (Degroeve et al. 2002). Rouzé and co-workers selected 50 nucleotides upstream and 50 nucleotides downstream of introns conforming the GT-AG consensus for the positive training set. Each nucleotide position was then encoded by a four-digit unary number resulting in 4*(50+50) = 400 features. A feature represents presence or absence of a particular nucleotide. Almost all positives were identified (96%) but only about 30% of predicted positives were correct. Nucleotides responsible for defining splicing sites were then identified using SVM feature selection approach. Wrapper-based feature selection was used, which first performs classification with all available features and then excludes the most irrelevant feature and performs classification again. This procedure was repeated until only a small number of relevant features remained. This approach is reported to be generally superior to methods that select features prior to classification (Kohavi and John 1994).

A different method based on scaling of parameters of the SVM kernel was applied by Reifman and co-workers to identify key amino acid positions responsible for the pathology of immunoglobulin-type beta domains (Zavaljevski et al. 2002). Features were created based on physico-chemical properties of amino acids. Parameters responsible for weighting amino acid properties at predefined positions were scaled to improve classification. For example, features were enhanced at positions that are conserved over the protein family, non-conserved features were down-weighted. In addition, automatic adaptive scaling was used, which was based on the sensitivity of the classifier function to changing of the scaling factors. It was shown that appropriate scaling could significantly reduce classification error.

SVM for predicting protein sub-cellular localization

As more and more sequences of proteins become available important information about protein function can be deduced by predicting their sub-cellular localization. To date, three conceptually different approaches have been proposed: i) using targeting signals as `address labels', ii) basing the prediction on the observation that proteins from different cellular compartments tend to differ in subtle ways in their overall amino acid composition, and iii) using evolutionary relationships to infer the sub-cellular localization (Bickmore and Sutherland 2002; Emanuelsson and von Heijne 2001). Recently, SVM was trained on a set of features that is capable to capture different types of information for sub-cellular localization (Chou and Cai 2002). The method was used to predict the following locations: chloroplast, cytoplasm, cytoskeleton, endoplasmic reticulum, extracellular matrix, Golgi apparatus, lysosome, mitochondria, nucleus, peroxisome, plasma membrane, and vacuole. The decision was based on amino acid composition and functional comparison of the query protein to a set of proteins with well-defined structural and functional domain types. This procedure results in a binary vector: the *i*th component of this vector is equal to one if both amino acid sequence (based on BLASTP) and domain structure (based on HSP (Altschul et al. 1990)) of the query protein and the *i*th protein from the above mentioned set are sufficiently similar. The concept allows for incorporation of information about both the amino acid sequence and evolutionary relationships into the feature vector. The SVM classifier yielded 80% correct predictions overall, when trained on the proteins from SWISS-PROT (Bairoch and Apweiler 2000).

Application of Support Vector Machine and Neural Network Models for Classification of GPCR Ligands

Applications of SVM models in various areas of bioinformatics have shown certain advantages of SVM over other learning algorithms, and first applications of SVM in pharmaceutical research have been described (Burbidge et al. 2001; Warmuth et al. 2003; Wilton et al. 2003; Trotter and Holden 2003). Since both SVM and multi-layered artificial neural networks (ANN) are capable of solving non-linear classification tasks, there is a debate as to which system should be preferred. SVM and ANN were designed for similar purposes, so they basically have the same structure of input and output, but the algorithm for training is significantly different. SVM results should be interpreted in the same way as ANN results. The main difference between SVM and ANN is that the SVM decision surface depends only on the support vectors, i.e. data points which are close to the decision surface. In contrast, in ANN training all data points contribute to the final solution. The decision surface constructed by SVM is essentially nonlinear. Nonlinearity is introduced by mapping data points to a very high-dimensional space, where a decision surface can be found in a form of a hyperplane. The mapping function provides the correspondence between the separating hyperplane in a very high-dimensional space and a decision surface in the original data space. The exact mapping is determined by the particular kernel used for SVM training. ANN approximate a decision function in the original data space. The SVM kernel method is particularly appealing because finding a decision surface in a form of a linear hyperplane can be algorithmically treated in a straightforward manner.

For demonstration of a typical application, we compared the performance of SVM classifiers to neural network-based prediction of GPCR ligands. The classification task was to find a model separating small organic molecules which interact with GPCR ("class") from compounds which were assumed not to modulate GPCR activity ("nonclass"). Similar approaches were successfully introduced using neural networks as classifying systems some years ago and have become very popular in the field of virtual screening (Ajay et al. 1998; Sadowski and Kubinyi 1998; Schneider et al. 2001; Schneider and Böhm 2002).

For this study we used the GPCR ligands compiled in the COBRA database (Schneider and Schneider 2003). COBRA contains a large collection of bioactive molecules published in the recent scientific literature and was developed to provide sets of reference compounds for virtual screening applications and method development. 1645 GPCR ligands ("class", label +1) and 2862 "nonclass" (label -1) compounds were selected. Molecular descriptors were used to represent the molecules in a form which is suitable for learning. In this work we used MOE descriptors (Molecular Operating Environment, MOE. Chemical Computing Group Inc., Montreal) and CATS descriptors (Schneider et al. 1999). These descriptors include various physico-chemical and geometrical properties of molecules. As a result, each molecule was represented by 407 descriptors and a class label.

The neural network model. Conventional two-layered neural networks with a single output neuron were used for neural network model development (Figure 2). As a result of network training a decision function is chosen from the family of functions represented by the network architecture. This function family is defined by the complexity of the neural network, i.e. number of hidden layers, number of neurons in these layers, and topology of the network. The decision function is determined by choosing appropriate weights for the neural network. Optimal weights usually minimize an error function for the particular network architecture. The error function describes the deviation of predicted target values from observed or desired values. For our class/nonclass classification problem the target values were 1 for "class" and -1 for "nonclass", and the mean-square-error (E) served as objective function for training. Standard two-layered neural networks with a single output neuron can be represented by the following equation:

$$y = \tilde{g}\left(\sum_{j=1}^{M} w_{1j}^{(2)} \cdot g\left(\sum_{i=1}^{d} w_{ji}^{(1)} \cdot x_i + w_{j0}^{(1)} \right) + w_{11}^{(2)} \right)$$

with the error function $E = \sum_{k=1}^{n} (y(x_k) - y_k)^2$. In this study, $\tilde{g}$ was a linear function and g was a tan-sigmoid transfer function. The ANN model produced values in]-1,1[, where a positive value meant "class" and a negative value "nonclass". For weight optimization the Levenberg-Marquardt method was employed (Hagan and Menhaj 1994; Foresee and Hagan 1997), as implemented in the MATLAB package (MATLAB 2002, The mathematical laboratory. The MathWorks GmbH, D-52064 Aachen, Germany).

The SVM model. A freely available SVM-package has been published by Joachims (SVM-light package, URL: http://svmlight.joachims.org/) (Joachims 1999), which was used to build SVM models. We compared the polynomial kernel

$$K(x,y) = ((x \bullet y) * s + r)^5 ,$$

and the radial basis function (RBF) kernel

$$K(x,y) = \exp\left(\frac{|x-y|^2}{d} \right) .$$

The SVM model for GPCR ligand/nonligand classification of a pattern *x* was:

$SVM(x) = \sum_i \left(a_i \ K\left(x_i^{SV}, x\right) + b\right)$, where

i runs only over support vectors. The value of *SVM(x)* is either positive ("class") or negative ("nonclass").

Model evaluation. Classification accuracy was evaluated based on prediction accuracy, expressed by the correlation coefficient according to Matthews (Matthews 1975):

$$cc = \frac{NP - OU}{\sqrt{(N+O)(N+U)(P+O)(P+U)}} ,$$

where P, N, O, U are the number of true positive, true negative, false positive and false negative predictions respectively. GPCR ligands were considered as "positive set", the non-GPCR ligands formed the "negative set". The values of *cc* can range from –1 to 1. Perfect prediction gives a correlation coefficient of 1. Different training and test sets were selected for 10-fold cross-validation (random 80 + 20 splits) and average values of <*cc*> were calculated.

Results and discussion. SVM and ANN classification models were developed using different numbers of training samples. Figure 3 shows the dependence of the classification accuracy as a function of the size of the training subset. It is evident that accuracy is gained by larger training sets, irrespective of the type of learning system applied. The SVM classifier yielded higher average classification accuracy (<*cc*> = 0.78) than the ANN models tested (<*cc*> = 0.67). Although ANN classification performance increased with the number of hidden neurons it did not reach the accuracy of the SVM model. In a previous comparison of SVM to several machine learning methods by Holden and co-workers it was demonstrated that an SVM classifier outperformed other standard methods, but a specially designed and structurally optimized neural network was again superior to the SVM model in a benchmark test (Burbidge et al. 2001). In the present study an SVM was optimized with regard to its model parameters and applied to the identical classification task (GPCR ligands vs. nonligands) as neural networks which were not specifically optimized. With appropriate structural optimization it might be possible to obtain an ANN architecture showing comparable performance. It is always required to properly understand the classification technique applied and optimize its tunable features in order to draw conclusions about strengths and weaknesses of a method in comparison to other approaches.

In a second study, we compared different kernels for SVM training. Figure 4 shows the dependence of the Matthews correlation coefficient on the type of kernel used for training. We compared the polynomial kernel with the RBF kernel. Both kernels performed similarly, with slightly higher classification accuracy yielded by the RBF kernel. Again, we wish to stress that this result does not justify conclusions regarding a general superiority of the RBF kernel to be drawn, since the accuracy obtained with the polynomial kernel was still within the standard deviation margin. The mapping to a high-dimensional space where the optimal hyperplane is determined is only defined by the SVM kernel function. Provided that the dimensionality of this very high-dimensional space is sufficiently large, classification accuracy should not depend on the particular kernel (Vapnik 1995). In practice, almost every kernel has parameters that should be tuned for optimal classification accuracy. Provided that these parameters are optimally tuned SVM classification accuracy only slightly depends the particular form of a kernel. This was demonstrated by our comparison of the polynomial and RBF kernel in the GPCR example (Figure 4).

In the light of medicinal chemistry it is remarkable that it was possible to obtain a prediction system for GPCR ligands with comparably high prediction accuracy (approximately 90% correct). Previous attempts were grounded on privileged structures (Klabunde and Hessler 2002), receptor-ligand docking (Vaidehi et al., 2002), phylogenetic analysis and ligand clustering (Vassilatis et al. 2003), neural network systems (Manallack et al. 2002), a self-organizing map (Schneider et al. 2001; Schneider and Nettekoven 2003), or property-based library design (Balakin et al. 2002). Our new tool can now be used for very fast early-phase virtual screening campaigns to collect promising candidates for further evaluation and testing. It represents a useful virtual filtering technique for constraining the size of GPCR-targeted libraries that will help speed up the identification of novel GPCR-ligands.

Conclusions

It was demonstrated by a number of independent research teams and the present study that irrespective of the particular application a properly designed SVM has the capability to compete with and sometimes outperform other data-driven methods including supervised and unsupervised learning. SVM performs particularly well when large numbers of features are available, as typified by gene chip analysis experiments. A particular advantage of SVM is "sparseness of the solution". This means that an SVM classifier depends only on the support vectors and the classifier function is not influenced by the whole data set, as it is the case for most neural network systems. It should be kept in mind that irrespective of the appeal of the technique, SVM should be considered as complementary to other methods. And just like for ANN training, there are model parameters to be tuned for optimal SVM performance. We expect that in the future prediction will be increasingly performed by a plethora of classifiers obtained by different methods, which are combined by a jury decision method. SVM are likely to play a role in this game.

Acknowledgement

Uli Fechner is thanked for helpful comments on the manuscript. This work was supported by the Beilstein-Institut zur Förderung der Chemischen Wissenschaften.

References

Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. 1990. Basic local alignment search tool. *J Mol Biol*, 215:403-410.

Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ. 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res,* 25:3389-3402.

Ajay, Walters WP, Murcko MA. 1998. Can we learn to distinguish between "drug-like" and "nondrug-like" molecules? *J Med Chem,* 41:3314-3324.

Bairoch A, Apweiler R. 2000. The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Res,* 28:45-48.

Bickmore WA, Sutherland HG. 2002. Addressing protein localization within the nucleus. *Embo J,* 21:1248-1254.

Bishop CM. 1995. Neural Networks for Pattern Recognition. Oxford: Oxford University Press.

Bock JR, Gough DA. 2002. A new method to estimate ligand-receptor energetics. *Mol Cell Proteomics,* 1:904-910.

Bock JR, Gough DA. 2001. Predicting protein--protein interactions from primary structure. *Bioinformatics,* 17:455-460.

Böhm HJ. 1998. Prediction of binding constants of protein ligands: a fast method for the prioritization of hits obtained from de novo design or 3D database search programs. *J Comput Aided Mol Des,* 12:309-323.

Böhm HJ, Schneider G, eds. 2003. Protein-ligand interaction. From molecular recognition to drug design. Weinheim, New York: Wiley-VCH.

Brown MP, Grundy WN, Lin D, Cristianini N, Sugnet CW, Furey TS, Ares M, Jr., Haussler D. 2000. Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc Natl Acad Sci USA*, 97:262-267.

Burbidge R, Trotter M, Buxton B, Holden S. 2001. Drug design by machine learning: support vector machines for pharmaceutical data analysis. *Comput Chem*, 26:5-14.

Burges CJC. 1998. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2:121-167.

Cai YD, Liu XJ, Xu XB, Chou KC. 2002. Prediction of protein structural classes by support vector machines. *Comput Chem*, 26:293-296.

Cai YD, Liu XJ, Xu XB, Chou KC. 2002. Support vector machines for the classification and prediction of beta-turn types. *J Pept Sci*, 8:297-301.

Chou KC, Cai YD. 2002. Using functional domain composition and support vector machines for prediction of protein subcellular location. *J Biol Chem*, 277:45765-45769.

Coleman TF, Li Y. 1996. A Reflective Newton Method for Minimizing a Quadratic Function Subject to Bounds on some of the Variables. *SIAM Journal on Optimization*, 6:1040-1058.

Cortes C, Vapnik V. 1995. Support-Vector Networks. *Machine Learning*, 20:273-297.

Cristianini N, Shawe-Taylor J. 2000. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge: Cambridge University Press.

Degroeve S, De Baets B, Van De Peer Y, Rouze P. 2002. Feature subset selection for splice site prediction. *Bioinformatics*, 18 Suppl 2:S75-S83.

Ding CH, Dubchak I. 2001. Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics*, 17:349-358.

Emanuelsson O, von Heijne G. 2001. Prediction of organellar targeting signals. *Biochim Biophys Acta*, 1541:114-119.

Foresee FD, Hagan MT. 1997. Gauss-Newton approximation to Bayesian regularization. *Proceedings of the 1997 International Joint Conference on Neural Networks*, pp.1930-1935.

Furey TS, Cristianini N, Duffy N, Bednarski DW, Schummer M, Haussler D. 2000. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16:906-914.

Guyon I, Weston J, Barnhill S, Vapnik V. 2002. Gene Selection for Cancer Classification using Support Vector Machines. *Machine Learning*, 46:389-422.

Hagan MT, Menhaj M. 1994. Training feedforward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks*, 5:989-993.

Head RD, Smythe ML, Oprea TI, Waller CL, Green SM, Marshall GR. 1996. VALIDATE: A New Method for the Receptor-Based Prediction of Binding Affinities of Novel Ligands. *J Am Chem Soc,* 118:3959-3969.

Hopp TP, Woods KR. 1983. A computer program for predicting protein antigenic determinants. *Mol Immunol*, 20:483-489.

Jaakkola T, Diekhans M, Haussler D. 2000. A discriminative framework for detecting remote protein homologies. *J Comput Biol*, 7:95-114.

Jaakkola T, Diekhans M, Haussler D. 1999. Using the Fisher kernel method to detect remote protein homologies. *Proc Int Conf Intell Syst Mol Biol*:149-158.

Joachims T. 1999. Making large-Scale SVM Learning Practical. In B. Schölkopf, C. Burges and A. Smola, eds. Advances in Kernel Methods - Support Vector Learning. Cambridge: MIT-Press. p 41-56.

Karchin R, Karplus K, Haussler D. 2002. Classifying G-protein coupled receptors with support vector machines. *Bioinformatics*, 18:147-159.

Klabunde T, Hessler G (2002) Drug design strategies for targeting G-protein-coupled receptors. *Chembiochem*, 3:928-944.

Kohavi R, John GH. 1994. Wrappers for Feature Subset Selection. *Artificial Intelligence*, 97:273-324.

Krogh A, Brown M, Mian IS, Sjolander K, Haussler D. 1994. Hidden Markov models in computational biology. Applications to protein modeling. *J Mol Biol*, 235:1501-1531.

Levitt M, Chothia C. 1976. Structural patterns in globular proteins. *Nature*, 261:552-558.

Lo Conte L, Ailey B, Hubbard TJ, Brenner SE, Murzin AG, Chothia C. 2000. SCOP: A structural classification of proteins database. *Nucleic Acids Res*, 28:257-259.

Lo Conte L, Chothia C, Janin J. 1999. The atomic structure of protein-protein recognition sites. *J Mol Biol*, 285:2177-2198.

Mahlknecht U, Ottmann OG, Hoelzer D. 2001. Far-Western based protein-protein interaction screening of high-density protein filter arrays. *J Biotechnol*, 88:89-94.

Manallack DT, Pitt WR, Gancia E, Montana JG, Livingstone DJ, Ford MG, Whitley DC (2002) Selecting screening candidates for kinase and G protein-coupled receptor targets using neural networks. *J Chem Inf Comput Sci*, 42:1256-1262.

Matthews BW. 1975. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim Biophys Acta*, 405:442-451.

Park J, Karplus K, Barrett C, Hughey R, Haussler D, Hubbard T, Chothia C. 1998. Sequence comparisons using multiple sequences detect three times as many remote homologues as pairwise methods. *J Mol Biol*, 284:1201-1210.

Pavlidis P, Furey TS, Liberto M, Haussler D, Grundy WN. 2001. Promoter region-based classification of genes. *Pac Symp Biocomput*:151-163.

Pavlidis P, Weston J, Cai J, Noble WS. 2002. Learning gene functional classifications from multiple data types. *J Comput Biol*, 9:401-411.

Pearson WR, Lipman DJ. 1988. Improved tools for biological sequence comparison. *Proc Natl Acad Sci USA*, 85:2444-2448.

Rudert F, Ge L, Ilag LL. 2000. Functional genomics with protein-protein interactions. *Biotechnol Annu Rev*, 5:45-86.

Sadowski J, Kubinyi H. 1998. A scoring scheme for discriminating between drugs and nondrugs. *J Med Chem*, 41:3325-3329.

Schneider G, Böhm HJ. 2002. Virtual screening and fast automated docking methods. *Drug Discov Today*, 7:64-70.

Schneider G, Neidhart W, Giller T, Schmid G. 1999. "Scaffold-Hopping" by topological pharmacophore search: a contribution to virtual screening. *Angew Chemie Int Ed*, 38:2894-2896

Schneider G, Neidhart W, Adam G. 2001. Integrating virtual screening to the quest for novel membrane protein ligands. *Curr Med Chem CNSA*, 1:99-112.

Schneider G, Neidhart W, Giller T, and Schmid G. 1999. "Scaffold-Hopping" by Topological Pharmacophore Search: A Contribution to Virtual Screening. *Angew Chem Int Ed Engl* 38:2894-2896.

Schneider G, Nettekoven M (2003) Ligand-based combinatorial design of selective purinergic receptor (A(2A)) antagonists using self-organizing maps. *J Comb Chem,* 5:233-237.

Schneider P, Schneider G. 2003. Collection of bioactive reference compounds for focused library design. *QSAR Comb Sci*, in press.

Trotter MWB, Holden SB (2003) Supprt vector machines for ADME property classification. *QSAR Comb Sci*, 5:533-548.

Vaidehi N, Floriano WB, Trabanino R, Hall SE, Freddolino P, Choi EJ, Zamanakos G, Goddard WA 3rd (2002) Prediction of structure and function of G protein-coupled receptors. *Proc Natl Acad Sci USA*, 99:12622-12627.

Vapnik V. 1995. The Nature of Statistical Learning Theory. Berlin: Springer.

Vassilatis DK, Hohmann JG, Zeng H, Li F, Ranchalis JE, Mortrud MT, Brown A, Rodriguez SS, Weller JR, Wright AC, Bergmann JE, Gaitanaris GA (2003) The G protein-coupled receptor repertoires of human and mouse. *Proc Natl Acad Sci USA*, 100:4903-4908.

Wanga R, Laib L, Wanga S. 2002. Further development and validation of empirical scoring functions for structure-based binding affinity prediction. *J Comput Aided Mol Des*, 16:11–26.

Warmuth MK, Liao J, Ratsch G, Mathieson M, Putta S, Lemmen C. 2003. Active learning with Support Vector Machines in the drug discovery process. *J Chem Inf Comput Sci*, 43:667-673.

Wilton D, Willett P, Lawson K, Mullier G. 2003. Comparison of ranking methods for virtual screening in lead-discovery programs. *J Chem Inf Comput Sci*, 43:469-474.

Zavaljevski N, Stevens FJ, Reifman J. 2002. Support vector machines with selective kernel scaling for protein classification and identification of key amino acid positions. *Bioinformatics*, 18:689-696.

Legend to the figures

Figure 1.
SVM classification. This figure illustrates values of an SVM model obtained during SVM training. The task was to separate two classes of objects indicated by squares and circles. Squares represent nonclass samples ("negative examples") and circles are class members ("positive examples"). $D(\mathbf{x}) = (\mathbf{w} \bullet \mathbf{x}) + w_0$ is the decision function defining class membership according to the SVM classifier. The SVM classifier is represented by the separating line ($D(\mathbf{x}) = 0$). The margin is indicated by dotted lines. Support vectors are indicated by filled objects (x_2, x_2, x_3, x_4). ξ_i are slack variables for support vectors that are not lying on the margin border. y_i are label-variables equal to 1 for positive examples (class membership) and -1 for negative examples (nonclass membership).

$\xi_1 = 1 - D(x_1)$

$y * D(x_2) = 1 - \xi_2 > 0$

$\xi 3 = 1 + D(x_3)$

$y * D(x_4) = 1 - \xi_4 < 0$

Figure 2.
Architecture of an artificial feed-forward neural network. Formal neurons are drawn as circles, weights are represented by lines connecting the neuron layers. Fan-out neurons are drawn in white, sigmoidal units in black, and linear units in gray.

Figure 3.
Matthews correlation coefficients for prediction of GPCRs inhibitors. SVM-based prediction was compared to neuronal network performance. Matthews correlation coefficients for the test samples are plotted versus number of samples used for training. Standard deviations are shown as dotted lines.

Figure 4.
Matthews correlation coefficients for comparison of different kernels used in SVM training. Classification performance of SVM classification performance depends only slightly depends on the nature of kernel function, provided that parameters of the kernel are optimized during training. Here, GPCR inhibitors were predicted by SVM with an RBF and a polynomial kernel. Standard deviations are shown as dotted lines.

Figure 1

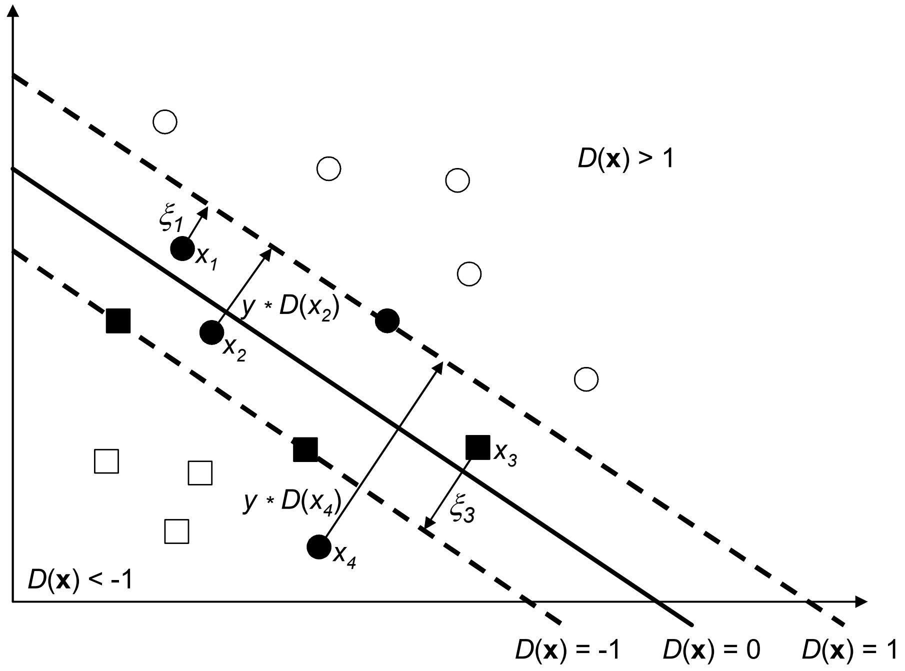

$D(\mathbf{x}) > 1$
ξ_1
x_1
$y * D(x_2)$
x_2
x_3
ξ_3
$y * D(x_4)$
x_4
$D(\mathbf{x}) < -1$
$D(\mathbf{x}) = -1$
$D(\mathbf{x}) = 0$
$D(\mathbf{x}) = 1$

Figure 2

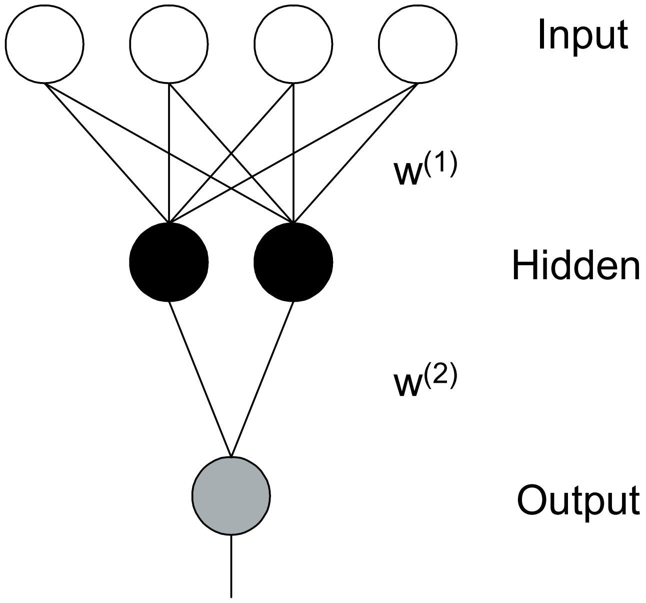


Figure 3

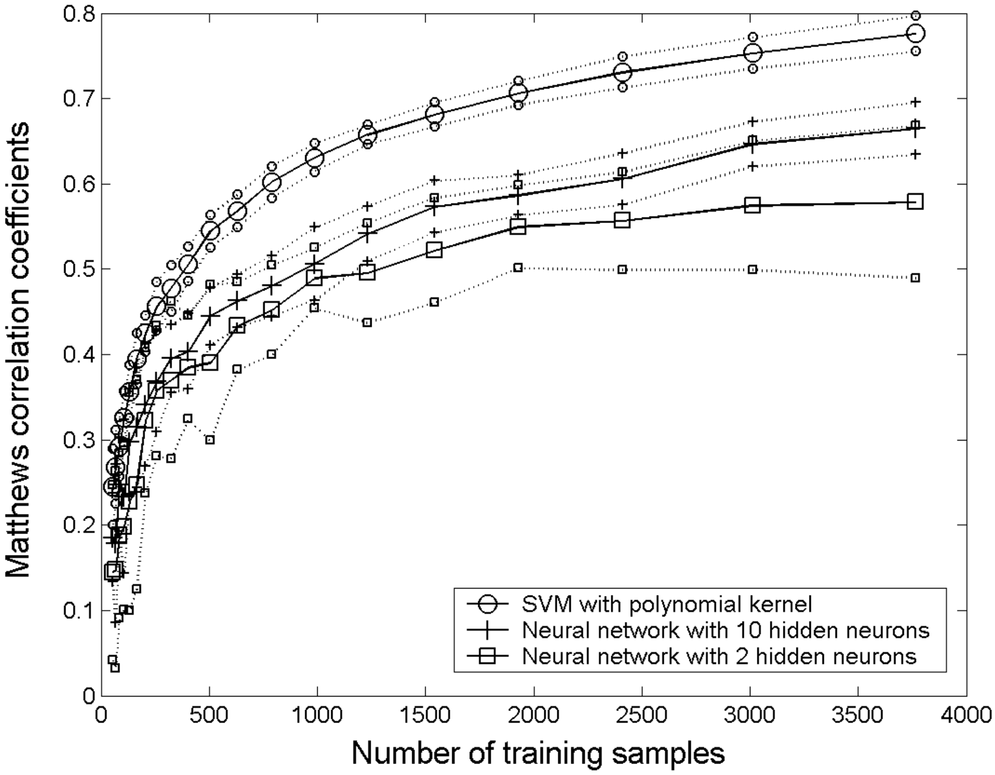
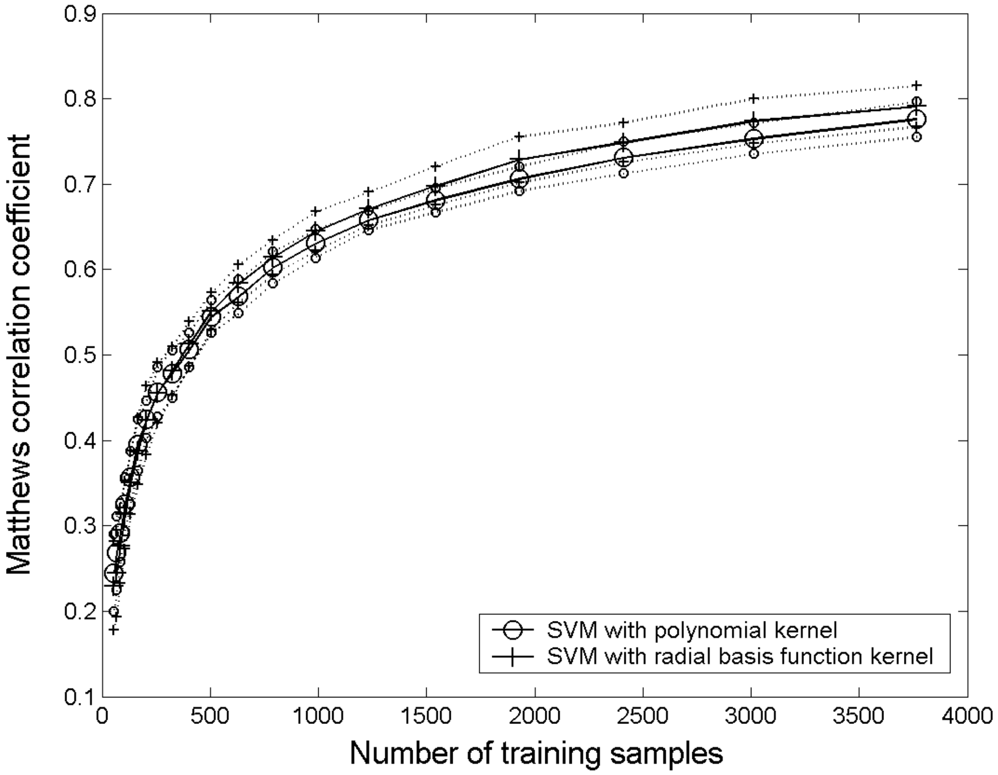

Matthews correlation coefficients
Number of training samples
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0
500
1000
1500
2000
2500
3000
3500
4000
SVM with polynomial kernel
Neural network with 10 hidden neurons
Neural network with 2 hidden neurons

Figure 4


Matthews correlation coefficient
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0
500
1000
1500
2000
2500
3000
3500
4000
Number of training samples
SVM with polynomial kernel
SVM with radial basis function kernel

7.2 Comparison of support vector machine and artificial neural network systems for drug/nondrug classification

Byvatov E., Fechner U., Sadowski J., Schneider G.
J. Chem. Inf. Comput. Sci. 2003; 43(6):1882-9

Comparison of Support Vector Machine and Artificial Neural Network Systems for Drug/Nondrug Classification

Evgeny Byvatov,[†] Uli Fechner,[†] Jens Sadowski,[‡] and Gisbert Schneider*,[†]

Institut für Organische Chemie und Chemische Biologie, Johann Wolfgang Goethe-Universität,
Marie-Curie-Strasse 11, D-60439 Frankfurt, Germany, and AstraZeneca R&D Mölndal, SC 264,
S-431 83 Mölndal, Sweden

Received June 13, 2003

Support vector machine (SVM) and artificial neural network (ANN) systems were applied to a drug/nondrug classification problem as an example of binary decision problems in early-phase virtual compound filtering and screening. The results indicate that solutions obtained by SVM training seem to be more robust with a smaller standard error compared to ANN training. Generally, the SVM classifier yielded slightly higher prediction accuracy than ANN, irrespective of the type of descriptors used for molecule encoding, the size of the training data sets, and the algorithm employed for neural network training. The performance was compared using various different descriptor sets and descriptor combinations based on the 120 standard Ghose-Crippen fragment descriptors, a wide range of 180 different properties and physicochemical descriptors from the Molecular Operating Environment (MOE) package, and 225 topological pharmacophore (CATS) descriptors. For the complete set of 525 descriptors cross-validated classification by SVM yielded 82% correct predictions (Matthews $cc = 0.63$), whereas ANN reached 80% correct predictions (Matthews $cc = 0.58$). Although SVM outperformed the ANN classifiers with regard to overall prediction accuracy, both methods were shown to complement each other, as the sets of true positives, false positives (overprediction), true negatives, and false negatives (underprediction) produced by the two classifiers were not identical. The theory of SVM and ANN training is briefly reviewed.

INTRODUCTION

Early-phase virtual screening and compound library design often employs filtering routines which are based on binary classifiers and are meant to eliminate potentially unwanted molecules from a compound library.[1,2] Currently two classifier systems are most often used in these applications: PLS-based classifiers[3,4] and various types of artificial neural networks (ANN).[5−9] Typically, these systems yield an average overall accuracy of 80% correct predictions for binary decision tasks following the "likeness concept" in virtual screening.[2,10] The support vector machine (SVM) approach was first introduced by Vapnik as a potential alternative to conventional artificial neural networks.[11,12] Its popularity has grown ever since in various areas of research, and first applications in molecular informatics and pharmaceutical research have been described.[13−15] Although SVM can be applied to multiclass separation problems, its original implementation solves binary class/nonclass separation problems. Here we describe application of SVM to the drug/nondrug classification problem, which employs a class/nonclass implementation of SVM. Both SVM and ANN algorithms can be formulated in terms of learning machines. The standard scenario for classifier development consists of two stages: training and testing. During first stage the learning machine is presented with labeled samples, which are basically n-dimensional vectors with a class membership label attached. The learning machine generates a classifier for prediction of the class label of the input coordinates. During the second stage, the generalization ability of the model is tested.

Currently various sets of molecular descriptors are available.[16] For application to drug/nondrug classification of compounds, the molecules are typically represented by n-dimensional vectors.[6,7] In this work, we focused on the fragment-based Ghose-Crippen (GC) descriptors[17−19] which were used in the original work of Sadowski and Kubinyi for drug/nondrug classification,[7] descriptors provided by the MOE software package (Molecular Operating Environment. Chemical Computing Group Inc., Montreal, Canada), and CATS topological pharmacophores.[20] Having defined this molecular representation, the task of the present study was to compare the classification ability of standard SVM and feed-forward ANN on the drug/nondrug data. A www-based interface for calculating the drug-likeness score of a molecule using our SVM solution based on the CATS descriptor was developed and can be found at URL: http://gecco.org.chemie.uni-frankfurt.de/gecco.html.

DATA AND METHODS

Data Sets. For SVM and ANN training we used the sets of "drug" and "nondrug" molecules prepared by Kubinyi and Sadowski.[7] From the original data set 9208 molecules could be processed by our descriptor generation software. The final working set contained 4998 drugs and 4210 nondrug molecules. Three sets of descriptors were calculated: counts of the standard 120 Ghose Crippen descriptors,[17−19] 180

* Corresponding author phone: +49-69 79829821; fax: +49-69 7982-9826; e-mail: gisbert.schneider@modlab.de.
[†] Johann Wolfgang Goethe-Universität.
[‡] AstraZeneca R&D Mölndal.

10.1021/ci0341161 CCC: $25.00 © 2003 American Chemical Society
Published on Web 09/27/2003

descriptors from MOE (Molecular Operating Environment. Chemical Computing Group Inc., Montreal, Canada), and 225 topological pharmacophore (CATS) descriptors.[20] MOE descriptors include various 2D and 3D descriptors such as volume and shape desciptors, atom and bonds counts, Kier–Hall connectivity and kappa shape indices, adjacency and distance matrix descriptors, pharmacophore feature descriptors, partial charges, potential energy descriptors, and conformation-dependent charge descriptors. Before calculating MOE descriptors, single 3D conformers were generated by CORINA.[21] 225 CATS descriptors were calculated using our own software taking into consideration pairs of atom types separated by up to 15 bonds (URL: http://gecco.org.chemie.uni-frankfurt.de/gecco.html).[20] All 225 descriptor columns were individually autoscaled. An alternative would have been block-scaling where each descriptor class is autoscaled as a whole, which was not applied here.

Support Vector Machine. SVM classifiers are generated by a two-step procedure: First, the sample data vectors are mapped ("projected") to a very high-dimensional space. The dimension of this space is significantly larger than dimension of the original data space. Then, the algorithm finds a hyperplane in this space with the largest margin separating classes of data. It was shown that classification accuracy usually depends only weakly on the specific projection, provided that the target space is sufficiently high dimensional.[11] Sometimes it is not possible to find the separating hyperplane even in a very high-dimensional space. In this case a tradeoff is introduced between the size of the separating margin and penalties for every vector which is within the margin.[11] The basic theory of SVM will be briefly reviewed in the following.

The separating hyperplane is defined as

$$D(\mathbf{x}) = (\mathbf{w} \cdot \mathbf{x}) + w_0$$

Here $\boldsymbol{x}$ is a samples vector mapped to a high dimensional space, and $\boldsymbol{w}$ and w_0 are parameters of the hyperplane that SVM will estimate. Then the margin can be expressed as a minimal τ for which holds

$$\frac{y_k D(\mathbf{x_k})}{||\mathrm{w}||} \geq \tau$$

Without loss of generality we can apply a constraint $\tau||w|| = 1$ to $\boldsymbol{w}$. In this case maximizing τ is equivalent to minimizing $||w||$ and SVM training is becoming the problem of finding the minimum of a function with the following constraints:

$$\textit{minimize} \quad \eta(w) = \frac{1}{2}(w \cdot w)$$

$$\textit{subject to constraints} \quad y_i[(w \cdot x_i) + w_0] \geq 1$$

This problem is solved by introduction of Lagrange multipliers and minimization of the function

$$Q(\mathbf{w}, w_0, \alpha) = \frac{1}{2}(\mathbf{w} \cdot \mathbf{w}) - \sum_{i=1}^{n} \alpha_i \{y_i[(\mathbf{w} \cdot \mathbf{x_i}) + w_0] - 1\}$$

Here α_i are Lagrange multipliers. Differentiating over $\boldsymbol{w}$ and w_i and substituting we obtain

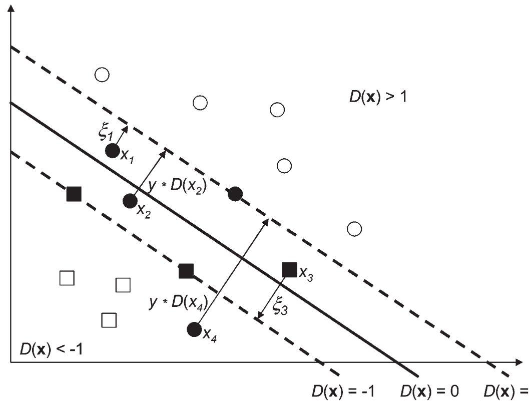


Figure 1. Principle of SVM classification. The task was to separate two classes of objects indicated by squares and circles. Squares represent nonclass samples ("negative examples", e.g. nondrugs) and circles are class members ("positive examples", e.g. drugs). $D(\mathbf{x})$ is the decision function defining class membership according to the SVM classifier which is represented by the separating line ($D(\mathbf{x}) = 0$). The margin is indicated by dotted lines. Support vectors are indicated by filled objects (x_2, x_2, x_3, x_4). ξ_i are slack variables for support vectors that are not lying on the margin border. y_i are label-variables equal to 1 for positive examples (class membership) and -1 for negative examples (nonclass membership). See text for details.

$$\max \quad Q(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$

$$\textit{subject to constraints} \quad \sum_{i=1}^{n} y_i \alpha_i = 0; \ \alpha_i \geq 0, i = 1,...,n$$

When perfect separation is not possible slack variables are introduced for sample vectors which are within the margin, and the optimization problem can be reformulated:

$$\textit{minimize } \eta(w) = \frac{1}{2}(w \cdot w) + C\sum_i \xi_i$$

$$\textit{subject to constraints } y_i[(w \cdot x_i) + w_0] \geq 1 - \xi_\mathrm{i}$$

Here ξ_i are slack variables. These variables are not equal to zero only for those vectors which are within the margin. Introducing Lagrange multipliers again we finally obtain

$$\max \quad Q(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$

$$\textit{subject to constraints} \quad \sum_{i=1}^{n} y_i \alpha_i = 0, \ C \geq \alpha_i \geq 0, i = 1,...,n$$

This is a quadratic programming (QP) problem for which several efficient standard methods are known.[22] Due to the very high dimensionality of the QP problem, which typically arises during SVM training, an extension of the algorithm for solving QP is used in SVM applications.[23]

A geometrical illustration of the meaning of slack variables and Lagrange multipliers is given in Figure 1. Points classified by SVM can be divided into two groups, support vectors and nonsupport vectors. Nonsupport vectors are classified correctly by the hyperplane and are located outside the separating margin. Slack variables and Lagrange multipliers for them are equal to zero. Parameters of the hyperplane do not depend on them, and even if their position is changed the separating hyperplane and margin will remain unchanged, provided that these points will stay outside the margin. Other points are support vectors, and they are the points which determine the exact position of the hyperplane. For all support vectors the absolute values of the slack variables are equal to the distances from these points to the edge of the separating margin. These distances are defined in the units of half of the width of the separating margin. For correctly classified points within the separating margin, slack variable values are between zero and one. For misclassified points within the margin the values of the slack variables are between one and two. For other misclassified points they are greater than two.

For points that are lying on the edge of margin, Lagrange multipliers are between zero and *C*, and slack variables for these points are still equal to zero. For all other points, for which the values of slack variables are larger than zero, Lagrange multipliers assume the value of *C*.

Explicit mapping to a very high-dimensional space is not required if calculation of the scalar product in this high dimensional space of every two vectors is feasible. This scalar product can be defined by introducing a kernel function$(\mathbf{x}\cdot\mathbf{x}') = K(\mathbf{x},\mathbf{x}')$,[24] where $\mathbf{x}$ and $\mathbf{x}'$ are vectors in a low-dimensional space for which a kernel function that corresponds to a scalar product in a high dimensional space is defined. Various kernels may be applied.[25] In our case, we used a kernel function of a fifth-order polynomial:

$$K(\mathbf{x},\mathbf{x}') = ((\mathbf{x}\cdot\mathbf{x}')s + r)^5$$

This kernel corresponds to the decision function

$$f(x) = sign(\sum_i \alpha_i K(x_i^{sv}, x) + b)$$

where α_i are Lagrange multipliers determined during training of SVM. The sum is only over support vectors x^{sv}. Lagrange multipliers for all other points are equal to zero. Parameter *b* determines the shift of the hyperplane, and it is also found during SVM training. Simultaneous scaling of *s, r*, and *b* parameters does not change the decision function. Thus, we can simplify the kernel by setting *r* equal to one:

$$K(x,x') = ((x\cdot x')s + 1)^5$$

In this case only the kernel parameter *s* and error tradeoff *C* must be tuned. Parameter *C* is not present explicitly in this equation; it is set up as a penalty for the misclassification error before the training of SVM is performed. For tuning parameters *s* and *C*, four-times cross-validation of training data was applied, and values for *s* and *C* that maximize accuracy were then chosen. Accuracy maximization was performed by heuristics based gradient descent.[26] Basically, the following procedure was applied. The data set was divided into two parts, training and validation set. The validation subset was put aside and used only for estimation of the performance of the trained classifier. Training data were divided into four nonoverlapping subsets. The SVM parameters to be determined were set to reasonable initial values. Then, the SVM was trained on the training data excluding one of the four subsets, and the performance of the obtained SVM classifier was estimated with the excluded subset. This procedure was repeated for each subset, and an average performance of the SVM classifier was obtained.

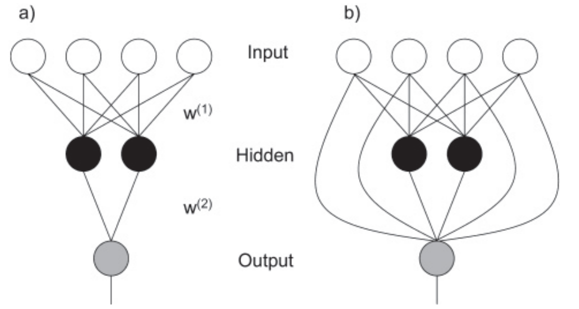


Figure 2. Architecture of artificial neural networks. Formal neurons are drawn as circles, weights are represented by lines connecting the neuron layers. Fan-out neurons are drawn in white, sigmoidal units in black, and linear units in gray. (a) conventional three-layered feed-forward system ("architecture I"); (b) network architecture used by Ajay and co-workers for drug-likeness prediction ("architecture II").[6]

For SVM training we used freely available SVM software (SVM-*Light* package; URL: http://svmlight.joachims.org/).[26,27] A Linux-based LSF (Load Sharing Facility; Platform Computing GmbH, D-40878 Ratingen, Germany) cluster was used for determination of the cross-validation error to reduce calculation time. All calculations were performed using the MATLAB package (MATLAB 2002, The mathematical laboratory. The MathWorks GmbH, D-52064 Aachen, Germany).

ARTIFICIAL NEURAL NETWORK

Conventional two-layered neural networks with a single output neuron were used for ANN model development (Figure 2a).[26] As a result of network training a decision function is chosen from the family of functions represented by the network architecture. This function family is defined by the complexity of the neural network: number of hidden layers, number of neurons in these layers, and topology of the network. The decision function is determined by choosing appropriate weights for the neural network. Optimal weights usually minimize an error function for the particular network architecture. The error function describes the deviation of predicted target values from observed or desired values. For our class/nonclass classification problem the target values were 1 for class (drugs) and −1 for nonclass (nondrugs). Standard two-layered neural network with a single output neuron can be represented by the following equation

$$y = \tilde{g}(\sum_{j=1}^{M} w_{1j}^{(2)}\cdot g(\sum_{i=1}^{d} w_{ji}^{(1)}\cdot x_i + w_{j0}^{(1)}) + w_{11}^{(2)})$$

with the error function $E = \sum_{k=1}^{n}(y(x_k) - y_k)^2$. In this work, $\tilde{g}$ is a linear function and g is a tan-sigmoid transfer function.

A second type network architecture containing additional connections from the input layer to the output layer was trained to reimplement the original drug/nondrug ANN developed by Ajay and co-workers (Figure 2b).[6]

Training of neural network is typically performed on variations of gradient descent based algorithms,[26] trying to

Table 1. Cross-Validated Results of Machine Learning[a]

	% correct		Matthews *cc*	
descriptors	ANN	SVM	ANN	SVM
GC	79.25 ± 0.66	80.01 ± 0.087	0.567 ± 0.012	0.592 ± 0.002
MOE	77.89 ± 0.74	80.19 ± 0.74	0.537 ± 0.013	0.593 ± 0.016
CATS_225	72.13 ± 0.88	73.90 ± 0.51	0.432 ± 0.013	0.485 ± 0.011
all (GC+MOE+CATS)	80.05 ± 1.02	82.24 ± 0.66	0.579 ± 0.018	0.633 ± 0.010

[a] Average values and standard deviations are given. The Levenberg−Marquardt training method was used for ANN training.

minimize an error function. To avoid overfitting cross-validation can be used for finding an earlier point of training.[28] In this work the neural network toolbox from MATLAB was used. Data were preprocessed identically to SVM based learning. We applied the following training algorithms to ANN optimization in their default versions provided by MATLAB: gradient descent with variable learning rate,[29,30] conjugated gradient descent,[30,31] scaled conjugated gradient descent,[32] quasi-Newton algorithm,[33] Levenberg−Marquardt (LM),[34,35] and automated regularization.[36] For each optimization ten-times cross-validation was performed (80+20 splits into training and test data), where the ANN weights and biases were optimized using the training data, and prediction accuracy was measured using test data to determine the number of training epochs, i.e., the endpoint of the training process. This was performed to reduce the risk of overfitting. It should be noted that the validation data were left untouched.

MODEL VALIDATION

The SVM model for drug/nondrug classification of a pattern *x* was

$$SVM(x) = \sum_i (a_i K(x_i^{SV}, x) + b)$$

Here, *i* runs only over support vectors (SV). The value of *SVM(x)* is either positive ("drug") or negative ("nondrug").

The ANN model for drug/nondrug classification produced values in]-1,1[, where a positive value meant "drug" and a negative value "nondrug".

Classification accuracy was evaluated based on prediction accuracy, i.e., percent of test compounds correctly classified, and the correlation coefficient according to Matthews:[37]

$$cc = \frac{NP - OU}{\sqrt{(N+O)(N+U)(P+O)(P+U)}}$$

where *P*, *N*, *O*, and *U* are the number of true positive, true negative, false positive, and false negative predictions, respectively. Drugs were considered as "positive set", the nondrug molecules formed the "negative set". The values of *cc* can range from −1 to 1. Perfect prediction gives a correlation coefficient of 1.

SVM and ANN models were developed using various sizes of training data to measure the influence of the size of the training set on the quality of the classification model. The number of training samples was iteratively diminished: Starting with an 80+20 random split of all available samples into training and validation subsets, at each of the following iterations we diminished the size of the training set to only 80% of the number of samples of the previous iteration. This allowed us to obtain better sampling for small training sets. 10-times cross-validation was performed, and average values of prediction accuracy and $\langle cc \rangle$ were calculated.

RESULTS AND DISCUSSION

The main aim of this study was to compare SVM and ANN classifiers in their ability to distinguish between sets of "drugs" and "nondrugs". We trained different neural network topologies, and performance of the best network was compared to the SVM classifier.

Two types of ANN architecture were considered: standard feed-forward networks with one hidden layer ("architecture I") and a feed-forward network with one hidden layer with additional direct connections from input neurons to the output ("architecture II") (Figure 2). The first type of ANN was used by Sadowski and Kubinyi in their original work on drug-likeness prediction;[7] the second architecture was employed by Ajay and co-workers serving the same purpose.[6] Using these networks and the GC descriptors in combination with the Levenberg−Marquardt training method, classification accuracy was identical to the original results (on average 80% correct) despite the use of a different training technique and different training data (Table 1). This observation substantiates the original findings. Both network types performed identically considering the error margin (approximately 80% correct classification). We observed that for some of the training algorithms a slightly lower standard deviation of the prediction accuracy was observed for architecture I (data not shown). Since the additional connections in network architecture II did not contribute to a greater accuracy of the model, we used only the standard feed-forward network with one hidden layer containing two neurons (architecture I) for further analysis.

For each training method and combination of input variables (descriptors) networks with different numbers of hidden neurons (2−10 neurons) were trained. Overall, we did not observe an overall best training algorithm. The Levenberg−Marquardt method was used for the development of the final ANN model. Also, we did not observe an improved classification result when the number of hidden neurons was larger than two (data not shown). ANN architecture I with two hidden neurons yielded the overall best cross-validated prediction result for all descriptors (GC+MOE+CATS), 80% correct predictions ($\langle cc \rangle = 0.58$). The rank order of descriptor sets with regard to the overall classification accuracy yielded was as follows: All > GC > MOE > CATS (Table 1). It should be stressed that the differences in classification accuracy are minute for the descriptors "All", MOE, and GC and should be regarded as comparable considering a standard deviation of 1%. The CATS descriptor led to approximately 5% lower accuracy.

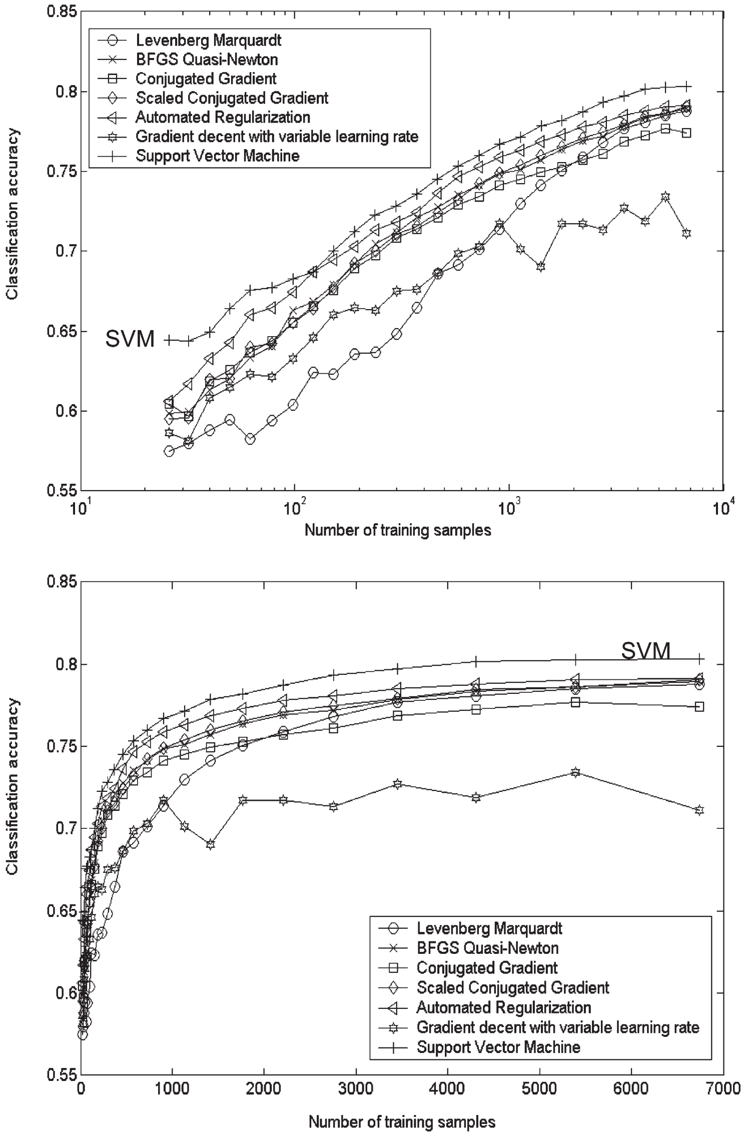


Figure 3. Average cross-validated prediction accuracy (fraction correct) of SVM and ANN classifiers optimized by various training schemes for GC descriptors (upper graph: logarithmic scale; lower graph: linear scale).

SVM training resulted in models showing slightly higher prediction accuracy than the ANN systems (Table 1). A 1−2% gain was observed, independent of the number of training samples and method used for neural network training. Figures 3 and 4 illustrate the dependency of the classification accuracy on the number of sample molecules used for training. In one experiment only GC descriptors were used (Figure 3), in a second study the combination of GC, MOE, and CATS descriptors was employed (Figure 4). With the GC descriptor the SVM estimator only slightly outperforms the neural networks (Figure 3). Similar results were obtained if only MOE or CATS descriptors were used for training (data not shown). The situation changed when all descriptors were used. With the complete descriptor set (525-dimensional) SVM clearly outperforms the neural network system (Figure 4). These results substantiate earlier findings that SVM performs better than ANN when large numbers of features or descriptors are used.[12]

A general observation was the fact that classification accuracy significantly improved with an increasing number of training samples, reaching a plateau in performance between 2000 and 3000 samples (Figures 3 and 4). The accuracy curves represent almost ideal learning behavior. It should be mentioned that the performance plateau observed does not reflect an inherent clustering of the data set, as training data subsets were randomly selected from the pool. The fraction correctly predicted grows from approximately 65% to 80% when the training set is increased by a factor of ~250. The combination of MOE, GC, and CATS descriptors improved classification accuracy by approximately two percent for SVM and by one percent for ANN compared to models based on individual descriptors. These results demonstrate that an optimal ANN training to a large extent depends on the number of training patterns available and the type of molecular descriptors used. For instance, for GC descriptors the best learning algorithm was training with

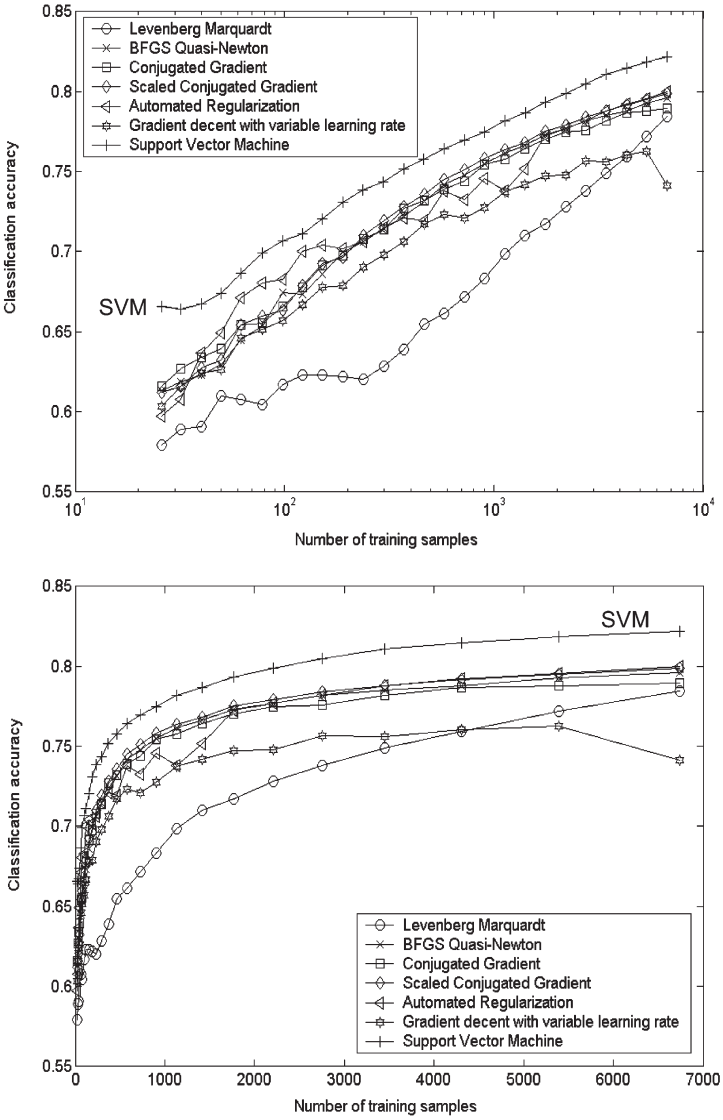


Figure 4. Average cross-validated prediction accuracy (fraction correct) of SVM and ANN classifiers optimized by various training schemes for the combination of GC, MOE, and CATS descriptors (upper graph: logarithmic scale; lower graph: linear scale).

automated regularization, but for the combination of GC, MOE, and CATS descriptors this algorithm was extremely slow and converged relatively unstable. In contrast, SVM generally performed more stably compared to ANN, with only a small increase in computation time for both sets of descriptors (Figures 3 and 4).

In a previous comparison of SVM to several machine learning methods by Holden and co-workers it was shown that an SVM classifier outperformed other standard methods, but a specially designed and structurally optimized neural network was again superior to the SVM model in a benchmark test.[13] This observation is supported by the observation that in the present study the set of molecules which were correctly classified by both SVM and ANN (mutual true positives) was 72% on average, and the fraction incorrectly classified by both systems (mutual false negatives) was 11%. 10% of the test data were correctly predicted by SVM but failed by ANN, and 6% were correctly classified by ANN but not by SVM using the full set of descriptors (GC+MOE+CATS). Examples of the latter two sets of molecules are shown in Figure 5. Clearly, the ANN classifier and the SVM classifier complement each other, and both methods could be further optimized, for example, by changing the SVM kernel or by exploring more sophisticated ANN architectures and concepts.

Fast classifier systems are mainly developed for first-pass virtual screening, in particular for identification (“flagging”) of potentially undesired molecules in very large compound collections.[2] Due to robust convergence behavior SVM seems to be well-suited for solving binary decision problems in molecular informatics, especially when a large number of descriptors is available for characterization of molecules. In this study we have shown that two drug-likeness estimators can produce complementary predictions. We recommend the parallel application of both predictive systems for virtual screening applications. One possibility to combine several estimators for “drug-likeness” or any other classification task is to employ a “jury decision”, e.g. calculate an ensemble

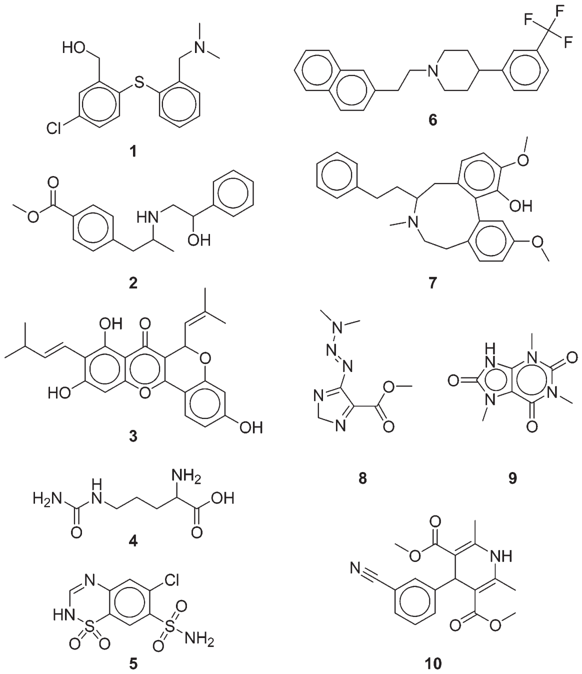


Figure 5. Examples of drugs correctly classified by ANN but not by SVM (structures 1−5), and drugs correctly classified by SVM but not by ANN (structures 6−10).

average.[38,39] As more and more different predictors become available for virtual screening a meaningful combination of prediction systems that exploits the individual strengths of the different methods will be pivotal for reliable compound library filtering.

CONCLUSION

It was demonstrated that the SVM system used in this study has the capacity to produce higher overall prediction accuracy than a particular ANN architecture. Based on this observation we conclude that SVM represents a useful method for classification tasks in QSAR modeling and virtual screening, especially when large numbers of input variables are used. The SVM classifier was shown to complement the predictions obtained by ANN. The SVM and ANN classifiers obtained for drug-likeness prediction are comparable in overall accuracy and produce overlapping, yet not identical sets of correctly and misclassified compounds. A similar observation can be made when two ANN models are compared. Different ANN architectures and training algorithms were shown to lead to different classification results. Therefore, it might be wise to apply several predictive models in parallel, irrespective of their nature, i.e., being SVM- or ANN-based. We wish to stress that our study does not justify the conclusion that SVM outperforms ANN in general. In the present work only a standard feed-forward network with a fixed number of hidden neurons was compared to a standard SVM implementation. Nevertheless, our results indicate that solutions obtained by SVM training seem to be more robust with a smaller standard error compared to standard ANN training. Irrespective of the outcome of this study, it is the appropriate choice of training data and descriptors, and reasonable scaling of input variables that determines the success or failure of machine learning systems. Both methods are suited to assess the usefulness of different descriptor sets for a given classification task, and they are methods of choice for rapid first-pass filtering of compound libraries.[40] A particular advantage of SVM is "sparseness of the solution". This means that an SVM classifier depends only on the support vectors, and the classifier function is not influenced by the whole data set, as it is the case for many neural network systems. Another characteristic of SVM is the possibility to efficiently deal with a very large number of features due to the exploitation of kernel functions, which makes it an attractive technique, e.g., for gene chip analysis or high-dimensional chemical spaces. The combination of SVM with a feature selection routine might provide an efficient tool for extracting chemically relevant information.

ACKNOWLEDGMENT

The authors are grateful to Norbert Dichter and Ralf Tomczak for setting up the LSF Linux cluster. Alireza Givehchi is thanked for assistance in installing the gecco! Web interface. This work was supported by the Beilstein-Institut zur Förderung der Chemischen Wissenschaften, Frankfurt.

REFERENCES AND NOTES

(1) Clark, D. E.; Pickett, S, D. Computational methods for the prediction of 'drug-likeness'. *Drug Discov. Today* **2000**, *5*, 49−58.

(2) Schneider, G.; Böhm, H.-J. Virtual screening and fast automated docking methods. *Drug Discov. Today* **2002**, *7*, 64−70.

(3) Wold, S. Exponentially weighted moving principal component analysis and projections to latent structures. *Chemomet. Intell. Lab. Syst.* **1994**, *23*, 149−161.

(4) Forina, M.; Casolino, M. C.; de la Pezuela Martinez, C. Multivariate calibration: applications to pharmaceutical analysis. *J. Pharm. Biomed. Anal.* **1998**, *18*, 21−33.

(5) *Neural Networks in QSAR and Drug Design*; Devillers, J., Ed.; Academic Press: London, 1996.

(6) Ajay; Walters, W. P.; Murcko, M. A. Can we learn to distinguish between "drug-like" and "nondrug-like" molecules? *J. Med. Chem.* **1998**, *41*, 3314−3324.

(7) Sadowski, J.; Kubinyi, H. A scoring scheme for discriminating between drugs and nondrugs. *J. Med. Chem.* **1998**, *41*, 3325−3329.

(8) Sadowski, J. Optimization of chemical libraries by neural networks. *Curr. Opin. Chem. Biol.* **2000**, *4*, 280−282.

(9) Schneider, G. Neural networks are useful tools for drug design. *Neural Networks* **2000**, *13*, 15−16.

(10) Sadowski, J. In *Virtual Screening for Bioactive Molecules*; Böhm, H.-J., Schneider, G., Eds.; Weinheim: Wiley-VCH: 2000; pp 117−129.

(11) Cortes, C.; Vapnik, V. Support-vector networks. *Machine Learning* **1995**, *20*, 273−297.

(12) Vapnik, V. *The Nature of Statistical Learning Theory*; Berlin: Springer, 1995.

(13) Burbidge, R.; Trotter, M.; Buxton, B.; Holden, S. Drug design by machine learning: support vector machines for pharmaceutical data analysis. *Comput. Chem.* **2001**, *26*, 5−14.

(14) Warmuth, M. K.; Liao, J.; Ratsch, G.; Mathieson, M.; Putta, S.; Lemmen, C. Active learning with Support Vector Machines in the drug discovery process. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 667−673.

(15) Wilton, D.; Willett, P.; Lawson, K.; Mullier, G. Comparison of ranking methods for virtual screening in lead-discovery programs. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 469−474.

(16) Todeschini, R.; Consonni, V. *Handbook of Molecular Descriptors*; Weinheim: Wiley-VCH: 2000.

(17) Ghose, A. K.; Crippen, G. M. Atomic physicochemical parameters for three-dimensional structure-directed quantitative structure−activity relationships 1. Partition coefficients as a Measure of hydrophobicity. *J. Comput. Chem.* **1986**, *7*, 565−577.

(18) Ghose, A. K.; Crippen, G. M. Atomic physicochemical parameters for three-dimensional structure-directed quantitative structure−activity

relationships 2. Modeling dispersive and hydrophobic interactions. *J. Comput. Chem.* **1987**, *27*, 21−35.

(19) Ghose, A. K.; Pritchett, A.; Crippen, G. M. Atomic physicochemical parameters for three-dimensional structure-directed quantitative structure−activity relationships 3. *J. Comput. Chem.* **1988**, *9*, 80−90.

(20) Schneider, G.; Neidhart, W.; Giller, T.; Schmid, G. "Scaffold-hopping" by topological pharmacophore search: a contribution to virtual screening. *Angew. Chem., Int. Ed. Engl.* **1999**, *38*, 2894−2896.

(21) Gasteiger, J.; Rudolph, C.; Sadowski, J. Automatic generation of 3D-atomic coordinates for organic molecules. *Tetrahedron Comput. Methods* **1990**, *3*, 537−547.

(22) Coleman, T. F.; Li, Y. A reflective Newton method for minimizing a quadratic function subject to bounds on some of the variables. *SIAM J. Optimization* **1996**, *6*, 1040−1058.

(23) Joachims, T. In Making large-scale SVM learning practical. *Advances in Kernel Methods − Support Vector Learning;* Schölkopf, B., Burges, C., Smola, A., Eds.; MIT-Press: Cambridge, MA, 1999; pp 41−56.

(24) Cristianini, N.; Shawe-Taylor, J. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*; Cambridge University Press: Cambridge, 2000.

(25) Burges, C. J. C. A tutorial on support vector machines for pattern recognition. *Data Mining Knowledge Discovery* **1998**, *2*, 121−167.

(26) Bishop, C. M. *Neural Networks for Pattern Recognition*; Oxford: Oxford University Press: 1995.

(27) Joachims, T. Learning to classify text using Support Vector Machines. Kluwer *International Series in Engineering and Computer Science 668*; Kluwer Academic Publishers: Boston, 2002.

(28) Duda, R. O.; Hart, P. E.; Stork, D. G. *Pattern Classification*; Wiley-Interscience: New York, 2000.

(29) Rumelhart, D. E.; McClelland, J. L.; The PDB Research Group. *Parallel Distributed Processing*; MIT Press: Cambridge, MA, 1986.

(30) Hagan, M. T.; Demuth, H. B.; Beale, M. H. *Neural Network Design*; PWS Publishing: Boston, 1996.

(31) Fletcher, R.; Reeves, C. M. Function minimization by conjugate gradients. *Comput. J.* **1964**, *7*, 149−154.

(32) Moller, M. F. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks* **1993**, *6*, 525−533.

(33) Dennis, J. E.; Schnabel, R. B. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*; Prentice-Hall: Englewood Cliffs, 1983.

(34) Hagan, M. T.; Menhaj, M. Training feedforward networks with the Marquardt algorithm. *IEEE Trans. Neural Networks* **1994**, *5*, 989−993.

(35) Foresee, F. D.; Hagan, M. T. Gauss−Newton approximation to Bayesian regularization. *Proceedings of the 1997 International Joint Conference on Neural Networks*; pp 1930−1935.

(36) MacKay, D. J. C. Bayesian interpolation. *Neural Comput.* **1992**, *4*, 415−447.

(37) Matthews, B. W. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim. Biophys. Acta* **1975**, *405*, 442−451.

(38) Krogh, A.; Sollich, P. Statistical mechanics of ensemble learning. *Phys. Rev. E* **1997**, *55*, 811−825.

(39) Baldi, P.; Brunak, S. *Bioinformatics − The Machine Learning Approach*; MIT Press: Cambridge, 1998.

(40) Byvatov, E.; Schneider, G. Support vector machine applications in bioinformatics. *Appl. Bioinf.* **2003**, *2*, 67−77.

CI0341161

7.3 SVM based Feature Selection for Characterization of Focused Compound Collections

Byvatov E., Schneider G.
J. Chem. Inf. Comput. Sci. 2004; 44(3):993-9

J. Chem. Inf. Comput. Sci. **2004,** *44,* 993−999 **993**

SVM-Based Feature Selection for Characterization of Focused Compound Collections

Evgeny Byvatov and Gisbert Schneider*

Institut für Organische Chemie und Chemische Biologie, Johann Wolfgang Goethe-Universität,
Marie-Curie-Strasse 11, D-60439 Frankfurt, Germany

Received December 9, 2003

Artificial neural networks, the support vector machine (SVM), and other machine learning methods for the classification of molecules are often considered as a "black box", since the molecular features that are most relevant for a given classifier are usually not presented in a human-interpretable form. We report on an SVM-based algorithm for the selection of relevant molecular features from a trained classifier that might be important for an understanding of ligand−receptor interactions. The original SVM approach was extended to allow for feature selection. The method was applied to characterize focused libraries of enzyme inhibitors. A comparison with classical Kolmogorov-Smirnov (KS)-based feature selection was performed. In most of the applications the SVM method showed sustained classification accuracy, thereby relying on a smaller number of molecular features than KS-based classifiers. In one case both methods produced comparable results. Limiting the calculation of descriptors to only the most relevant ones for a certain biological activity can also be used to speed up high-throughput virtual screening.

INTRODUCTION

Feature selection methods can help determine molecular descriptors that are important for the characterization of target-family specific classes of drugs and drug-like molecules by machine learning systems. Currently large numbers of descriptors are available for molecule characterization. Traditional feature selection methods such as forward and backward selection[1] or evolutionary algorithms[2] are computationally too expensive to be applied to very large descriptor sets directly. The most time-consuming step is retraining of the classifier after every modification of the set of selected features. This step needs to be reiterated sufficiently often before the process converges to the final set of features. Parallelizing computations is usually the only way to speed up the procedure.

An alternative approach is to select the important features prior to classifier training. In this case, the classifier needs to be trained only once for the selected features. Several techniques are known to implement this concept, e.g. correlation coefficients,[3,4] Fisher discriminant analysis,[1] and Kolmogorov-Smirnov (KS) statistics.[5] KS statistics was shown to be well-suited for feature selection in different fields of research.[6,7] Recently several model-dependent methods for feature selections were developed,[8] where the classifier is trained prior to feature selection, and features are selected based on a statistical model of the trained classifier. These methods have been predicted to outperform model-independent feature selection algorithms.[8]

For the present study we developed and applied a support vector machine (SVM)-based feature selection and compared it with a KS-based algorithm. An advantage of the SVM-based classification[9] in comparison to other methods, e.g. multilayered feed-forward neural networks,[1] is that the construction of the surface that separates classes of data depends only on the support vectors.[10] Support vectors are samples that are lying close to the border that separates two classes. Using only these samples can help increase the accuracy of the SVM prediction.[11] We extended the same principle to feature selection. Once an SVM classifier has been trained with all molecular descriptors, feature selection is based on the identified support vectors only, disregarding other samples.

The method was applied to feature selection from SVM classifiers for kinase inhibitors, factor Xa inhibitors, and thrombin inhibitors. The approach complements related work on "drug-likeness" prediction[12] and extends it to target- and target-family specific sets of inhibitors.

DATA AND METHODS

Data Sets. For SVM training and feature selection we used subsets of the COBRA database, version 2.1.[13] Three different splits of the COBRA collection were used for evaluation of the feature selection algorithms: 226 kinase inhibitors and 4479 noninhibitors; 227 factor Xa inhibitors and 4478 noninhibitors; and 227 factor Xa inhibitors and 195 thrombin inhibitors. The subset of kinase inhibitors represents a diverse set of molecules in that they are specific to a family of targets that differ significantly from each other. On the contrary, factor Xa and thrombin inhibitors are drug molecules which are specific for a single target. We expected that factor Xa and thrombin inhibitors should share a certain degree of similarity due to the similarity of the target binding sites.

Two sets of descriptors were calculated: 182 descriptors from MOE (Molecular Operating Environment)[14] and 225 topological pharmacophore (CATS) descriptors.[15] MOE descriptors include various 2D and 3D descriptors. 2D descriptors were physical properties, subdivided surface areas, atom and bonds counts, Kier-Hall connectivity and

* Corresponding author phone: +49-69 79829821; fax: +49-69 79829826; e-mail: gisbert.schneider@modlab.de.

10.1021/ci0342876 CCC: $27.50 © 2004 American Chemical Society
Published on Web 03/03/2004

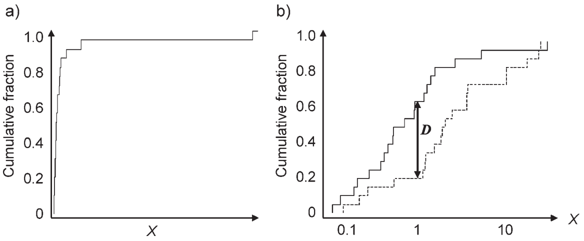


Figure 1. (a) Cumulative fraction plot. *X* denotes a molecular feature. (b) KS-test comparison. Cumulative fraction plots for two classes of data are shown by solid and dotted lines. *D* denotes the maximum difference of feature *X* values observed for the two classes.

Kappa Shape indices, adjacency and distance matrix descriptors, pharmacophore feature descriptors, and partial charges descriptors. 3D descriptors were potential energy descriptors; surface area, volume and shape descriptors; and conformational dependent charge descriptors. Before calculating MOE descriptors, single 3D conformers were generated by Corina.[16] CATS descriptors were calculated taking into consideration pairs of atom types separated by 0 up to 15 bonds. All descriptor columns were scaled to have zero mean and unit standard deviation.

Kolmogorov-Smirnov Statistics. KS-based statistics represent a model-independent method for feature selection. It is routinely used for feature selection from different data sets and features. Its main advantage over other methods is the independence from the particular statistical model that generates the data, in contrast to other methods, that perform well only if the data adopts certain statistics. For instance, “correlation coefficient”[3,4] based feature selection performs best if the data can be modeled by Gaussian mixtures,[1] and its accuracy drops otherwise. Very often it is impossible to correctly guess statistical models of the data a priori, which results in only approximately correct models. If the underlying statistics is not known or a Gaussian mixture model is not appropriate, KS statistics can be a method of choice.

In KS statistics each feature is first tested to have different statistics for class and nonclass samples. This is done by merging feature values for class and nonclass and building two separate cumulative fraction functions, one for class and one for nonclass. The cumulative fraction function represents the dependency of the percentage of samples whose feature values are below a certain threshold, on the position of the threshold value in the sorted list of feature values. An example of the cumulative function for the data set {0.08, 0.10, 0.15, 0.17, 0.24, 0.34, 0.38, 0.42, 0.49, 0.50, 0.70, 0.94, 0.95, 1.26, 1.37, 1.55, 1.75, 3.20, 6.98, 50.57} is given in Figure 1a. The maximum difference *D* of two cumulative functions for class and nonclass is then used as a measure for the significance of a distinguishing feature. An example of this measure is given in Figure 1b.

A KS statistics test is performed for all available features, which are then sorted with respect to the KS test results, and only the most relevant features are considered for further training.

SVM-Based Feature Selection. Usually feature selection algorithms are applied prior to the classifier training: A feature selection algorithm first selects a set of features and then a classifier is trained based on the features of this subset. Recently it was demonstrated that feature selection schemes, where the feature selection algorithm relies on the model that is created during training, produce better results.[8] Accordingly an alternative scheme for feature selection was suggested: The classifier is first trained using all available features. Then, the least important features are deleted. The drawback of this approach is that the trained classifier usually assumes a certain statistical model for the data, which might be only approximately correct. Current algorithms for nonlinear classifier training like artificial neural networks or SVM estimate a statistical model for the data sufficiently well to make this approach an alternative to model-independent feature selection.

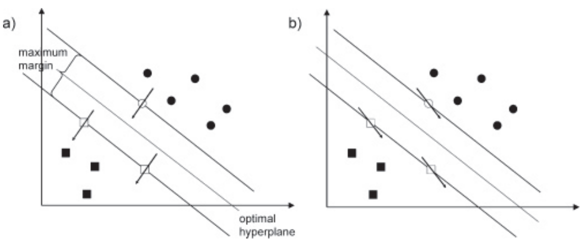


Figure 2. SVM-based feature selection. The optimal SVM hyperplane is shown with examples of class and nonclass samples (filled circles and squares). In the example support vectors are indicated by open symbols. For an estimation of the feature relevance the gradient (shown by arrows) of the feature change is calculated only for support vectors. (a) relevant features have a gradient perpendicular to the separating hyperplane; (b) irrelevant features.

The separating surface generated by SVM is given by

$$f(\mathbf{x}) = \sum_i a_i \cdot K(\mathbf{x}_\mathbf{i}^{\mathbf{sv}},\mathbf{x}) + b$$

Here a_i, b, and $\mathbf{x}_\mathbf{i}^{\mathbf{sv}}$are parameters of the SVM, determined during training. $\mathbf{x}_\mathbf{i}^{\mathbf{sv}}$ are support vectors, which represent a subset of the training samples that determine the separating surface. This surface corresponds to the linear separation in a very high-dimensional space, where data points are mapped during SVM training.[17] This mapping is determined solely by the kernel function $K(\boldsymbol{x},\boldsymbol{x}')$.[18] In this high-dimensional space the separating surface is given by

$$f(\mathbf{x}) = (\mathbf{w}\cdot\mathbf{x}) + b$$

where

$$\mathbf{w} = \sum_\mathbf{i} a_i \mathbf{x}_\mathbf{i}^{\mathbf{sv}}$$

is a normal vector of the separating hyperplane. To estimate the importance R_f of a feature to the accuracy of the SVM prediction we calculated a projection of the feature change in the mapped space to the normal of the SVM plane (Figure 2):

$$R_f = \frac{(\mathbf{w}\cdot\Delta\mathbf{x}_f)}{\Delta\mathbf{x}_f} = \frac{(\mathbf{w}\cdot\mathbf{x}_f^e) - (\mathbf{w}\cdot\mathbf{x}_f^b)}{\Delta\mathbf{x}_f} = \frac{\Delta f(\mathbf{x})}{\Delta\mathbf{x}_f} \rightarrow \frac{\partial f(\mathbf{x})}{\partial\mathbf{x}_f}$$

Calculating the derivative we obtain:

$$R_f(\mathbf{x}) = \frac{\partial f(\mathbf{x})}{\partial\mathbf{x}_f} = \sum_i a_i * \frac{\partial K(\mathbf{x}_\mathbf{i}^{\mathbf{sv}},\mathbf{x})}{\partial\mathbf{x}_f} + b$$

For estimating the relevance of a feature to classification we should calculate R_f only in the vicinity of the separating hyperplane. To achieve it we will sum R_f only over support vectors, extending the principle of SVM that the position of the classifying hyperplane depends only on support vectors:

$$R_f = \sum_j R_f(\mathbf{x_j}^{\mathbf{sv}}) = \sum_{i,j} a_i * \frac{\partial K(\mathbf{x_i}^{\mathbf{sv}},\mathbf{x_j}^{\mathbf{sv}})}{\partial \mathbf{x}_f} + b.$$

Empirically we observed that data normalization improved the performance in some cases; therefore, the final formula that we used to perform feature selection is

$$R_f = \sum_j R_f(\mathbf{x_j}^{\mathbf{sv}}) = \sum_j \left(\sum_i a_i * \frac{\partial K(\mathbf{x_i}^{\mathbf{sv}},\mathbf{x_j}^{\mathbf{sv}})}{\partial \mathbf{x}_f} + b \right) / \left(\sum_{i,k} a_i * \frac{\partial K(\mathbf{x_i}^{\mathbf{sv}},\mathbf{x_j}^{\mathbf{sv}})}{\partial \mathbf{x_k}} + b \right)$$

Summarizing, R_f was calculated for all features, and those features with low R_f value were excluded from the features used for training. It is important to note that R_f depends only on the support vectors.

For constructing SVM models we used the SVM-light package.[19] A fifth-order polynomial kernel was used in SVM training: $K(\mathbf{x}',\mathbf{x}) = (s(\mathbf{x}'\cdot\mathbf{x}) + 1)^5$. Training parameters ***s*** and ***C*** were optimized using a gradient decent-like algorithm to achieve maximum accuracy of prediction for the validation set. Parameter ***C*** is an internal parameter that is set prior to SVM training. It defines the tradeoff between the separating margin and the penalty for incorrect predictions.[17]

Model Validation. Classification accuracy was evaluated based on prediction accuracy and the correlation coefficient according to Matthews[20]

$$\mathrm{cc} = \frac{NP - \mathrm{OU}}{\sqrt{(N+O)(N+U)(P+O)(P+U)}}$$

where P, N, O, and U are the numbers of true positive, true negative, false positive, and false negative predictions, respectively. Active molecules with specific activity were considered as the "positive set", and the other molecules formed the "negative set". The values for cc can range from −1 to 1. A perfect prediction gives a correlation coefficient of 1. Different training and test subset were selected, 80% of samples for the training and 20% for the test. Ten cross-validations were performed to estimate average and standard deviation of the accuracy. Prediction accuracy and average value of ⟨cc⟩ were calculated for the test subsets.

RESULTS AND DISCUSSION

We compared two methods for feature selection, KS-based and SVM-based. Both methods were able to effectively select sets of the most relevant features. Figure 3 shows the dependency of the classification accuracy and Matthews correlation coefficient on the number of selected features for each subset. In all three sample applications the SVM-based feature selection method outperformed the KS-based approach, i.e., the classification accuracy remained at a high level even for small numbers of remaining features. The prediction accuracy dropped when the number of features fell between 100 and 200 for the KS-based method. In contrast, using the SVM-based method for feature selection we were able to go down to about 40 features with only a slight reduction in classification accuracy. This indicates potential advantages of the SVM-based method. Considering the error margins in the thrombin vs factor Xa classification, KS-based feature selection may be regarded as comparable to the SVM approach. This might have a relatively simple explanation: A large portion of features might be relatively easily discarded as "irrelevant" for correct classification. In this case no significant advantage of an SVM-based versus a KS-based scheme is observed. Still, when the number of features was below 100 SVM-based feature selection performed better. We wish to stress that a general statement about the relative usefulness of the two methods is not possible based on this single study. Also, we cannot fully exclude that the difference seen in Figure 3 between SVM and KS might in part result from different levels of parameter optimization.

Table 1 contains a list of the features which were selected being the most relevant for subset classification. Table 2 contains average property values calculated for the sets of inhibitors used in this study.

Both factor Xa and thrombin inhibitors are relatively large molecules containing characteristic fragments that are specific for binding to the S1 pocket of the trypsin-like serine proteases.[21] Typically, these fragments are positively charged. Most of the known faxtor Xa inhibitors exploit the S4 pocket and S3 "cation recognition pocket" of factor Xa to gain binding affinity.[22] A difference between the two classes of the molecules might be noted by observing the most relevant features in more detail. The distance of a positive charge on the one side and lipophilic, hydrogen-bond donor and acceptor groups on the other side was suggested being a key property for a distinction between factor Xa and thrombin inhibitors by our SVM-based feature selection. This property is most easily observed by comparing CATS descriptors for large distances. As expected, these descriptors are found in the top listed of the ranked features (Table 1a). These features can be highlighted in the two-dimensional structures of selective factor Xa inhibitors (Figure 4). Compound **1**[23] and compound **2**[24,25] have an approximately 3300-fold selectivity for factor Xa over thrombin and contain the topological pharmacophores selected by SVM. Structure **2** is a representative member of several covalent, peptide-derived bis-cation factor Xa inhibitors which were used for SVM-training. It is not surprising, therefore, that the most "relevant" molecular features according to the SVM classifier are found in these molecular structures. Structure **1** was not part of the training data, but some of the high-ranking features are present in this molecule, too.

Our compilation of kinase inhibitors represents a compound collection containing much broader activities than the collection of factor Xa and thrombin inhibitors. Looking at their average molecular weight and lipophilicity (clogP) one can conclude that they are smaller and more lipophilic than factor Xa and thrombin inhibitors (Table 2). This might explain the observation that in the list of top-ranking SVM features the topological descriptors are less prominent, and various van der Waals based estimations of surface charges were selected as "relevant" (Table 1b).

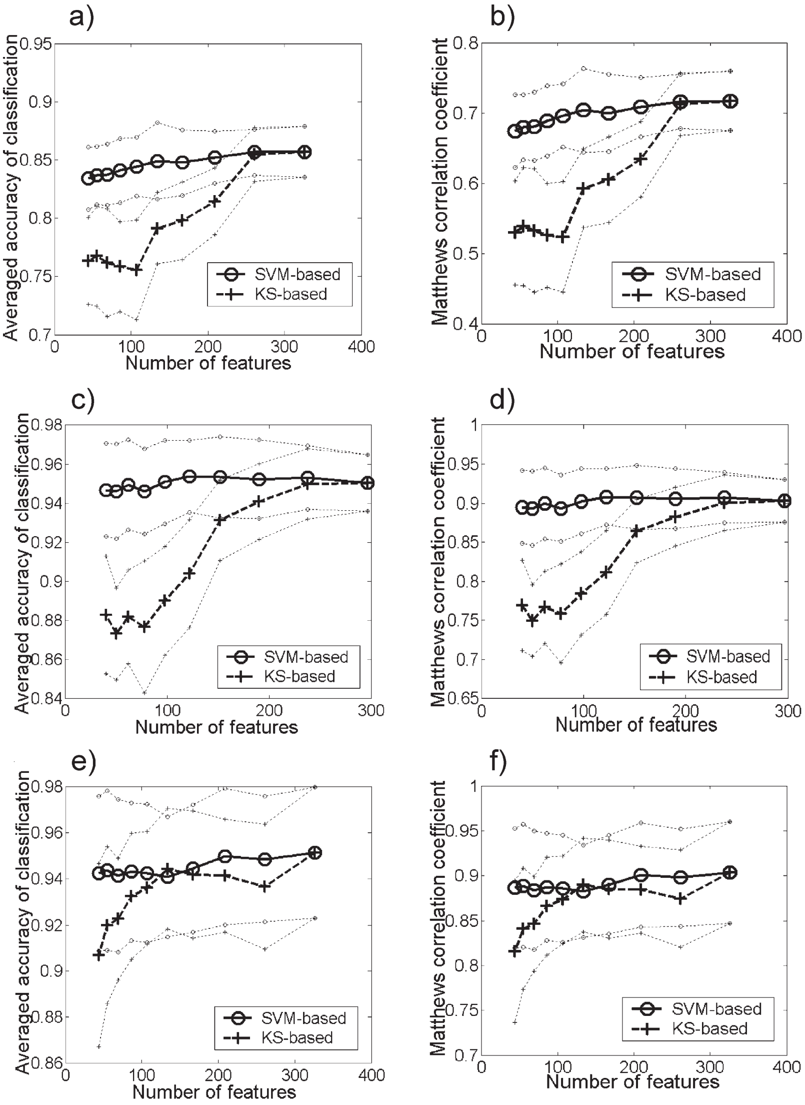


Figure 3. Results of feature selection by SVM- and KS-based algorithms. Matthews correlation coefficient and average classification accuracy are plotted as a function of the number of selected features. Standard deviations are shown as dotted lines. (a, b) Classification of kinase inhibitors versus the remainder of the COBRA data set. (c, d) Classification of factor Xa inhibitors versus the remainder of the COBRA data set. (e, f) Classification of factor Xa versus thrombin inhibitors.

Factor Xa inhibitors represent a relatively diverse set of molecules. Nonetheless, by examining their structures it is possible to assume that they have certain topological similarity. This could be a reason, why various topological descriptors are found within the first 20 most important descriptors (Table 1c). Surprisingly simple descriptors, like the number of aromatic atoms and aromatic bonds are also at the top of the list. Certainly, these simplistic descriptors cannot explain selectivity of factor Xa inhibitors, rather the whole list of “relevant” features must be taken into consideration if one tries to make sense out of a classifier system. This example demonstrates that feature selection does not necessarily deliver clear answers.

Although similar approaches were applied to perform SVM-based feature selection by Guyon and co-workers,[26] an advantage of our method is that feature selection was performed only based on the position of support vectors. It allows us to discard a large portion of data which is irrelevant for construction of the separating hyperplane. A potential additional advantage of our implementation is that classification of new molecules is quick and straightforward: computation time needed for a single molecule is approximately comparable to the time for reading its descriptors. Further information about computational efficiency of SVM can be found elsewhere.[18] Our results demonstrate that a central idea of SVM, namely the construction of a separating surface which is based only on support vectors, results in an efficient algorithm for feature selection when equipped with a feature selection scheme. We have successfully applied this algorithm to characterize groups of enzyme inhibitors. The algorithm was able to select crucial molecular features from a rather loosely defined compound class (kinase inhibitors) as well as features that might be relevant for inhibition of a particular target (factor Xa). It is important to mention that such feature selection methods do not explain why subsets can be classified or what the chemical explanation for an

Table 1. (a) Selected Features of Factor Xa Inhibitors versus Thrombin Inhibitors,[a] (b) Selected Features of Kinase Inhibitors,[b] and (c) Selected Features of Factor Xa Inhibitors[c]

feature	description
	(a)
SMR_VSA4	sum of v_i such that R_i is in (0.39,0.44]
CATS_207	correlation for the distance of 13 bonds between positive and lipophilic atoms
CATS_171	correlation for the distance of 11 bonds between acceptor and acceptor atoms
CATS_153	correlation for the distance of 10 bonds between donor and positive atoms
CATS_120	correlation for the distance of 8 bonds between lipophilic and lipophilic atoms
a_nN	number of nitrogen atoms
CATS_91	correlation for the distance of 6 bonds between donor and donor atoms
CATS_63	correlation for the distance of 4 bonds between donor and positive atoms
CATS_57	correlation for the distance of 3 bonds between positive and lipophilic atoms
CATS_50	correlation for the distance of 3 bonds between acceptor and acceptor atoms
CATS_47	correlation for the distance of 3 bonds between donor and acceptor atoms
SMR_VSA5	sum of v_i such that R_i is in (0.44,0.485]
PEOE_FPNEG	fractional negative polar van der Waals surface area. This is the sum of the v_i such that q_i is less than −0.2 divided by the total surface area. The v_i were calculated using a connection table approximation.
PEOE_VSA+3	sum of v_i where q_i is in the range [0.15,0.20).
CATS_187	correlation for the distance of 12 bonds between acceptor and positive atoms
CATS_33	correlation for the distance of 2 bonds between donor and positive atoms
Dens	mass density: molecular weight divided by van der Waals volume.
PEOE_VSA_PNEG	total negative polar van der Waals surface area. This is the sum of the v_i such that q_i is less than −0.2. The v_i were calculated using a connection table approximation.
PEOE_VSA-1	sum of v_i where q_i is in the range [−0.10,−0.05).
	(b)
VDistEq	If *m* is the sum of the distance matrix entries, then VdistEq is defined to be the sum of $\log_2 m$-$pi\log_2 pi/m$ where p_i is the number of distance matrix entries equal to *i*. [28]
diameter	largest value in the distance matrix[28]
CATS_188	correlation for the distance of 12 bonds between acceptor and negative atoms
SMR_VSA4	sum of v_i such that R_i is in (0.39,0.44].
VSA_other	approximation to the sum of VDW surface areas of atoms that are not a donor, acceptor, positive, negative, or hydrophobe
a_nCL	number of chlorine atoms
std_dim1	standard dimension 1: the square root of the largest eigenvalue of the covariance matrix of the atomic coordinates. A standard dimension is equivalent to the standard deviation along a principal component axis
FASA_H	fractional ASA_H calculated as ASA_H/ASA. Here, ASA_H is the water accessible surface area of all hydrophobic ($\|q_i\|<0.2$) atoms and ASA is the water accessible surface area of all atoms.
Q_VSA_FPOS	fractional positive van der Waals surface area. This is the sum of the v_i such that q_i is nonnegative divided by the total surface area. The v_i were calculated using a connection table approximation.
Q_VSA_FHYD	fractional hydrophobic van der Waals surface area. This is the sum of the v_i such that $\|q_i\|$ is less than or equal to 0.2 divided by the total surface area. The v_i were calculated using a connection table approximation.
radius	If r_i is the largest matrix entry in row *i* of the distance matrix *D*, then the radius is defined as the smallest of the r_i[28]
CATS_192	correlation for the distance of 12 between positive and lipophilic atoms
b_ar	number of aromatic bonds
a_aro	number of aromatic atoms
CATS_147	correlation for the distance of 9 bonds between donor and lipophilic atoms
a_nF	number of fluorine atoms
petitjian	value of (diameter-radius)/diameter.[28] Here *diameter* is the largest value in the distance matrix; radius is defined as follows, if r_i is the largest matrix entry in row *i* of the distance matrix *D*, then the radius is defined as the smallest of the r_i
petitjianSC	Petitjean graph shape coefficient as defined in ref 28
CATS_200	correlation for the distance of 13 bonds between donor and lipophilic atoms
CATS_186	correlation for the distance of 12 bonds between acceptor and acceptor atoms
	(c)
PEOE_VSA+1	sum of v_i where q_i is in the range [0.05,0.10).
balabanJ	Balaban's connectivity topological index[29]
b_ar	number of aromatic bonds
a_aro	number of aromatic atoms
SLogP_VSA1	sum of v_i such that L_i is in (−0.4,−0.2]
wienerPol	Wiener polarity number: half the sum of all the distance matrix entries with a value of 3 as defined in ref 30
vsa_acid	approximation to the sum of VDW surface areas of acidic atoms
a_acc	number of hydrogen bond acceptor atoms (not counting acidic atoms but counting atoms that are both hydrogen bond donors and acceptors such as −OH).

Table 1 (Continued)

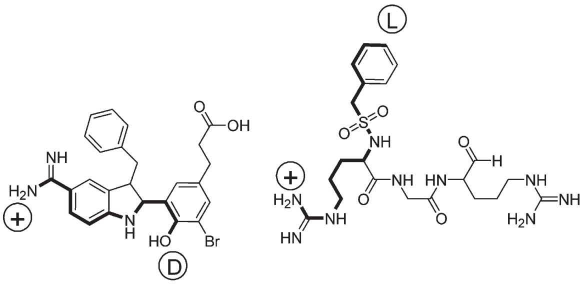
feature	description
reactive	indicator of the presence of reactive groups. A nonzero value indicates that the molecule contains a reactive group. The table of reactive groups was based on the Oprea set[31] and includes metals, phospho-, N/O/S−N/O/S single bonds, thiols, acyl halides, Michael acceptors, azides, esters, etc
b_rotR	fraction of rotatable bonds: b_rotN divided by b_count. Here b_count is the number of bonds including implicit hydrogens.
vsa_pol	approximation to the sum of VDW surface areas of atoms that are both hydrogen bond donors and acceptors, such as −OH
vsa_base	approximation to the sum of VDW surface areas of basic atoms.
Q_VSA_FPOS	fractional positive van der Waals surface area. This is the sum of the v_i such that q_i is nonnegative divided by the total surface area. The v_i were calculated using a connection table approximation.
Q_VSA_FHYD	fractional hydrophobic van der Waals surface area. This is the sum of the v_i such that $\|q_i\|$ is less than or equal to 0.2 divided by the total surface area. The v_i were calculated using a connection table approximation.
b_heavy	number of bonds between heavy atoms
CATS_91	correlation for the distance of 6 bonds between donor and donor atoms
SLogP_VSA2	sum of v_i such that L_i is in (−0.2,0].
Pmi	principal moment of inertia.
Zagreb	Zagreb index[32]
Chi1_qC	carbon connectivity index (order 1)[32]

[a] Calculation of the subdivided surface areas descriptors, like SMR_VSA, PEOE_VSA, was based on an approximate accessible van der Waals surface area calculation for each atom, v_i, along with some other atomic property p_i. The v_i were calculated using a connection table approximation. Each descriptor in a series was defined to be the sum of the v_i over all atoms i such that p_i is in a specified range (a,b]. For SMR_VSA p_i is R_i, which denotes the contribution to Molar Refractivity for atom i. For PEOE_VSA p_i is L_i, which denotes the contribution to logP(o/w) for atom i.[27] [b] Here, v_i is the van der Waals surface area of atom i (as calculated by a connection table approximation). R_i denotes the contribution to the molar refractivity of atom i.[27] [c] For definition of v_i, R_i, and L, see Table 1a.

Table 2. Average Property Values of the Three Sets of Inhibitors[a]

target	MW	PSA[b]/Å^2	clogP
factor Xa	490	132	2.9
thrombin	503	140	2.6
kinase	405	89	3.2

[a] Properties were calculated using MOE.[14−32] [b] PSA: polar surface area.



1 2

Figure 4. Examples of molecular features selected by SVM. Compounds 1 and 2 are selective factor Xa inhibitors. Two features are highlighted which were identified by an SVM classifier for discrimination between factor Xa and thrombin inhibitors. In structure **1** a positive charge (+) is separated by 10 bonds from a hydrogen-bond donor (D) site; in structure **2** a positive charge is separated by 13 bonds from a lipophilic point (L). These two-point pharmacophore features might be relevant for binding to the factor Xa active site pocket.

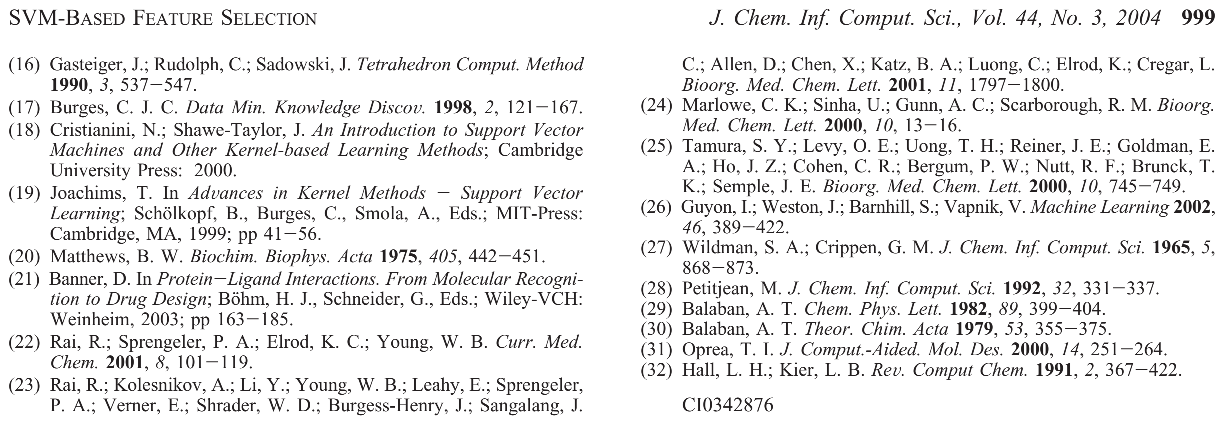
observed biological activity is. They might be suited for reducing the number of variables used in QSAR studies. It should be stressed that different feature selection algorithms tend to select different sets of “relevant” features. Therefore, the ranked list of features produced by the SVM-based method need not necessarily be more meaningful than a selection obtained by other methods, as one might conclude from the observation that the selected features resulted in a sustained high level of classification accuracy. It is possible that certain feature sets represent approximately the same chemical information, and as long as we only roughly describe a chemical agent using molecular descriptors, there will exist several almost equally suited partial solutions to the same classification task.

ACKNOWLEDGMENT

Uli Fechner is thanked for contributing his software for calculation of the CATS descriptors. This work was supported by the Beilstein-Institut zur Förderung der Chemischen Wissenschaften, Frankfurt.

REFERENCES AND NOTES

(1) Richard O. Duda; Hart, P. E.; Stork, D. G. *Pattern Classification*; Wiley-Interscience, 2000.
(2) Mitchell, M. *An Introduction to Genetic Algorithms* (*Complex Adaptive Systems*); MIT Press: 1998.
(3) Golub, T. R.; Slonim, D. K.; Tamayo, P.; Huard, C.; Gaasenbeek, M.; Mesirov, J. P.; Coller, H.; Loh, M. L.; Downing, J. R.; Caligiuri, M. A.; Bloomfield, C. D.; Lander, E. S. *Science* **1999**, *286*, 531−537.
(4) Furey, T. S.; Cristianini, N.; Duffy, N.; Bednarski, D. W.; Schummer, M.; Haussler, D. *Bioinformatics* **2000**, *16*, 906−914.
(5) Rassokhin, D. N.; Agrafiotis, D. K. *J. Mol. Graphics Modell.* **2000**, *18*, 370−384.
(6) Harter, H. L.; Khamis, H. J.; Lamb, R. E. *Commun. Statistics, Simulat. Comput.* **1984**, *13*, 293−323.
(7) Khamis, H. J. *J. Statistical Planning Inference* **1990**, *24*, 317−335.
(8) Kohavi, R.; John, G. H. *Artif. Intelligence* **1997**, *97*, 273−324.
(9) Byvatov, E.; Schneider, G. *Appl. Bioinf.* **2003**, *2*, 67−77.
(10) Vapnik, V. *The Nature of Statistical Learning Theory*; Springer: 1995.
(11) Cortes, C.; Vapnik, V. *Machine Learning* **1995**, *20*, 273−297.
(12) Byvatov, E.; Fechner, U.; Sadowski, J.; Schneider, G. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 1882−1889.
(13) Schneider, P.; Schneider, G. *QSAR Comb. Sci.* **2003**, *22*, 713−718.
(14) MOE; Chemical Computing Group Inc.: Montreal, 2003.
(15) Schneider, G.; Neidhart, W.; Giller, T.; Schmid, G. *Angew. Chem., Int. Ed. Engl.* **1999**, *38*, 2894−2896.



(16) Gasteiger, J.; Rudolph, C.; Sadowski, J. *Tetrahedron Comput. Method* **1990**, *3*, 537−547.

(17) Burges, C. J. C. *Data Min. Knowledge Discov.* **1998**, *2*, 121−167.

(18) Cristianini, N.; Shawe-Taylor, J. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*; Cambridge University Press: 2000.

(19) Joachims, T. In *Advances in Kernel Methods − Support Vector Learning*; Schölkopf, B., Burges, C., Smola, A., Eds.; MIT-Press: Cambridge, MA, 1999; pp 41−56.

(20) Matthews, B. W. *Biochim. Biophys. Acta* **1975**, *405*, 442−451.

(21) Banner, D. In *Protein−Ligand Interactions. From Molecular Recognition to Drug Design*; Böhm, H. J., Schneider, G., Eds.; Wiley-VCH: Weinheim, 2003; pp 163−185.

(22) Rai, R.; Sprengeler, P. A.; Elrod, K. C.; Young, W. B. *Curr. Med. Chem.* **2001**, *8*, 101−119.

(23) Rai, R.; Kolesnikov, A.; Li, Y.; Young, W. B.; Leahy, E.; Sprengeler, P. A.; Verner, E.; Shrader, W. D.; Burgess-Henry, J.; Sangalang, J. C.; Allen, D.; Chen, X.; Katz, B. A.; Luong, C.; Elrod, K.; Cregar, L. *Bioorg. Med. Chem. Lett.* **2001**, *11*, 1797−1800.

(24) Marlowe, C. K.; Sinha, U.; Gunn, A. C.; Scarborough, R. M. *Bioorg. Med. Chem. Lett.* **2000**, *10*, 13−16.

(25) Tamura, S. Y.; Levy, O. E.; Uong, T. H.; Reiner, J. E.; Goldman, E. A.; Ho, J. Z.; Cohen, C. R.; Bergum, P. W.; Nutt, R. F.; Brunck, T. K.; Semple, J. E. *Bioorg. Med. Chem. Lett.* **2000**, *10*, 745−749.

(26) Guyon, I.; Weston, J.; Barnhill, S.; Vapnik, V. *Machine Learning* **2002**, *46*, 389−422.

(27) Wildman, S. A.; Crippen, G. M. *J. Chem. Inf. Comput. Sci.* **1965**, *5*, 868−873.

(28) Petitjean, M. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 331−337.

(29) Balaban, A. T. *Chem. Phys. Lett.* **1982**, *89*, 399−404.

(30) Balaban, A. T. *Theor. Chim. Acta* **1979**, *53*, 355−375.

(31) Oprea, T. I. *J. Comput.-Aided. Mol. Des.* **2000**, *14*, 251−264.

(32) Hall, L. H.; Kier, L. B. *Rev. Comput Chem.* **1991**, *2*, 367−422.

CI0342876

7.4 SMILIB: Rapid assembly of combinatorial libraries in SMILES

Schüller A., Schneider G., **Byvatov E.**
QSAR Comb. Sci. 2003; 22:719-721

SMILIB: Rapid Assembly of Combinatorial Libraries in SMILES Notation

Andreas Schüller, Gisbert Schneider*, Evgeny Byvatov

Johann Wolfgang Goethe-Universität, Institut für Organische Chemie und Chemische Biologie, Marie-Curie-Strasse 11, D-60439 Frankfurt, Germany

Full Paper

A software tool was developed for fast combinatorial library enumeration (SMILIB). Its particular features are its simplicity to use, high flexibility in constructing combinatorial libraries and high speed of library construction. SMILIB offers the possibility to construct very large combinatorial libraries using the flexible and portable SMILES format. Libraries are generated at rates of approximately 30,000 molecules per minute. Combinatorial building blocks are attached to scaffolds by means of linkers rather than to concatenate them directly. This allows for creation of easily customized libraries using linkers of different size and chemical nature. A web interface for a limited web-based version of the software is available at URL: www.modlab.de. An unlimited binary version of SMILIB for command line execution on Linux systems is available from this URL.

Rapid construction of virtual combinatorial products is a prerequisite for *in silico* library enumeration and design [1, 2]. It has been demonstrated that for library design purposes virtual screening of combinatorial reaction products is usually preferable to purely educt-based screening and filtering [3, 4], and several commercially available software suites offer a possibility for this kind of combinatorial enumeration. Here we present a freely available software tool (SMILIB) which was developed to offer a means for straightforward library assembly and may serve as a basis for subsequent virtual screening and filtering of enumerated combinatorial libraries. The main features of SMILIB are its flexibility in constructing combinatorial libraries and high speed of library construction. SMILIB offers the possibility to rapidly construct very large combinatorial libraries using the compact and portable SMILES format [5]. Libraries are created at rates of approximately 30,000 molecules per minute on a Linux-based personal computer. A web interface for SMILIB is available at URL: http://www.modlab.de. For performance reason the number of reaction products is restricted to 10,000 molecules using the web-interface. An unlimited binary version for Linux systems is also available from the URL. The SMILIB binary was compiled using GNU C Compiler (GCC) 3.2 on SuSE 8.1 Linux running kernel 2.4.19.

Construction of combinatorial products with SMILIB follows the concept of “scaffolds”, ”linkers“ and ”building blocks“ (Figure 1) [6]. Building blocks are attached to the scaffold via linker groups. This allows for different chemical reactions to be considered implicitly by using different linker types and generic building block collections. It also simplifies library enumeration since connecting functional groups may be completely left out in the set of building blocks. An advantage of this conceptual idea of a combinatorial library is its simplicity and ease of implementation yielding very fast code. An obvious downside is the fact that realistic chemical reactions cannot be modeled deliberately. For example, ring formation during building block attachment cannot be modeled by SMILIB; or scaffold formation during the combinatorial reaction.

SMILIB uses basic ANSI C string functions to perform virtual reactions of building blocks and linkers with

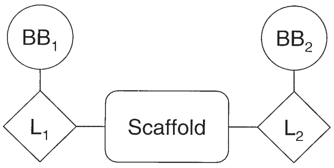


Figure 1. Schematic composition of a virtual reaction product. BB denotes building block, L denotes linker.

* To receive all correspondence: Johann Wolfgang Goethe-Universität, Institut für Organische Chemie und Chemische Biologie, Marie-Curie-Strasse 11, D-60439 Frankfurt, Germany, phone +49(69)798—29821, fax +49(69)798—29826, E-mail: G.Schneider@chemie.uni-frankfurt.de

Key words: combinatorial chemistry, library design, virtual screening

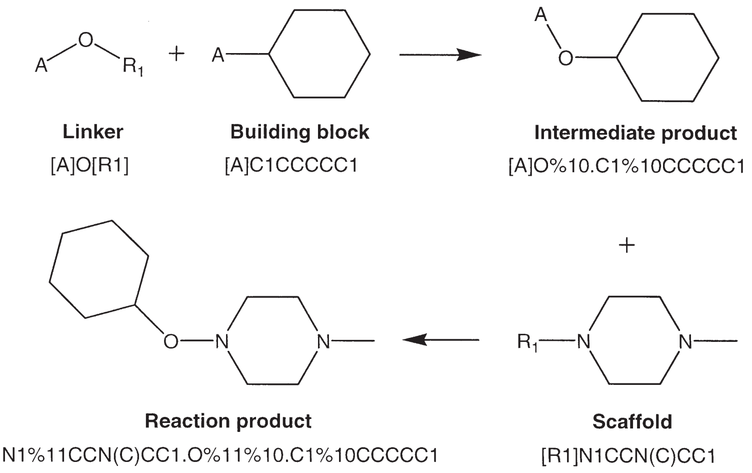


Figure 2. A virtual reaction along with corresponding SMILES: The building block is connected to a linker having the building block's A-group react with the linkers R1-group forming intermediate product and a virtual A-R1 by-product (neglected). The intermediate product's A-group then undergoes reaction with the scaffolds R1-group yielding in the final reaction product and a second A-R1 by-product (also neglected).

scaffolds. All educts have to be formulated in an enhanced notation of SMILES: Special labels "[R1]", "[R2]", "[R3]", etc. and "[A]" were introduced to specify sites of variability (R) and attachment sites (A) respectively, the latter being necessary to allow directional concatenation of building blocks to linkers and linkers to scaffolds (Figure 2). Basic help for generating SIMLIB compliant SMILES is given with these examples:

Scaffolds:
[R1]N2CCN([R2])C1=CC=CC=C1C2
[R2]N(C3=C2C=CC=C3)CC12CCN([R1])CC1
[R1]N1CCN([R2])CC1
Linkers:
[A][R1] ("pseudo linker")
[A]S(= O)(N[R1]) = O
[A]O[R1]
Building blocks:
[A]C
[A]CC1=CC=CC=C1
S(CC[A])C

In order to facilitate flexible library generation each individual product needs to be explicitly determined by a reaction scheme. Similar to a connection table, the reaction scheme determines the constituents of a virtual reaction product by referring to them by numbers. Each line of the reaction scheme thus refers to a number-encoded virtual chemical reaction yielding a desired combinatorial product. A sample reaction scheme for a complete six-membered virtual library from one scaffold, two linkers, three building blocks and one R-group on the scaffold is given by:

1	1	1
1	1	2
1	1	3
1	2	1
1	2	2
1	2	3

The first column specifies the scaffold number, the second column determines the linker group, and the third column specifies the building block. A more detailed guideline of how to use SMILIB and generate reaction schemes is available from the web site. A tool for generating the reaction scheme is also available on this web site.

Reaction products are generated by concatenation of input SMILES strings using "unsatisfied" ring closures [7]. For example, according to the SMILES convention, either of the following notations for ethane is valid: "CC" or "C1.C1". Following this scheme, SMILIB uses unsatisfied ring closures to form chemical bonds between the constituents of a reaction product as shown in Figure 2. The resulting SMILES look unconventional – yet they are perfectly valid. We tested compatibility to the following programs: CLIFF molecule file converter [8], Molecular Operating Environment (MOE) [9], ChemDraw molecule editor [10], and CORINA conformer generator [11].

SMILIB is intended to support molecular designers by providing a fast means for virtual library generation. Its principal strength is full or partial library enumeration. Certainly, even the fastest enumeration method combined with the largest storage capacity will be limited by the maximal upper size of the virtual compound library. To avoid exhaustive library enumeration, a trend in virtual combinatorial library generation and exploration is to perform a guided search in very large chemical spaces [2]. SMILIB can also be used for this task. In this scenario, predictive QSAR models, e.g. for "drug-likeness", "frequent-hitter" liability or binding affinity [12, 13], are coupled to virtual molecule generators like SMILIB, and only small compound sub-sets are actually assembled in silico, rather than the complete library. In an iterative process the overall quality of the sub-set is improved [14, 15]. Several such optimization protocols have been suggested and successfully applied [16–20], and SMILIB can be used to support these activities.

Acknowledgements

Uli Fechner and Norbert Dichter are thanked for stimulating discussions and assistance. This research was supported by the Beilstein-Institut zur Förderung der Chemischen Wissenschaften, Frankfurt.

References

[1] A. K. Ghose, V. N. Viswanadhan (Eds.), *Combinatorial Library Design and Evaluation: Principles, Software Tools, and Applications in Drug Discovery*, Marcel Dekker, New York, **2001.**

[2] G. Schneider, *Curr. Med. Chem.* **2002**, *9*, 2095–2101.

[3] V. J. Gillet, *Mol. Divers.* **2002**, *5*, 245–254.

[4] E. A. Jamois, *Methods Mol. Biol.* **2002**, *4*, 576–583.

[5] D. J. Weininger, *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31–36.

[6] J. M. Barnard, G. M. Downs, A. von Scholley-Pfab, R. D. Brown, *J. Mol. Graph. Model.* **2000**, *18*, 452–463.

[7] J. M. Barnard, *Reactions to Markush*, presentation at MUG 2000 (Daylight User Group Meeting), Santa Fe, NM, 24 Feb 2000.

[8] CLIFF molecule file converter, Molecular Networks GmbH, Nägelsbachstraße 25, 91052 Erlangen, Germany.

[9] Molecular Operating Environment (MOE), Chemical Computing Group, 1010 Sherbrooke St. West, #910, Montreal, Canada, H3A 2R7, (http://www.chemgroup.com).

[10] ChemOffice Ultra, CambridgeSoft Corporation, 100 CambridgePark Drive, Cambridge, MA 02140 USA, (http://www.cambridgesoft.com).

[11] J. Gasteiger, C. Rudolph, J. Sadowski. *Tetrahedron Comp. Method.* **1990**, *3*, 537–547. (http://www2.chemie.uni-erlangen.de/software/corina/index.html)

[12] W. P. Walters, M. T. Stahl, M. A. Murcko, *Drug Discov. Today* **1998**, *3*, 160–178.

[13] H.-J. Böhm, G. Schneider (Eds.), *Virtual Screening for Bioactive Molecules*, Wiley-VCH, Weinheim, **2000.**

[14] G. Schneider, S.-S. So. *Adaptive Systems in Drug Design*, Landes Bioscience, Georgetown, **2003.**

[15] J. Bajorath, *Nature Rev. Drug. Discov.* **2002**, *1*, 882–894.

[16] S. Grüneberg, M. T. Stubbs, G. Klebe, *J. Med. Chem.* **2002**, *45*, 3588–3602.

[17] R. P. Sheridan, S. K. Kearsley, *Drug Discov. Today* **2002**, *7*, 903–911.

[18] R. S. Pearlman, K. M. Smith, *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 28–35.

[19] C. A. Nicolaou, S. Y- tamura, B. P. Kelley, S. I. Bassett, R. F. Nutt, *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 1069–1079.

[20] V. J. Gillet, P. Willett, J. Bradshaw, *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 165–179.

[21] D. K. Agrafiotis, V. S. Lobanov, R. F. Salemme, *Nature Rev. Drug. Discov.* **2002**, *1*, 337–346.

Received: May 12, 2003; Accepted: June 9, 2003

© 2003 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim

7.5 From Virtual to Real Screening for Novel D_3 Dopamine Receptor Ligands

Byvatov E., Sasse B.C., Stark H., Schneider G.
ChemBioChem. *in press*

From Virtual to Real Screening for D_3 Dopamine Receptor Ligands

Evgeny Byvatov[a], Britta C. Sasse[b], Holger Stark[b], Gisbert Schneider[a]*

[a] E. Byvytov, Prof. Dr. G. Schneider, Johann Wolfgang Goethe-Universität, Institut für Organische Chemie und Chemische Biologie,Marie-Curie-Str. 11, D-60439 Frankfurt am Main, Germany

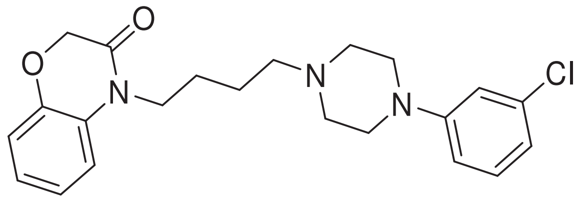
[b] B. C. Sasse, Prof. Dr. H. Stark, Johann Wolfgang Goethe-Universität, Institut für Pharmazeutische Chemie, Marie-Curie-Str. 9, D-60439 Frankfurt am Main, Germany

* author to whom correspondence should be addressed:

Prof. Dr. Gisbert Schneider
Beilstein Endowed Chair for Cheminformatics
Johann Wolfgang Goethe-Universität
Fachbereich Chemische und Pharmazeutische Wissenschaften
Institut für Organische Chemie und Chemische Biologie
Marie-Curie-Str. 11
D-60439 Frankfurt, Germany

Tel: +49 (0) 69 798 29821/2
Fax: +49 (0) 69 798 29826
Email: G.Schneider@chemie.uni-frankfurt.de
URL: www.modlab.de

Key words:
Bioinformatics, similarity, pharmacophore, G-protein coupled receptor, drug discovery



ABSTRACT. Iterative virtual screening cycles using Support Vector Machines (SVM) were successfully applied to ligand-based searching for novel ligands. The approach offers a rapid way to identify novel lead structure candidates with minimal experimental effort even in the absence of receptor-structure information. Virtual screening was performed in two consecutive cycles. In the first stage, an SVM was trained for prediction of D_3 receptor-selective ligands. Based on the prediction of this virtual filter, twelve compounds were tested for binding affinity at D_2 and D_3 receptors. In the second stage, a similarity search was performed with the most promising candidate molecule from the first round as the query. Four out of five compounds from the final hit list exhibited nanomolar affinity at the D_3 receptor including a novel scaffold structure. The K_i value of the best molecule was 40 ± 6 nM.

Imbalance of the dopaminergic system is involved in various neurological and neuropsychiatric disorders, e.g. Parkinson's disease, schizophrenia, and drug abuse.[1] Selective attraction of one dopamine receptor subtype could represent an improved therapeutic approach or at least a good way to evaluate the (patho)physiological functions of this subtype in the disorder. Here we focused on the dopamine D_3 receptor since this subtype displays in several diseases an important role on neuroregulation and possesses a distinct localisation in the central nervous system.[2] As D_3 receptors display high sequence identity to D_2 receptors, cross-reactivity is a problem for most compounds used. Although this field of research has been worked out for decades many lead structures are unsatisfying concerning selectivity. Since numerous described compounds with diverse structural elements showed some D_3 receptor preference we focused on these elements first by virtual and then by real screening of the most promising compounds to find new lead candidates for further optimization.

Virtually screened synthetic compounds from collections of Specs (release January 2004, Specs, 2628 XH Delft, The Netherlands) and Interbioscreen (IBS) (release February 2004, Interbioscreen Ltd., 121019 Moscow, Russia) were investigated as potentially selective ligands at dopamine D_3 receptors. We performed this screening by using analogues of BP897 (**1**), a D_3 receptor-preferring partial agonist in clinical development, and related structures as a starting point (Chart 1). Virtual screening was performed in two stages. In the first stage, we trained a Support Vector Machine (SVM) on the reference set and constructed a filter for D_3 receptor-selective ligands. Based on the prediction of this virtual filter twelve compounds from the IBS collection were tested for binding affinity at D_2 and D_3 receptors. In the second stage, we performed a similarity search with the most promising candidate molecule from the first round against the Specs collection. The parameters for this similarity search were extracted from the SVM model of the BP897 analogues. Four out of five compounds exhibited nanomolar affinity at the D_3 receptor including a novel scaffold structure. The K_i value for the best molecule was 40 ± 6 nM.

Ligand-based virtual screening. As a reference active set we used analogues of BP897 and related structures.[3] The compounds from this set possess the following features: i) a lipophilic amine moiety, i.e. phenylpiperazine in BP897, ii) a spacer, usually a linear tetramethylene chain, and iii) a hydrophobic residue connected by an amide bond, which has proven to be favorable for high receptor affinity.[3] In order to fulfill structural requirements for high-affinity binding, the basic nitrogen connected to the aryl group through an aliphatic linker was preserved. For all compounds in this series K_i values of D_2 and D_3 receptor affinities were screened in radioligand binding assays as described.[3]

[Chart 1]

Compounds were encoded by three-point pharmacophore (3PP) fingerprints available from the MOE software suite.[4] For the first virtual screening round, an SVM was trained on the prediction of potential D_3 receptor ligands. As "active" compounds we defined molecules which have measured K_i values below 1 µM for the D_2 or D_3 receptor (331 out of 395 reference compounds).[5] For cross-validation this active set was split into four non-overlapping subsets. During validation we "mimicked" a real screening experiment by addition of compounds known to bind D_2 or D_3 receptor to the screening database and estimated the efficiency with which these compounds were retrieved from the screened database. For this, we ranked all screening compounds based on the SVM predictions and optimized SVM parameters, so that compounds that we mixed with the screening data were at the top of the ranked list.[6-8] The observed enrichment gave an estimation of what is the expected percentage of active compounds from the IBS dataset among the top 1% of the ranked compounds. In the cross-validation study 50.6 ± 1.3% of the known active compounds were retrieved within 1% of the IBS collection -- a result which is significantly above random screening. The training procedure with parameter optimization lasted less than 30 minutes on a Linux cluster with 16 CPUs.

Application of "active learning" further increased the enrichment to 91.8 ± 1.2% of validation actives in the 1% of the ranked IBS collection (for details of the SVM training procedure and the active learning concept, cf. Supporting Information). This was a consequence of the more fine-grained compound sampling from the neighborhood of the known actives in pharmacophore space.

Selection of D_3 receptor-specific ligands. We trained a regression SVM for prediction of the logarithm of the ratio between K_i values for D_2 and D_3 receptors. The $<q^2>$ of the four-fold cross-validation was 0.40 ± 0.15. The relatively low $<q^2>$ is explained by the marked similarity between D_2 and D_3 receptor binding behavior.[2] The final prediction system was a combination of the two virtual filters described above: binary SVM optimized with active learning, and regression SVM. First, we selected compounds that were similar to the reference set and then we ranked them according to the predicted $\log(K_iD_3 / K_iD_2)$ to pick up potential D_3 receptor-selective compounds. The list of the selected molecules obtained was manually further processed to exclude compounds with potentially reactive groups or poor solubility. Compounds which are too similar to the reference set were also excluded, in order to identify compounds with novel scaffolds. K_i measurement followed a similar protocol as for the BP897 analogues.[3]

Results and Conclusions. Individual compounds exhibited preferential binding at D_3 receptor, although K_i values for most of the molecules are in the micromolar range if any could be determined at all (cf. Supporting Information). This observation can be explained by the bias introduced during manual post-selection of molecules. We avoided a pronounced similarity to BP897-like compounds, which obviously resulted in lowering the D_2 and D_3 binding activity.

[Chart 2]

In order to further increase D_3 receptor affinity we optimized compound **2** using a similarity searching approach (Chart 2). Molecule **2** was the only ligand found in the first virtual screening round with an experimental $K_i < 2$ μM at the D_3 receptor, and $K_i \geq 2$ μM at the D_2 receptor. For similarity calculation we employed a modified distance metric for 3PP fingerprints space, where fingerprints were weighted based on their importance in our SVM regression model (cf. Supporting Information). This procedure allowed for the selection of compounds that are similar to **2**, focusing on features that were considered being important for interaction with the receptor. Very similar compounds and compounds with reactive groups were again manually excluded. The testing results for the selected molecules are given in Table 2. The chemical structures of the tested molecules are shown in Figure 1, aligned at their basic nitrogen which is assumed to be essential for this type of G-protein receptor binding. As can be seen from Table 1, all active compounds possess a common pattern of the aromatic residue coupled to a potential hydrogen-bond donor and separated by an aryl moiety from the positively charged amine with an adjacent ring system.

[Table 1]

Although the most active compound in this series **4** shows nanomolar affinity at the D_3 receptor accompanied by a 10-fold D_3 receptor preference in comparison to its D_2 receptor affinity it must be stressed that **4**[9] and **5** are quite similar to the reference set. By the use of compound libraries one can hardly expect to retrieve totally unknown lead candidates. Nevertheless, compounds **2**, **3**, **6**, and especially **7** disclose some novel structural features resulting in first hits as well as promising new leads for dopamine receptor subtype ligands in this overcrowded area of drug development. Together with the other data obtained from virtual and real compound screening (cf. Supporting Information) one may extract structural characteristics which have not or have only rarely been applied to dopamine D_3 receptor ligands. Compounds **6** and **7** already display slight D_3 receptor preferences showing the success of our approach, and give for **7** good hopes for further optimization that is distinct from well-known structure-activity relationships. For the first time, iterative virtual screening cycles using SVM were successful applied to entirely ligand-based searching for novel ligands. The concept offers a rapid way to identify lead structure candidates with minimal experimental effort even in the absence of receptor-structure information.

[Figure 1]

Acknowledgement. We thank Dr. P. Sokoloff (INSERM, Paris), and Dr. J. Shine (The Garvan Institute of Medical Research, Sydney) for providing cell lines expressing dopamine D_3 and D_2 receptors, respectively. Technical assistance of C. Bereiter is highly acknowledged. This research was supported by the Beilstein-Institut zur Förderung der Chemischen Wissenschaften, Frankfurt am Main.

Supporting Information: Construction of a homology model for the D_3 receptor, docking of compounds into the constructed homolgy model, and analysis of predicted binding modes are provided in the Supporting Information. Supporting Information also includes full details of SVM training and the binding studies. This material is available free of charge from the authors.

References

[1] a) A. E. Hackling, H. Stark, *ChemBioChem* **2002**, *3*, 946-961; b) J. N. Joyce, *Pharmacol. Ther*. **2001**, *90*, 231-259.

[2] a) J. R. Bunzow, H. H. Van Tol, D. K. Grandy, P. Albert, J. Salon, M. Cristie, C. A. Machida, K. A. Neve, O. Civelli, O. *Nature (London)* **1988**, *336*, 783-787; b) P. Sokoloff, B. Giros, M. P. Martres, M. L. Bouthenet, J. C. Schwartz, *Nature (London)* **1990**, *347*, 146-151; c) H. H. Van Tol, J. R. Bunzow, H. C. Guan, R. K. Sunahara, P. Seeman, H. B. Niznik, O. Civelli, O., *Nature (London)* **1991**, *350*, 610-614.

[3] a) U. R. Mach, A. E. Hackling, S. Perachon, S. Ferry, C. G. Wermuth, J. C. Schwarz, P. Sokoloff, H. Stark, *ChemBioChem* **2004**, *5*, 508-518; b) A. E. Hackling, R. Ghosh, S. Perachon, A. Mann, H. D. Höltje, C. G. Wermuth, J. C. Schwarz, W. Sippl, P. Sokoloff, H. Stark, *J. Med. Chem.* **2003**, *46*, 3883-3899.

[4] MOE (Molecular Operating Environment) version 2004.05 by Chemical Computing Group, Inc., Monteal. URL: http://www.chemcomp.com

[5] M. Pilla, S. Perachon, F. Sautel, F. Garridol, A. Mann, C. G. Wermuth, J. C. Schwartz, B. J. Everitt, P. Sokoloff, *Nature (London)* **1999**, *400*, 371-375.

[6] a) G. Schneider, P. Schneider, in *Chemogenomics in Drug Discovery*; H. Kubinyi, G. Müller, Eds.; Wiley-VCH: Weinheim, **2004**, pp. 341-376; b) E. Byvatov, G. Schneider. *Appl. Bioinformatics* **2003**, *2*, 67-77.

[7] J. Joachims, in *Advances in Kernel Methods - Support Vector Learning;* B. Schölkopf, C. Burges, A. Smola, Eds.; MIT-Press: Cambridge, **1999**, pp. 41-56.

[8] a) E. Byvatov, G. Schneider, *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 993-999; b) E. Byvatov, U. Fechner, J. Sadowski, G. Schneider, *J. Chem. Inf. Comput. Sci*. **2003**, *4*, 1882-1889.

[9] a) M. J.Mokrosz, P. Kowalski, T. Kowalska *et al. Arch. Pharm. (Weinheim)* **1999**, *332*, 373-379; b) M. J. Mokroz, S. Charakchieva-Minol, P. Kowalski, *Arch. Pharm. (Weinheim)* **2001**, *334*, 25-29.

Legend to the figure

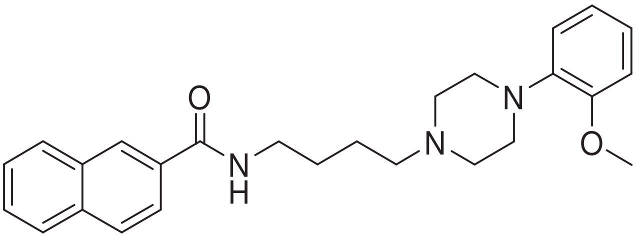
Figure 1.
Compounds selected for testing based on similarity to compound **2**. Structures were aligned according to the position of the basic nitrogen (dotted line). Three different parts of the molecules were distinguished: A) an aromatic moiety, B) an aliphatic linker, C) a hydrophobic part connected through a basic nitrogen.

Table 1. Dopamine receptor affinities of compounds from the first virtual screening round (from IBS catalogue)

Molecule No.	K_i (D_2) ± SD [nM][a]	n[b]	K_i (D_3) ± SD [nM]a	n[b]
3	1414±516	2	1408±1068	2
4	554± 97	4	40±6	4
5	417±60	8	139±17	5
6	201±48	8	96±21	7
7	4395±497	6	914±307	6

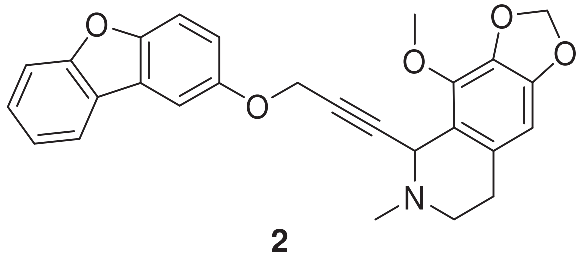
[a] K_i values (mean value with standard deviation (SD)) were measured in CHO cells stably expressing hD_{2s} and hD_3 receptors by using [³H]spiperone.
[b] Number of experiments.

Chart 1.



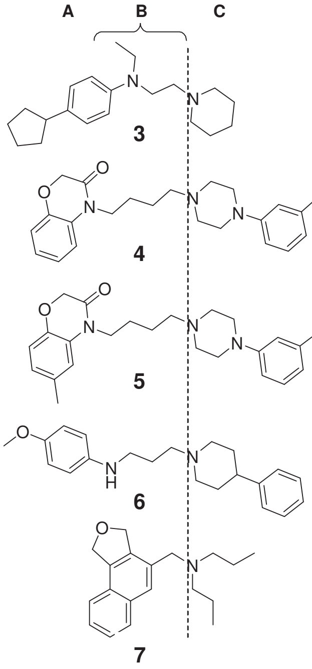
1

Chart 2.



2

Figure 1.



Supporting Information

From Virtual to Real Screening for D_3 Dopamine Receptor Ligands

Evgeny Byvatov[1], Britta C. Sasse[2], Holger Stark[2], Gisbert Schneider*,[1]

Johann Wolfgang Goethe-Universität, Institut für Organische Chemie und Chemische Biologie,Marie-Curie-Str. 11, D-60439 Frankfurt am Main, Germany, and Johann Wolfgang Goethe-Universität, Institut für Pharmazeutische Chemie, Marie-Curie-Str. 9, D-60439 Frankfurt am Main, Germany

Contents

1. Brief description of experimental procedures (Synopsis)
2. Experimental results of the first screening round
3. Docking of ligands into a homology model
4. SVM training and active learning
5. Regression SVM model for predicting the $\log \frac{K_i^{D3}}{K_i^{D2}}$ ratio
6. Compound ranking based on the similarity to the reference compounds
7. Homology model of the D_3 receptor
8. Binding studies
9. References

[1] Institut für Organische Chemie und Chemische Biologie, Johann Wolfgang Goethe-Universität

[2] ~~Institut für Pharmazeutische Chemie, Johann Wolfgang Goethe-Universität~~

1. Brief description of experimental procedures

Data sets and descriptors. 395 compounds with reported affinity at D_2 and D_3 receptors served as the SVM training sets. These compounds were synthesized analogues of BP897 and related structures reported in literature.[1] Each compound was represented by a fingerprint of 3PP pharmacophores using MOE version 2004.05 (Chemical Computing Group, Montreal). The individual 3PP pharmacophore was represented as a triangle. We considered all possible triangles with their vertexes located at atom centers. Presence or absence of a certain triangle defines the one or zero state of the corresponding bit of the fingerprint. Triangles were distinguished by the type of atom at vertexes and by the path length of their edges. The vertex was either donor (D) and planar (Dpl), acceptor (A) and planar (Apl), polar (P), or hydrophobic (H) and planar (Hpl) as defined by the atom-types implemented in MOE.[4] Lengths of the edges were calculated along the molecular graph, no estimation of the 3D structure of molecule was performed at this stage.

Binding studies. Human dopamine D_{2s} and dopamine D_3 receptors were expressed in stably transfected Chinese hamster ovary (CHO) cells.[2,3] In brief, radioligand binding screening was performed on cell membrane preparations in triplicate by using 0.2 nM [^{3}H]spiperone (Amersham Biosciences, Freiburg, Germany). Nonspecific binding was determined in the presence of 10 µM BP897. For rapid screening the compounds have been tested at four concentrations (10 µM, 1 µM, 0.1 µM, and 0.01 µM) in two independent experiments. Competition binding data were analyzed using the software GraphPad Prism™ (San Diego, CA, USA), using a non-linear least squares fit. K_i values were calculated from the IC_{50} values according to the Cheng-Prusoff equation.[4]

(1) Pilla, M., Perachon, S., Sautel, F., Garridol, F., Mann, A., Wermuth, C. G., Schwartz, J. C., Everitt, B. J., Sokoloff, P. *Nature (London)* **1999**, *400*, 371-375.

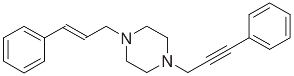
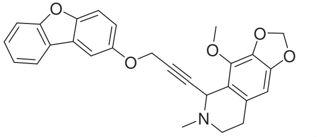
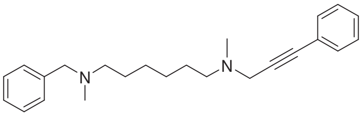
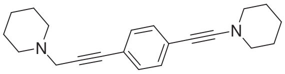
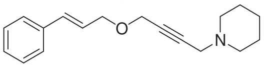
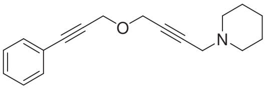
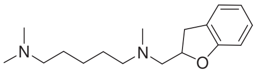
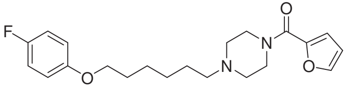
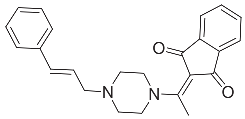
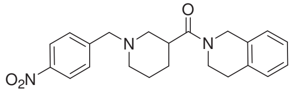
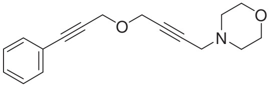
(2) Hayes, G., Biden, T. J., Selbie, L. A., Shine, *J. Mol. Endocrinol.* **1992**, *6*, 920-926.

(3) Sokoloff, P., Andrieux, M., Besancon, R., Pilon, C., Martres, M. P., Giros, B., Schwartz, J. C. *Eur. J. Pharmacol.* **1992**, *225*, 331-337.

(4) Cheng, Y.C., Prusoff, W.H. *Biochem. Pharmacol.* **1973**, *22*, 3099-3108.

2. Results of binding studies of the first screening round

Table S1. Dopamine receptor affinities of compounds from the first virtual screening round (from IBS catalogue)

No.	Chemical Structure	*K*K_i (D_2) [μM][a]	K_i (D_3) [μM][a]
S1		<2	<2
S2		2-6	<2
S3		2-6	2-6
S4		>6	2-6
S5		>6	2-6
S6		>6	2-6
S7		>6	>6
S8		>6	>6
S9		>6	>6
S10		>6	>6
S11		>6	>6

[a] K_i values were measured in CHO cells stably expressing hD_{2s} and hD_3 receptors by using [^{3}H]spiperone (two experiments).

All compounds were aligned according to the basic nitrogen.

3. Docking of ligands into a homology model. To get an idea of a potential binding pose of the found actives, we constructed a homology model of the dopamine D_3 receptor. Docking of the compounds into a homology model of human D_3 receptor highlights two potential hydrogen bonds (Figure S1): one to Ser192, and a second one between the basic amine and Asp110. It was previously shown that mutation of Ser192 to Ala (S192A) leads to approximately ten-fold reduced ligand binding to the mutated D_3 receptor.[1] The importance of optimal hydrogen bonding interaction between the hydroxyl group of the well-studied ligand *R*-(+)-7-OH-DPAT and Ser192 is also supported by SAR data, which shows that replacement of the hydroxyl group by a methoxy group in *R*-(+)-7-OH-DPAT reduces its binding affinity by 100-fold.[2]

A potential hydrogen bonding pattern between the protonated basic amine of the ligands and Asp110 is also visible in the homology model. It was previously demonstrated that the basic amine function is important for receptor-ligand interaction for many different GPCR ligands.[3] For the D_2 and D_3 receptors the interaction with Asp110 (Asp114 for D_2) is a generally accepted hypothesis.[4]

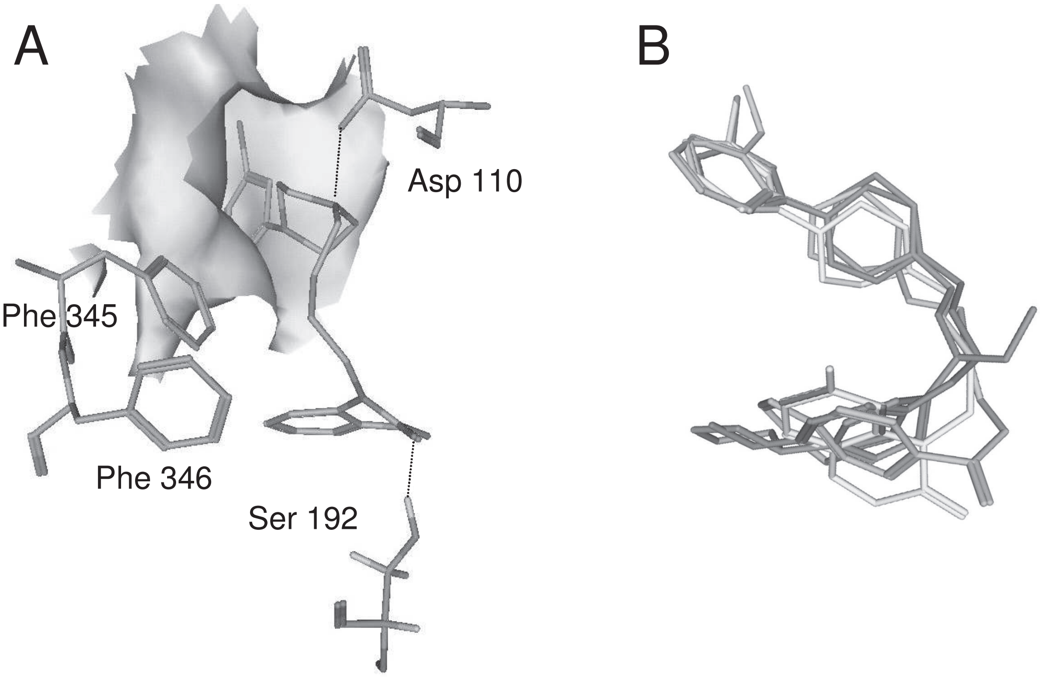


Figure S1. A) Docking of compound **4** into a homology model of the human dopamine D_3 receptor. Parts of the predicted binding pocket are shown. Potential hydrogen bonds between the ligand and Asp110 and Ser192 are indicated by dotted lines. The predicted lipophilic pocket for part C of the molecule (cf. Figure 1) is represented by a Connolly surface. **B)** Superposition of docked structures of compounds **3-6**. The overlapping of the basic amine, as well as part C of the molecules is observed. Potential hydrogen bond acceptors that potentially interact with Asp110 are in close vicinity. Compound **7** is not present in the alignment as it probably adopts a different mode of binding. (Note: compound numbering according to Table 1 and Figure 1 of the main manuscript).

Residues corresponding Phe345 and Phe346 were shown to be important for ligand binding in many different GPCRs.[5] Phe345 of the D_3 receptor corresponds to the Phe389 of D_2; its mutation to Ala was shown to abolish the binding of several ligands.[6] Although we observed these residues to be in contact with the docked ligand in the homology model, we cannot unambiguously identify face-to-face or face-to-edge aromatic interactions between rings of these two residues and the aromatic moiety of our compounds. This is easily explained by the inaccuracy of the constructed homology models due to low sequence identity

(28%) between the D_3 receptor and the rhodopsin template. We wish to stress that although the docking experiments were able to propose a common binding mode for several ligands (Figure S1b), and the model of the binding site is in accordance with receptor mutation studies, the homology model must be treated with great caution. Unarguably, homology models have their value in molecular modeling,[7] but we wish to stress that one should consider our D_3 receptor model only as an "idea generator" potentially guiding the following steps of hit exploration and generation of structure-activity relationships.

Since the compounds selected were taken from public compound collections it is clear that some of the compounds were already used in other investigations e.g., **4** for adrenergic and serotonergic receptors (cf. Ref. 9 in main manuscript).

4. SVM training and active learning. For constructing SVM models we used the *SVM-light* package.[8] Details of the SVM training protocol can be found elsewhere.[9,10] The prediction of a trained SVM is given by Eq. 1.

$$f(\mathbf{x}) = \sum_i \alpha_i K(\mathbf{x}, \mathbf{x}_i^{sv}) + b \text{, where } K(\mathbf{x},\mathbf{y}) = ((\mathbf{x} \bullet \mathbf{y})s + 1)^5 . \quad (1)$$

The greater f the higher is the probability for a compound to be active. **x** and **y** are molecular fingerprint vectors, $\mathbf{x}^{sv}$ are support vectors, i.e. molecular fingerprints that define the exact shape of the separating SVM hyperplane. The kernel function K defines the complexity of the surface that will be constructed. We used a fifth order polynomial kernel for all SVM models. Kernel parameter s was optimized to achieve improved ranking of compounds.[10] For active learning, we considered all compounds with $K_i < 1$ µM for D_2 or D_3 receptors (331 compounds) as active compounds; they were labeled as "Class" (C). 50,000 "Non-Class" (NC) substances were selected from the IBS collection (~240,000). The NC compounds were selected for minimal distance to the SVM hyperplane allowing a more fine-grained re-sampling of the "near-active-compound" space. The resulting filter consisted of two consecutive SVM models: the first SVM model ranked all available IBS compounds with respect to the distance to the active reference set, the second SVM model re-orders the most promising candidates.

SVM training aimed at maximizing the enrichment factor, expressed by the percentage of active compounds retrieved in the top 1% of a ranked screened database. It was done by standard four-fold cross-validation.[11]

5. Regression SVM model for predicting the $\log \frac{K_i^{D3}}{K_i^{D2}}$ ***ratio***. *SVM-light* was used to construct the model. The $<q^2>$ value was used as the criterion for optimization (Eq. 2):

$$< q^2 >= 1 - \frac{\sum_i \left(f^i_{measured} - f^i_{predicted}\right)^2}{\sum_i \left(f^i_{measured} - < f^i_{measured} >\right)^2} . \quad (2)$$

Here $f^{i}_{measured}$ and $f^{i}_{predicted}$ are measured and predicted $\log\frac{K_i^{D3}}{K_i^{D2}}$, where $< f^{i}_{measured} >$ is the averaged ratio of measured binding constants. For model optimization four-fold cross-validation was applied, in which the model was trained on the compounds excluding the validation subset, and then $<q^2>$ was calculated for the compounds of the validation subset. Parameters of the SVM were optimized to achieve maximum performance for the four validation subsets. The final model was trained using all molecules yielding optimized SVM parameters *s* (Eq.1) and *w* epsilon width for the regression tube.[8]

6. Compound ranking based on the similarity to the reference compounds. The selection of the final set of compounds was performed by using a similarity measure, where the distance between molecules was calculated by Equation 3.

$$L(M1, M2) = \sum_i w_i (\mathbf{x}_i^{M1} - \mathbf{x}_i^{M2}) . \qquad (3)$$

Here, L is a distance between molecules *M1* and *M2*; f_i^{M1} and f_i^{M2} are 3PP fingerprint bits, and w_i are the weights of the features estimated by the SVM model. The weights w_i were extracted from the SVM model by estimating their influence on the predicted activity (Eq. 4).

$$w_i = \frac{\sum_{k=1}^{n} \left(f_{pred}(\mathbf{x}^k_{x_i=1}) - f_{pred}(\mathbf{x}^k_{x_i=0}) \right)}{n} . \qquad (4)$$

The summation is over all active compounds (n = 395) of the reference set. The $f_{pred}(\mathbf{x}^k_{x_i=1,0})$ is the prediction of the $\log\frac{K_i^{D3}}{K_i^{D2}}$ for compound $\mathbf{x}^k$, where fingerprint *i* is set to one or zero.

7. Homology model of the D_3 receptor. The transmembrane region of the dopamine D_3 receptor was homology-modeled based on a 2.8 Å resolution rhodopsin crystal structure (PDB-code 1F88).[12] The sequence alignment for the D_3 receptor and rhodopsin was obtained from the multiple sequence and Hidden Markov model of rhodopsin-like GPCRs from the PFAM database.[13] For homology modeling, energy minimization and structural analysis of protein we used MOE software package.[14] The active site of protein was predicted by analyzing the positions of the following residues, that are known to be important for ligand binding to D_3 receptor: Asp110, Ser192, Phe345, Phe346.[5,15] Compounds were docked into the proposed active site using the MOE built-in docking routine. For scoring partial charges from MMFF94s were used.[16] We generated 100 different docked conformations for each compound. A final conformation of each ligand was manually selected, taking into consideration mutation data for the D_3 receptor.[1,2,5] We selected conformations that possess potential hydrogen bonds to Ser192, and between the positively charged amine of the ligands and Asp110.

8. Binding studies. CHO-D_{2S} cells, expressing the recombinant human D_2(short) dopamine receptor gene,[17] were grown in Dulbecco's Modified Medium/Nutrient Mixture F12 1:1 Mixture supplemented with 2 mM glutamine, 10% foetal bovine serum, and 10 µl•ml-1 penicillin/streptomycin in an atmosphere of 5% CO_2 at 37 °C (Gibco^{TM}, Karlsruhe, Germany). Human D_3 receptors stably expressed in CHO cells as previously described by Sokoloff *et al*.[18] were used. The cell line was cultured in Dulbecco`s Modified Eagle Medium supplemented with 2 mM glutamine, and 10% dialysied fetal bovine serum, and were grown in an atmosphere of 5% CO_2 at 37 °C (Gibco^{TM}). Human D_{2S} and D_3 receptors expressing cell lines were grown to confluence. The medium was removed, and the cells were washed with 10 ml PBS buffer (140 mM NaCl, 3 mM KCl, 1.5 mM KH_2PO_4, 8 mM Na_2HPO_4, pH 7.4) at 4 °C. After removing the wash buffer, the cells were scraped from the flasks into 15 ml of ice-cold media, and centrifuged at 3,000 rpm for 10 min at 4 °C. After centrifugation the medium was removed and the supernatant resuspended in ice-cold Tris-HCl buffer containing 5 mM $MgCl_2$, pH 7.4 and disrupted with a Polytron and centrifuged at 20,000 rpm, for 30 min at 4 °C. The pellet was resuspended by sonication in ice-cold Tris-HCl buffer (containing 5 mM $MgCl_2$, pH 7.4), membrane aliquots were stored at -70 °C. Determination of membrane protein was carried out by the method of Bradford.[19] Cell membranes containing human D_2s and D_3 receptors from CHO cells were thawed, rehomogenized with sonication at 4 °C in Tris-HCl, pH 7.4 containing 120 mM NaCl, 5 mM KCl, 2 mM $CaCl_2$ and 1 mM $MgCl_2$ (incubation buffer), and incubated with 0.2 nM [³H]spiperone (106 Ci•mmol $^{-1}$, Amersham Biosciences, Freiburg, Germany), and drug diluted in incubation buffer. Nonspecific binding was determined in the presence of 10 µM BP897 (prepared by same of the authors). Incubations were run at 25 °C for 120 min, and terminated by rapid filtration through PerkinElmer GF/B glass fibre filters (PerkinElmer Life Sciences, Rodgau, Germany) coated in 0.3% polyethylenimine (Sigma-Aldrich, Taufkirchen, Germany) using an Inotech cell harvester (Inotech AG, Dottikon, Switzerland). Unbound radioligand was removed with four washes of 1 ml of ice-cold 50 mM Tris-HCl buffer, pH 7.4, containing 120 mM NaCl. The filters were soaked in 8 ml Beta plate scint scintillator and counted using a PerkinElmer MicroBeta®Trilux scintillation counter (PerkinElmer Life Sciences). Competition binding data were analysed by the software GraphPad Prism™ (2000, version 3.02, San Diego, CA, USA), using non-linear least squares fit. For fast screening the compounds have been tested at four concentrations (10 µM, 1 µM, 0.1 µM, and 0.01µM) in triplicate carrying out two binding experiments for human dopamine D_{2s} and for human dopamine D_3 receptors. K_i values were calculated from the IC_{50} values according to Cheng-Prusoff equation.[20]

9. References

(1) Sartania, N., Strange, P. G. *J. Neurochem.* **1999**, *72*, 2621-2624.

(2) Malmberg, A., Nordvall, G., Johansson, A. M., Mohell, N., Hacksell, U. *Mol. Pharmacol*. **1994**, *46*, 299-312.

(3) Boehm, H. J., Klebe, G., Kubinyi, H. *Wirkstoffdesign*, Spektrum Akademischer Verlag GmbH, Heidelberg, **2002**, p. 355.

(4) Strange, P. G. *Trends Pharmacol. Sci.* **1996**, *17*, 238-244.

(5) Cotte, N., Balestre, M. N., Aumelas, A., Mahe, E., Phalipou, S., Morin, D., Hilbert, M., Manning, M., Durroux, T., Barberis, C., Mouillac, B. *Eur. J. Biochem.* **2000**, *267*, 4253-4263. Chen, S., Xu, M., Lin, F., Lee, D., Riek, P., Graham, R. M. *J. Biol. Chem.* **1999**, *274*, 16320-16330. Noda, K., Saad, Y., Karnik, S. S. *J. Biol. Chem.* **1995**, *270*, 28511-28514. Spadling, T. A., Burstein, E. S., Henderson, S. C., Ducote, K. R., Brann, M. R. *J. Biol. Chem.* **1998**, *273*, 21563-21568. Granas, C., Nordvall, G., Larhammar, D. *J. Recept. Signal. Transduct. Res.* **1998**, *18*, 225-241. Choudhary, M. S., Craigo, S., Roth, B. L. *Mol. Pharmacol.* **1993**, *43*, 755-761.

(6) Cho, W., Taylor, L. P., Mansour, A., Akil, H. *J. Neurochem.* **1995**, *65*, 2105-2115.

(7) Hillisch, A., Pineda, L. F., Hilgenfeld, R. *Drug Discov. Today* **2004**; *9(15),* 659-69.

(8) Joachims T. In *Advances in Kernel Methods - Support Vector Learning;* Schölkopf, B., Burges, C., Smola A., Eds.; MIT-Press: Cambridge, 1999; pp. 41-56.

(9) Byvatov, E., Schneider. G. *Appl. Bioinformatics* **2003**, *2*, 67-77.

(10) Byvatov, E., Schneider, G. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 993-9. Byvatov, E., Fechner, U., Sadowski, J., Schneider, G. *J. Chem. Inf. Comput. Sci.* **2003**, *4*, 1882-1889.

(11) Duda, R. O., Hart, P. E., Stork, D. G. *Pattern Classification,* Wiley-Interscience, New York, **2000**.

(12) Palczewski, K., Kumasaka, T., Hori, T., Behnke, C. A., Motoshima, H., Fox, B. A., Le Trong, I., Teller, D. C., Okada, T., Stenkamp, R. E., Yamamoto, M., Miyano, M. *Science* **2000**, *289,* 739-745.

(13) Bateman, A., Coin, L., Durbin, R., Finn, R. D., Hollich, V., Griffiths-Jones, S., Khanna, A., Marshall, M., Moxon, S., Sonnhammer, E. L., Studholme, D. J., Yeats, C., Eddy, S. R. *Nucleic Acids Research* **2004,** *32*, D138-D141.

(14) MOE (Molecular Operating Environment) version 2004.05. Chemical Computing Group Inc, Montreal. URL: http://www.chemcomp.com

(15) Varady, J., Wu, X., Fang, X., Min, J., Hu, Z., Levant, B., Wang, S. *J. Med. Chem.* **2003**, *46*, 4377-4392.

(16) Halgren, T. A. *J. Comp. Chem.* **1996**, *17*, 490. Halgren, T. A. *J. Comput. Chem.* **1996**, *17*, 490-512, 520-552, 553-586, 587-615, 616-641.

(17) Hayes, G., Biden, T. J., Selbie, L. A., Shine, J. *Mol. Endocrinol*. **1992**, *6*, 920-926.

(18) Sokoloff, P., Andrieux, M., Besancon, R., Pilon, C., Martres, M. P., Giros, B. Schwartz, J. C. *Eur. J. Pharmacol*. **1992**, *225*, 331-337.

(19) Bradford, M. M. *Anal. Biochem*. **1976**, *72*, 248-254

(20) Cheng, Y.C., Prusoff, W.H. *Biochem. Pharmacol.* **1973**, *22*, 3099-3108.

7.6 Virtual Screening Filter to Identify Cytochrome P450 2C9 (CYP2C9) Inhibitors based on SVM for Model Building and Feature Visualization

Byvatov E., Matter H., Baringhaus K.H., Schneider G.

in preparation

A Virtual Screening Filter to Identify Cytochrome P450 2C9 (CYP2C9) Inhibitors based on SVM for Model Building and Feature Visualization

Evgeny Byvatov[1], Karl-Heinz Baringhaus[2], Gisbert Schneider[1], Hans Matter[2,*]

[1] Johann Wolfgang Goethe-Universität
Institut für Organische Chemie und Chemische Biologie
Marie-Curie-Str. 11
D-60439 Frankfurt, Germany

[2] Aventis Pharma Deutschland GmbH,
A company of the sanofi-aventis group
Chemical Sciences, Drug Design, Building G878
D-65926 Frankfurt am Main, Germany

* send all correspondence to:
Dr. Hans Matter
Aventis Pharma Deutschland GmbH,
A company of the sanofi-aventis group
Chemical Sciences, Drug Design, Building G878
D-65926 Frankfurt am Main, Germany
e-mail: hans.matter@aventis.com

Abstract

Cytochrome P450 2C9 (CYP2C9) is one of the most important phase 1 metabolising enzymes in humans for many therapeutically relevant pharmaceuticals. Any new chemical candidate inhibiting this membrane-associated heme protein thus would significantly affect the metabolism of physiologically important molecules and drugs, resulting in clinically significant drug-drug interactions. In search for computational tools to identify potential CYP2C9 inhibitors early in drug discovery, we constructed a filter based on a collection of 1100 structurally diverse molecules tested for CYP2C9 inhibition under identical conditions. The chemical structures were encoded using several 2D descriptors, followed by the generation of different statistical models using support vector machines (SVM). This approach consistently leads to significant and predictive models for regression and classification of CYP2C9 inhibitors. Their predictive ability was underscored by successfully applying them to a test set of 238 compounds. Even more important for early drug discovery phases is the ability of these models to correctly discriminate CYP2C9 inhibitors from inactive molecules on this enzyme. This filter also allows extracting and visualizing important ligand substructures and functional groups, which are essential to understand protein-ligand interactions for CYP2C9. To validate the correct identification of essential functional groups connected to CYP2C9 affinity, predicted features from the SVM models for some local structure-activity series in that dataset were analysed in detail. Furthermore the application of these models to the substrate S-warfarin, which recently has been co-crystallized with CYP2C9, revealed that the identified substructures are involved in the interaction with the CYP2C9 inhibitor binding site. For example, the model correctly indicated the aromatic stacking interactions with Phe114 and Phe476 as well as a hydrogen bond with backbone of Phe100. Hence, these models consistently provide guidelines for reducing CYP2C9 inhibition in novel candidate molecules.

Keywords

Classification / Regression / Cytochrome / CYP2C9 / Prediction / SVM / Virtual screening

1. Introduction

Mammalian cytochrome P450 proteins are a class of membrane-associated heme containing proteins that recognize and metabolize a diverse range of xenobiotics such as environmental molecules, pollutants and drug compounds. The human isoforms CYP1A2, CYP2C9, CYP2C19, CYP2D6 and CYP3A4 were identified as major drug-metabolizing enzymes. Those have been reported to contribute to the oxidative metabolism of ~90 % of currently used pharmaceuticals in clinical application. Cytochrome P450 2C9 (CYP2C9) is one of the important cytochromes involved in drug metabolism in humans, responsible for ~18 % of all reactions catalysed by this superfamily ([1]). There are several CYP2C9 substrates that belong to the class of nonsteroidal anti-inflammatory drugs, such as diclofenac ([2,3]), ibuprofen ([4]), naproxen ([5]), flurbiprofen ([6],[7]), piroxicam ([6]). CYP2C9 is also involved in the metabolism of polar acidic drugs ([8]), progesterone and anticoagulants with a coumarin substructure like S-warfarin. Any interference of novel drug candidates with these known pathways of CYP metabolism thus might cause undesirable drug-drug interactions upon clinical development and co-medication during a therapy.

Until recently, the molecular basis of drug binding to human CYP2C9 and other human cytochromes has been derived from X-ray structures of bacterial CYPs with ligands ([9,10,11,12]). Then the structure of rabbit CYP2C5 provided additional insights of key residues located within the putative ligand and substrate-binding site ([13]). Following these studies, Williams et al. determined the X-ray structure of human CYP2C9 in the absence and presence of the substrate S-warfarin (PDB 1OG2, 1OG5 ([14])). CYP2C9 was identified as a two-domain protein with the typical fold characteristics of the CYP450 family. The heme is located between helices I and L, the iron ion is pentacoordinated with Cys435 as single ligand. This structure reveals unexpected interactions of warfarin to the CYP2C9 binding site and highlights an additional binding area close to the heme pocket.

The structure of a different construct of human CYP2C9 in complex with flurbiprofen has been determined to a resolution of 2.0 Å by X-ray crystallography by Wester et al. (PDB code 1R9O ([7])). Interestingly in this structure a distinct conformation of the helix B to helix C region allows Asp108 to form hydrogen bonds with Asp293 and Asn289 and to interact directly with the carboxylate of flurbiprofen. Obviously these essential interactions are responsible to position the substrate for regioselective oxidation in the CYP2C9 binding site and they account for the preference of this CYP isoform for anionic nonsteroidal anti-inflammatory drugs ([7]). This region adopts a different conformation in the CYP2C9 structure from Williams and does not orient Arg108 towards the substrate binding site ([14]). Several other X-ray crystal structures of important mammalian and human cytochromes have appeared in the meanwhile, which provide additional structural details on important protein-ligand interactions in this protein family ([15,16,17,18,19]).

Any reliable prediction of CYP2C9 inhibition would greatly increase the efficiency in earlier drug discovery phases. There has been a constant development in understanding interaction features in the active site of CYP2C9 using different approaches. Some studies have involved overlapped CYP substrates to identify a binding template ([20,21]). Others have used tienilic acid derivatives ([22]), phenytoin analogs and bis-triazole antifungals ([23]) to establish the structure-activity-relationship (SAR) for rationalizing known substrate and inhibitor specificity of CYP2C9. NMR and molecular modelling have also been combined to assist in defining the positioning of substrates in the CYP2C9 active site ([3]). Site-directed mutagenesis indicated the importance of the I-helix residues Ser286 and Asn289 for specificity for the substrates diclofenac and ibuprofen ([24]) in agreement to interactions revealed in the CYP2C9-flurbiprofen X-ray structure ([7]). This X-ray structure 1R9O also confirmed the observations of a CYP2C9 preference for small acidic lipophilic compounds in an "anionic binding site" ([21,22,25]), while in the 1OG5 binding site no basic amino acids being able to interact with substrates or inhibitors could be identified. In addition numerous hydrophobic residues are lining the CYP2C9 active site from the analysis of both CYP2C9 X-ray structures.

Based on these findings, numerous approaches employing a variety of 3D-QSAR methods have been useful for developing predictive models and understanding binding site requirements for CYP2C9 ([26,27,28,29,30]), although these models are consistently based on small training data sets of diverse molecules.

Recent advances in synthetic methodology have expanded the diversity of chemically accessible structures, leading to an increasing number of high-quality compounds for lead identification and optimization in drug discovery. As the application of filter criteria was shown to increase the quality of newly synthesized candidates ([31]), our interest was to develop a fast and reliable filter for this "antitarget" as complement to existing virtual screening and compound optimization tools focussed on affinity toward the desired molecular target. The filter should be based on internal experimental assay results from a wide range of compounds tested under identical conditions. The purpose of such a filter differs from classical QSAR approaches such that larger numbers of structures are passed through this and related models focussed for target affinity and ADME properties in virtual screening.

In this study we applied support vector machines (SVM) as statistical approach for constructing a structure-activity model for a large and diverse set of compounds tested for CYP2C9 inhibition under identical experimental conditions. This algorithm was invented by Vapnik to derive a classifier for a data set of actives and inactives for a given experimental observation ([32]). It then was successfully applied in different research areas ([33]) including bio- and chemoinformatics ([34]). In some examples, SVM outperformed other learning machines, for instance, ANN (artificial neuronal networks ([35])), while the general utility of any statistical approach depends on many factors like data set, descriptors, size and others. The final quality of any model, however, can only be assessed from application to external test sets. Another important criterion is the ability to understand structural reasons for compound inhibition, which might be useful for subsequent lead optimization.

For the present data set, the use of molecular descriptors requiring information about the inhibitors' three-dimensional (3D) structure and their putative alignment was prevented due to its diversity including many chemotypes from internal drug discovery programs. Consequently two different sets of two-dimensional (2D) descriptors known to capture relevant information for protein-ligand interactions have been employed.

First, three-point pharmacophoric fingerprints (3PP) have been used. Each bit in this fingerprint encodes the presence or absence of a particular pharmacophoric triangle with certain distance requirements in a molecule ([36]). Molecules are represented as collections of pharmacophoric points separated by topological distances (i.e. bonds). This description results in a total of ~10^5 bits per molecule. One advantage of SVM versus other methods is its ability to work with large number of features ([37]). Furthermore there might be several possible binding modes and thus different acceptable pharmacophores for affinity within the CYP2C9 substrate and inhibitor binding pocket. It was not possible in this data set to decide *a priori*, which chemotype is engaged in which orientation within the CYP2C9 binding pocket. SVM was earlier shown to be intrinsically able to deal with larger numbers of descriptors in a vector and to correctly analyse data sets with multiple binding modes ([38]). In addition to standard binary classification, SVM regression models were built to incorporate consistent IC_{50} values for our datasets. These models then allowed visualization of important features for receptor ligand interactions on selected SAR series and compounds from this data set. The most important features for binding of S-warfarin to CYP2C9 could be derived from applying the models, which are in good agreement to its recent X-ray crystal structure in CYP2C9 ([14]).

Subsequently, a collection of different descriptor types was used. This descriptor collection encompasses substructure keys following the MACCS key definitions ([39]), topological pharmacophores based on CATS descriptors ([40]), computed pKa values for basic and acidic groups from ACD/Labs ([41]) and surface based and related descriptors computed using QikProp ([42]). This collection, named CMQA, has been useful for deriving significant statistical models in other internal projects.

These models were further validated by predicting test sets of 238 diverse compounds and a library of compounds focussed towards G-protein coupled receptors (GPCRs), for which experimental data under similar conditions were available. This CYP2C9 prediction model shows sufficient performance and thus can be used early in drug discovery to identify molecules with potential drug-drug interaction problems involving this isoform.

2. Results and Discussion

2.1. Statistical Models using SVM and PLS for CYP2C9 Inhibition

2.1.1. SVM Regression Models using 3PP Descriptors

The resulting models derived using the training set of 1100 compounds followed by validation using 238 compounds as test set are presented in table 1. This table summarizes six models differing by descriptors (3PP, CMQA), statistical methods (SVM, PLS ([43])) and the SVM approach (classification, regression). The original data set, consisting of 1338 molecules from multiple chemical series and internal projects, was divided into a training set of 1100 compounds and a test set of 238 compounds using statistical design applying a maximum diversity approach ([44]).

(Table 1)

First, the SVM regression approach using 3PP fingerprints was applied to build a model for CYP2C9 affinity prediction for the training set of 1100 compounds (table 1a). Internal parameters for the SVM were optimized using four-fold cross-validation (leave-25%-out approach, see Experimental Section). In addition, this approach allows visualization of the inhibitor functional groups that were predicted by SVM to be important for binding to CYP2C9. A significant model with a q^2 value (crossvalidated r^2 from four-fold-crossvalidation) of 0.34 and an r^2 of 0.81 resulted, which correctly predicted the external test set of 238 compounds (predictive r^2 0.63).

After successful prediction of the external set based on the SVM model derived on 1100 compounds, both data sets were merged and analysed again, resulting in a second SVM model for a total of 1338 molecules (table 1b). The q^2 value of 0.43 for this combined training set is larger than the q^2(cv) for 1100 compounds alone due to increased information in this collection. As the original split into training and test sets was done using statistical design, some features are obviously not any longer represented in the training set. However, the final assessment about model quality and usefulness was done based on its application to the test set and the predicted r^2 for this set, which was not used to derive the model (table 1a).

2.1.2. SVM Regression Models using 2D Descriptors

In order to check that relevant information to describe the biological affinity against CYP2C9 was captured by 3PP-fingerprint descriptors, an additional SVM model was generated using a total of 339 CMQA descriptors (table 1c). These descriptors encompass structural keys following the MACCS definition ([39]), CATS pharmacophore correlation over 2D molecular graph ([40]), pKa values from ACD/Labs ([41]) and QikProp derived descriptors ([42]). Due to the smaller number of CMQA descriptors (339 versus ~10^5 3PP fingerprints), a non-linear SVM model performed better on predicting the test set than the corresponding linear SVM model, respectively. In contrast, it was observed for 3PP fingerprints that the linear SVM model resulted in a more predictive model (non-linear SVM results not shown).

This application of non-linear SVM is only meaningful, if the dimensionality of the descriptor space is below a few thousands, like for the CMQA descriptors. Contrastingly for higher dimensionality descriptor spaces, linear SVM model are expected to perform better, as observed for 3PP fingerprints (results not shown). This could partially be attributed to additional noise introduced by non-linear mixing of descriptors relevant and irrelevant to the correct regression ([45]).

When comparing these results versus training a SVM model using 3PP fingerprints (table 1a versus 1c), this regression model exhibited similar quality: the predictive r^2 was only slightly higher with 0.68 in comparison to 0.63 for 3PP fingerprints, respectively. Although the crossvalidated q^2 of the training set derived using four-fold crossvalidation was significantly larger for the non-linear SVM with CMQA descriptors, this can be explained by a different splitting for the cross-validation of the training set in *a* and *c* (Table 1), as this splitting is performed on a random basis prior to each model generation and validation. However, one of the significant advantages of the 3PP fingerprint based SVM model over CMQA is the ability to map fingerprints back to the functional groups of the molecules and thus provide a visual analysis of the relevant features linked to activity. For most CMQA descriptors such a mapping is less obvious.

2.1.3. Comparison to PLS Regression Models

To compare the performance of SVM with other standard regression techniques, another model was built using CMQA descriptors and PLS (*Partial Least Squares*) regression to predict CYP2C9 inhibition (table 1d). All descriptors were autoscaled and columns without variance were rejected for analysis. A 6 component PLS model results with a q^2 value of 0.338 for 1100 compounds (leave-one-out crossvalidation) and a conventional r^2 value of 0.475. When applying this model to the test set of 238 molecules, a predictive r^2 value of 0.55 resulted, which indicates significantly lower predictivity than the predictive r^2 value of 0.68 obtained for the SVM model in table 1c, respectively.

2.1.4. SVM Classification Models

In addition to these regression based SVM models, a SVM classification model was developed as binary filter to estimate whether a compound is able to inhibit CYP2C9. In this case, prior to the SVM training the experimental IC_{50} values were assigned to zero for inactive compounds (IC_{50} >10 µM) and one for actives (IC_{50} < 10 µM). The results of the SVM training are shown in table 1e and f for the training set of 1100 compounds and the combined data set of 1338 molecules, respectively. A total of 85 % of the compounds from the test set with 238 molecules was correctly classified as active or inactive by this SVM classifier from table 1e. For the SVM classifier trained with the combined data set, the overall accuracy could only be monitored using four-fold-crossvalidation. Here, 73% of the compounds in all validation subsets were correctly classified, depending on the initial splitting for internal model validation (table 1f).

2.1.5. Analysis of SVM and PLS Models

Figure 1 shows the graphs of predicted pIC_{50} values (y-axis) versus experimental data on the x-axis for models a, c and d from table 1. Figures 1a, b, c and d indicate the fit of predicted versus experimental pIC_{50} values for linear and non-linear SVM models a and c and both the training set of 1100 compounds (figure 1a, c) and the prediction for the test set of 238 compounds (figure 1b, d). Figures 1e and f provide the same information for PLS model using CMQA descriptors (table 1d). One characteristic feature of SVM training is that SVM does not attempt to adjust the pIC_{50} prediction, if the residual between predicted and experimentally measured pIC_{50} values is less than the tube width parameter of the SVM ([46]). This allows SVM to compensate for inaccuracies in the experimental data for this collection of pIC_{50} values, which might occur especially in the case of high throughput screening data.

The results in Figures 1a and c show that most of the predicted pIC_{50} values are localized within the tube width close to the experimental pIC_{50} value. This reduces the number of outliers for SVM training in comparison to PLS, as seen by comparing figures 1a, c and d. This width parameter is also optimized in crossvalidation by maximizing the q^2 for the randomly chosen validation subsets within the training sets.

(Figure 1)

Most outliers are localized in three characteristic regions of these graphs: inactive compounds, very active compounds (IC_{50} > 0.4 µM, upper right circle) and compounds with IC_{50} values > 10 µM (lower left circle). Outliers with experimental IC_{50} values of 10 µM and higher are related in most cases to insufficient solubility at assay concentrations, as estimated from QikProp solubility predictions ([51]). As a consequence it cannot be excluded that SVM correctly estimates their IC_{50} values, but experimental values are misleading. A similar observation has been made for some compounds with experimental IC_{50} values higher as 2 µM, where solubility was experimentally determined as limiting factor to obtain a more accurate value. Predictions using all models produce consistently lower pIC_{50} values in agreement with this solubility limit. In contrast, some compounds with high experimental affinity to CYP2C9 were predicted only moderately active; on the other hand some of the inactive compounds were predicted to have binding activity for CYP2C9. This might be a consequence of the complexity of inhibitor interactions with CYP2C9. Furthermore the presented model does not explicitly account for entropic effects upon ligand binding. It is also known that some of the CYP2C9 ligands have at least two binding modes ([14,7]). The assumption of a consistent binding site area, which is occupied by this very diverse data set, might also be an oversimplification, as the binding site is relatively large and thus offers several possibilities for protein-ligand binding. Furthermore, the binding of inhibitors to CYP2C9 could additionally be affected by geometrical requirements in the substrate–access channel ([47]). Hence, the experimental binding affinities might be affected by multiple mechanisms and observations in such a diverse data set.

2.1.5. Additional Validation Studies

As additional validation the model 1c (table 1c) was applied to predict compounds from another external dataset published by Afzelius et al. ([29]). For these compounds the binding affinities were reported as K_i instead of IC_{50} values. Moreover, the assay conditions differ to those used in our study ([29]). Hence, in order to compare the SVM filter for this dataset, the compounds were only classified as active (IC_{50}< 10 µM) or inactive (IC_{50}>10 µM) and SVM predictions were compared with known CYP2C9 affinity for this set. In fact, a total of 75 % compounds from this series was correctly classified (table 1c, Accuracy test).

Another more strict validation of the three most relevant models from our study was carried out by an outlier analysis, as presented in table 2. This analysis was performed the original test set with 238 compounds encompassing 95 actives (40 %) and 143 inactives (60 %), plus a second external set consisting of 344 representative members from a GPCR–targeted library on a limited number of scaffolds (see Experimental Section). This second set contains 147 actives (43 %) and 197 inactives (57 %). As mentioned above, the experimental biological affinities for these two sets were obtained under identical conditions. For interpretation, the compounds were classified based on an affinity threshold, namely smaller or larger affinities than an IC_{50} value of 10 μM.

(Table 2)

All models were useful to discriminate compounds active as CYP2C9 inhibitors from inactive compounds, as seen from inspecting table 2. The performance on the GPCR-targeted library compounds in general was slightly worse in comparison to the 238 compound test set. Consistently all models are characterized by a relatively small number of false positives in comparison to false negatives. For model a) using SVM and 3PP fingerprints, 6 and 5 % false positives were found for the test set and the GPCR compounds, while 16 and 24 % false negatives were classified, respectively. Furthermore a predictive r^2 value of 0.362 for the test set derived from representative compounds of the GPCR targeted library was calculated, now taking the actual predicted values into account. These success rates are comparable for model b) using SVM and CMQA descriptors. Here, 6 and 1 % false positives were obtained, while 18 and 28 % false negatives were found for the test and GPCR set, respectively. Here the best predictive r^2 value of 0.45 for the GPCR targeted library compounds were obtained with this non-linear SVM model and CMQA descriptors. Finally the PLS model d) resulted in 3 and 5 % false positives, while 22 and 26 % false negatives were obtained. A predictive r^2 value of 0.412 for representative GPCR compounds was obtained with this model.

Hence, these models are collectively able to identify true CYP2C9 inhibitors with a relatively low rate of wrong classifications (false positives). As this scenario is the primary application of any virtual screening filter for CYP2C9 inhibition complementary to target binding affinities, all these models are useful to rank appropriate compounds during early phases in drug discovery projects. The success rates for identification of inactive compounds are slightly lower, which requires additional experimental testing for those compounds passing the initial filter and shown to be interesting in terms of affinity at the desired target .

2.2. Visualization of Important Functionalities for CYP2C9 Affinity

Subsequently the SVM models were applied to identify ligand features linked to CYP2C9 inhibition. To this end, the influence of every atom to the model was estimated by summing up contributions from 3PP features including this atom as vertex (cf. Experimental Section, Pharmacophore Visualization). The following section provides a chemical interpretation of this model on the basis of important features for selected inhibitors.

Two similar CYP2C9 inhibitors **1** and **2** with different IC_{50} values (4.4 and 30 μM) are displayed in figure 2. The SVM model 1a based on 1100 compounds and 3PP fingerprints (see table 1a) correctly predicts the CYP2C9 binding affinity for **1** (pIC_{50} 2.36 / 2.52 predicted) and **2** (1.52 / 1.78), respectively. Interestingly, the substitution of chlorine against fluorine in **1** is correctly predicted to reduce affinity to CYP2C9. Furthermore, the replacement of the aliphatic side chain in **1** to a compact and less hydrophobic group is also responsible for a lower affinity to CYP2C9. Similar observations are made from feature visualization using two related inhibitors **3** and **4** with significantly higher CYP2C9 binding affinity. For **3** an experimental pIC_{50} value of 3.13 is observed, while 2.56 is predicted; for **4** the experimental pIC_{50} value is 3.00 in comparison to the prediction of 2.23 (Figure 2). Again the SVM model correctly predicts a higher binding affinity of both compounds to CYP2C9; in particular the ranking of **3** and **4** is correctly reproduced. The increased binding affinity in **3** in particular is related to the significant influence of hydrophobic interactions resulting from the biaryl substructure (see Figure 2c).

(Figure 2)

(Figure 3)

However, binding affinities for **1** and **2** were slightly overestimated, but slightly underestimated for **3** and **4**. Thus, the most active inhibitor **3** was docked into the CYP2C9 binding site taken from the PDB structure 1OG5 using flexible docking in QXP ([48]). The most likely binding mode is shown in Figure 3 with the **3** imidazole nitrogen in close contact to the CYP2C9 heme iron, suggesting a pivotal role for protein-ligand recognition in this series. For compounds **1** and **2**, any direct binding interaction to heme is less likely due to additional methyl groups attached to the imidazole, which might result in lower CYP2C9 affinity. However, these additional methyl groups indicating sterically unfavourable regions were not considered during construction of the SVM model: no pharmacophoric point was assigned to these methyl groups according to the PATTY classification scheme ([49]) used to compute the 3PP descriptors. Hence, SVM did not consider any negative influence of those methyl groups, which led to an overestimation of the binding affinity for **1** and **2**, while the lack of this feature results in the underestimation for **3** and **4**. Other inhibitors with important contributions of the imidazole to CYP2C9 binding affinity are displayed in figure 4. Again the aromatic rings of **5-7** are likely to adopt an orientation similar to those compounds shown from Figure 2 in complex with CYP2C9.

(Figure 4)

(Figure 5)

As the active site of the cytochrome CYP2C9 is large and capable to accommodate even several ligands ([14]), this might result in uncertainties in determining binding modes by docking. In the absence of any experimental structure information of a CYP2C9 inhibitor bound to this enzyme, we analysed the X-ray structure of the CYP2C9 substrate S-warfarin (PDB code 1OG5), an anti-coagulant drug with a K_i value of 20µM ([14]). Figure 5a indicates key interaction points of S-warfarin with CYP2C9, namely the aromatic stacking between Phe114, Phe476 and the corresponding aromatic rings of S-warfarin. In addition, two hydrogen bonds between Phe100, Ala103 and the ligand are essential for substrate binding in this region of the binding pocket. 3PP-fingerprints were then calculated based only on the connection graph of the molecule and standard PATTY atomic features (see figure 5c). The sphere diameter close to the atomic features of S-warfarin indicate their relative importance for protein-ligand interactions, estimated as average importance of 3PP triangles that include this feature as a vertex. In fact, three out of four essential interactions are correctly identified (Figure 5c, Table 1b); namely the two aromatic interactions with Phe114 and Phe476 and the hydrogen bond to the Phe100 amide nitrogen.

We have noticed that mainly hydrophobic (H) and hydrophobic planar (H_{pl}, aromatic) features were considered by SVM as important for the interaction with CYP2C9, which is consistent with general assumptions on essential requirements for CYP2C9 inhibition. However, it should be noted that the presented model were derived to identify CYP2C9 inhibitors, while information about substrates cannot be expected, when connected with weak binding affinities. Figure 6 shows two other examples of feature visualization using model 1b for Miconazole (K_i = 6 µM) and the inactive Acebutalol ([30]). Again, mainly hydrophobic features are highlighted by SVM model in good agreement to other reports ([50]).

(Figure 6)

(Figure 7)

Finally the SVM model allows analysing features positively or negatively contributing to CYP2C9 affinity, as exemplified using molecules shown in figure 7. The same SVM model as before was used analysis (Table 1,b). Green and blue spheres highlight the resulting summations separated by positive or negative contribution: green spheres indicate features that are favourable for interaction with CYP2C9 and blue refer to detrimental features. As the same atom can be a part of different 3PP features it is often observed that the same functional group is predicted to confirm positively and negatively to the interaction with CYP2C9. In this case one tendency significantly dominates.

The only difference between **11** and **12** in Figure 7 is the presence of a hydrophobic cyclopropyl group, which is predicted to positively influence CYP2C9 affinity in agreement with experimental data. Again the IC_{50} values are correctly estimated by SVM model (**11**: IC_{50} = 1.0 µM, measured pIC_{50} = 3.0, predicted p IC_{50} = 2.8; **12**: IC_{50} = 30.0 µM, measured pIC_{50} = 2.3, predicted pIC_{50} = 2.4). Features with negative influence to CYP2C9 affinity are less important, as seen from the diameters of corresponding spheres.

When comparing **12** and **13**, the influence of negative features is significantly larger. First, the replacement of the amide by an ester (see arrow in Figure 7.b,c) changes a planar donor to planar polar group, which results in a strong negative feature in this part of the molecule. Adding the hydrophobic methyl group adjacent to the aromatic ring (see arrow Figure 7c) results in another negative feature. Furthermore almost all negative features increase their importance, while almost all positive features decrease their size and importance in **13** in comparison to **12**. This change is also observed for features whose atoms did not change between **12** and **13**. It is attributed to the fact that the negative planar and hydrophobic groups in **13** correspond to a vertex of a negative 3PP feature. As every 3PP feature corresponds to a triangle, the two other vertexes are also negative features and are responsible for the overall increase of negative features in **13**. Hence, this feature visualisation allows for a reasonable chemical interpretation and is in good agreement to the observed ligand SAR trends for CYP2C9 inhibition.

3. Conclusions

The application of SVM as statistical approach allows generating a significant virtual screening filter to identify CYP2C9 inhibitors early in drug discovery. It provides an efficient way to construct significant QSAR models for a large compound set consisting of 1100 structurally diverse molecules tested for CYP2C9 inhibition under identical assay conditions without obvious molecular alignment. Two significant models for classification and affinity prediction of CYP2C9 inhibitors have been obtained; those are collectively useful to select and rank novel compounds during early phases of drug discovery based on their potential to interact with this important cytochrome. Success rates for the identification of inactive compounds are slightly lower compared to those for active compounds. Hence, additional experimental testing is suggested for those compounds, which passed the initial filter, have been synthesized and showed favourable affinity for the desirable biological target.

It is possible to extract and visualize relevant chemical features that are either favourable or detrimental for binding affinity to CYP2C9. This contribution of chemical features to affinity is consistent with known structure-activity trends for some chemotypes and with protein-ligand interactions from the X-ray structure of the substrate S-warfarin with CYP2C9. Hence the constructed models might help to identify compounds with a potential liability early in the drug discovery phase and provide some guidelines to understand structural reasons for their interaction with CYP2C9.

4. Experimental Section

4.1. Data sets and descriptors

For a total of 1338 compounds from different chemotypes and internal drug discovery programs at Aventis, IC_{50} values for CYP2C9 inhibition were determined using a globally harmonized protocol employing human recombinant CYP2C9 with 7-MFC (7-Methoxy-4-trifluoromethyl-coumarin) as substrate. Inhibition constants were obtained in the range between 0.4 µM and 50 µM; higher or lower values were assigned to threshold values. All IC_{50} values were converted to pIC50 values using the relationship $1/(\log(IC_{50})*1000)$. The quality of some data points was limited by low solubility. Solubility was estimated by prediction of aqueous solubility from QikProp (QPlogS ([42,51])). Any interpretation in those cases was consequently done with care.

In general, all molecules were treated as neutral. Counter ions and salts were removed. Canonical 3D structures required for global QikProp descriptors were generated using the program Corina ([52,53]). This data set, which encompasses multiple chemical series from internal projects, was divided into a training set of 1100 representative structures and 238 molecules as test set using statistical design. To this end, 2D fingerprints were computed using the program UNITY ([54]) and used for compound selection using a *maximum dissimilarity* approach ([44,55,56]).

For constructing a classification models, the activity of the molecules was mapped to 0 and 1 depending on the experimental IC_{50} value. Compound with an IC_{50} value of 10 µM or higher were marked inactive, compounds with lower IC50 values were marked active otherwise.

Every compound was represented by a fingerprint of three-point pharmacophores (3PP), derived from their 2D structure using the standard MOE fingerprint implementation ([36]) or by a collection of substructure, topological pharmacophore and physicochemical descriptors (see below). The individual 3PP fingerprint is a triangle of atoms with specific pharmacophore properties assigned to them ([57]). We consider all possible triangles with their vertexes located at the atom centres of a molecule. Presence or absence of a certain triangle defines the “1” (i.e., bit is set) or “0” state of the corresponding bit in the fingerprint. We distinguished triangles by the type of atom at vertexes and by the length of their edges. The vertex can be either donor (D), acceptor (A), polar (P), donor and planar (D_{pl}), acceptor and planar (A_{pl}), hydrophobic (H) and hydrophobic and planar (H_{pl}) as defined by the atom-type implemented in the software suite MOE, version 2004.03. Lengths of the edges were calculated along the molecular graph, so no estimation of the 3D structure of molecule was performed.

The second descriptor collection (CMQA) encompasses 163 substructure keys implemented in Sybyl/SLN following the MACCS key definitions ([39]), 150 topological pharmacophore descriptors implemented in Sybyl based on the CATS pharmacophore correlation over 2D molecular graph ([40]), computed pKa values for basic and acidic groups derived from ACD/Labs ([41]) and a collection of surface based descriptors computed using QikProp ([42]). This collection was useful in other projects for deriving significant statistical models for multiple properties. Those descriptors were analysed using PLS (*partial least squares*) ([43]), as implemented in Sybyl, and SVM (see below). PLS allows deriving a linear relationship for highly underdetermined matrices. Again, crossvalidation ([58]) is used to check for consistency and predictivity of the resulting models.

To additionally validate the resulting statistical models, another dataset analysed by Afzelius et al. was used ([29,30]). For this collection of compounds with 34 active and 49 inactive molecules, K_i values for CYP2C9 inhibition have been measured using a CYP2C9 assay at AstraZeneca ([29]). However, we classified those compounds as active or inactive to compare these results with our model based on a different biological assay. Another internal dataset comprised 344 representative samples from a focussed library for GPCR targets on a limited number of chemotypes. IC_{50} values were determined using the same protocol as for the main reference set of 1100 training and 238 test set compounds.

Flexible docking of selected compounds into the CYP2C9 binding site from the PDB file 1OG5 was done using the Monte-Carlo based *mcdock* algorithm implemented in QXP version 2.0 ([48]) starting from different input poses to achieve convergence to a most likely binding pose. Selected protein side chains at the inhibitor binding site were treated as flexible based on the comparative analysis of other public CYP2C9 X-ray structures.

4.2. Support Vector Machine

Two different algorithms for regression and classification are in general available for training of the support vector machine. In both cases SVM constructs a surface in the *n*-dimensional space that in case of classification separates active from inactive compounds and in case of regression predicts the pIC_{50} value ([37]). Here *n* is the number of parameters that were used to describe a molecule. Prior to construction of the hyperplane the data is mapped to a very high-dimensional space, where this surface is found in a form of a plane. This plane is then mapped backed to the original space. The result of the SVM training for both classification and regression can be given by the following equation (Eq. 1).

$$f(\mathbf{x}) = \sum_i \alpha_i K(\mathbf{x}, \mathbf{x}_i^{sv}) + b \text{, where } K(\mathbf{x}, \mathbf{y}) \text{ is a kernel function} \qquad (1)$$

For building a classification model, *f(***x***)* gives prediction of the value related to the probability of the molecule to be active, i.e. the greater the value of *f*, the higher is the probability for this molecule to be active. For the SVM regression case, *f(x)* corresponds to the pIC_{50} value to be predicted. The sum in equation (1) runs over the employed support vectors as part of the training set. **x** and **y** represent vectors of molecular features, $\mathbf{x}^{sv}$ are support vectors, i.e. vectors that define the exact shape of the separating hyperplane. Parameters a_i and *b* were determined during SVM training as described ([35]). For constructing SVM models, the SVM-light package was used ([37]). The kernel function *K* defines the complexity of the surface that will be constructed. Different standard kernels can be used during SVM training ([46]).
We used the following kernel functions:

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \bullet \mathbf{y}) \text{, for 3PP fingerprints} \qquad (2)$$

While a large number of 10^5 different 3PP fingerprints were used to describe individual molecules, only a small portion of them was relevant for deriving the final models. It was unlikely that the use of a non-linear kernel in this case would significantly improve prediction results. In contrast, the use of a non-linear kernel function resulted in significantly lower prediction quality (results not shown), most likely due to the introduction of noise by considering non-linear dependency of relevant and irrelevant features together. For the final models with CMQA descriptors condensing structural and physicochemical information, we used a fifth-order polynomial kernel for SVM instead:

$$K(\mathbf{x},\mathbf{y}) = ((\mathbf{x} \bullet \mathbf{y})s + 1)^5 \qquad (3)$$

This fifth order polynomial kernel is known to construct a surface to predict sufficiently complex non-linear dependencies. ([32]) Kernel parameter *s* was optimized to achieve better ranking of compounds as described. ([35])

4.3. Pharmacophore Visualization

Potential pharmacophore models of the inhibitors were visualized by showing to which extent certain atom is important for favourable or unfavourable protein-ligand interactions. The importance R_i of each 3PP feature was calculated based on the change of the SVM prediction for a molecule with this feature removed (Eq. 4).

$$R_i = f(\mathbf{x}(F_i = 1)) - f(\mathbf{x}(F_i = 0))\,. \qquad (4)$$

Here **x** is a fingerprint representation of a molecule with presence or absence of feature F_i. Then, each atom contributing to feature F_i received the weight R_i. The importance of every atom was estimated as an averaged sum of the 3PP features, which included this atom as one of the vertexes.

When considering separately negatively and positively contributing features, summation was done independently for corresponding contributing 3PP features. It allows independent visualization of features favourable and unfavourable for the interaction with CYP2C9.

5. Acknowledgment

We gratefully acknowledge the discussions with T. Klabunde on the GPCR-targeted library. Furthermore we would like to thank C. Giegerich and A. Kugelstadt for computational support and A. Dudda and G.U. Kürzel for providing experimental data for this project.

Figure Legends

Figure 1. Graph of predicted versus experimental pIC_{50} values for three statistical models with splitting into training and test subsets. a) and b) graphs for training and test subsets for linear SVM model based on 3PP fingerprints. c) and d) same graphs for non-linear SVM with 5-order polynomial kernel and CMQA descriptors. e) and f) PLS models with CMQA descriptors and similar splitting into training and test subsets.

Figure 2. Interpretation of the chemical features predicted by SVM to be important for interaction with CYP2C9. The diameter of a sphere near the feature indicates its relative importance for the final model (green favourable to interaction, blue unfavourable). Larger spheres indicate features with higher importance. a) IC_{50} = 4.4 µM, pIC_{50} = 2.36, predicted pIC_{50} = 2.52. b) less active as **1**; IC_{50} = 30 µM, pIC_{50} = 1.52, predicted pIC_{50} = 1.78. Arrows mark chemical groups that according to SVM contribute to the different affinity of these compounds to CYP2C9. The replacement of chlorine by fluorine at the aromatic ring was predicted to reduce CYP2C9 affinity. The same effect is observed, when a bulkier group replaces the lipophilic substituent of compound **1**. c) IC_{50} = 0.74 µM, pIC_{50} = 3.13, predicted pIC_{50} = 2.56. d) IC_{50} = 1.0 µM, pIC_{50} = 3.00, predicted pIC_{50} = 3.13.

Figure 3. Docking of the compound **3** (IC50 0.74 µM) into the active site of CYP2C9. The pivotal role of the imidazole ring of this compound is obvious due to the interaction with the heme.

Figure 4. Related compounds similar to **1**-**4**. For these compounds the imidazole ring is also crucial for the interaction with CYP2C9. Their aromatic rings can adopt similar conformations as assumed for **1**-**4** in complex with CYP2C9. a) IC_{50} = 0.74 µM, pIC_{50} = 3.13, predicted pIC_{50} = 2.56. b) IC_{50} = 0.74 µM, pIC_{50} = 3.13, predicted pIC_{50} = 2.56. c) IC_{50} = 0.74 µM, pIC_{50} = 3.13, predicted pIC_{50} = 2.56.

Figure 5. X-ray crystal structure of S-warfarin in complex with CYP2C9 in comparison to features contributing to the final SVM model. a) Protein-ligand interactions between the binding site of CYP2C9 and S-warfarin from the X-ray crystal structure (PDB code 1OG5) b) Chemical formula of S-warfarin. c) Indication of the relative importance of S-warfarin features to CYP2C9 interaction. Larger spheres indicate features with higher relative importance.

Figure 6. Molecular features important for inhibition of CYP2C9 for miconazole (a) (K_i = 6 µM) and acebutalol (b) (inactive), as estimated from analysis of the SVM model.

Figure 7. Molecular features important for inhibition of CYP2C9. Features favourable for CYP2C9 inhibition are indicated by green spheres, features detrimental for affinity are indicated as blue spheres. The size of the sphere indicates its relative importance. a) Compound **11** IC_{50} = 1.0 µM, pIC_{50} = 3.0, predicted pIC_{50} = 2.8. b) Compound **12** IC_{50} = 5.0 µM, pIC_{50} = 2.3, predicted pIC_{50} = 2.4. c) Compound **13** IC_{50} = 30.0 µM, pIC_{50} = 1.52, predicted pIC_{50} = 1.68.

Table 1. Evaluation of different statistical models and descriptors for predicting CYP2C9 inhibition.

	Method	Descriptors	# cpds training	#cpds test	r^2	$q^2(cv)$[e]	r^2_{pred}[f]	Accuracy training	Accuracy test
a	SVM Linear Regression[a]	3PP fingp.[c]	1100	238	0.81	**0.34**	**0.63**	-	-
b	SVM Linear Regression	3PP FPs[c]	1338	-	0.78	**0.43**	**-**	-	-
c	SVM Non linear Regression[a]	339 CMQA descr.[d]	1100	238	0.89	**0.48**	**0.68**	-	75 % for Afzelius Dataset[g]
d	PLS Linear Regression[a]	339 CMQA descr.[d]	1100	238	0.475	**0.338**	**0.55**	-	-
e	SVM Classification Linear model[a,b]	3PP FPs[c]	1100	238	-	**-**	**-**	93.9 %	**85 %**
f	SVM Classification Linear model[b]	3PP FPs[c]	1338	-	-	**-**	**-**	86 %	**73 %**[h]

[a] Training and test set selection from initial 1338 compounds done using maximum dissimilarity strategy based on 2D fingerprints and statistical design. [b] Classification model based on the following experimental thresholds: actives IC_{50} < 2 µM, inactives by IC_{50} > 10 µM [c] MOE three point pharmacophore (3PP) fingerprints [d] MACCS, CATS, pKa and QikProp. [e] q^2 was calculated for four validation subsets from the training set (leave-25%-out) [f] r^2_{pred} was calculated for the test set of 238 compounds [g] Computed for the dataset reported by Afzelius et al.; all model predictions were used to classify compounds as either active (IC_{50} < 10 µM) or inactive (IC_{50} > 10 µM) [h] Computed for four crossvalidation subsets.

Figure 1.

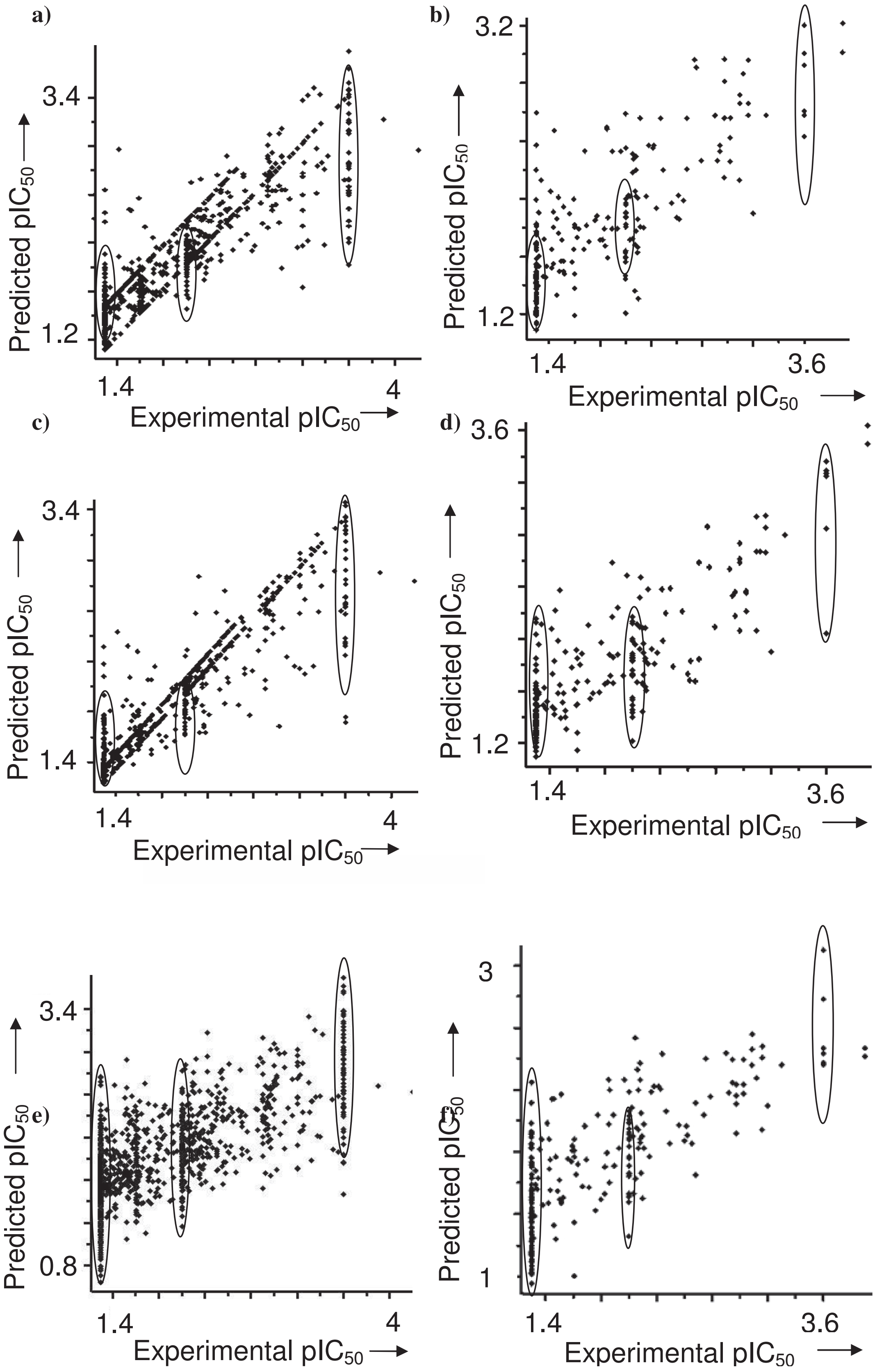
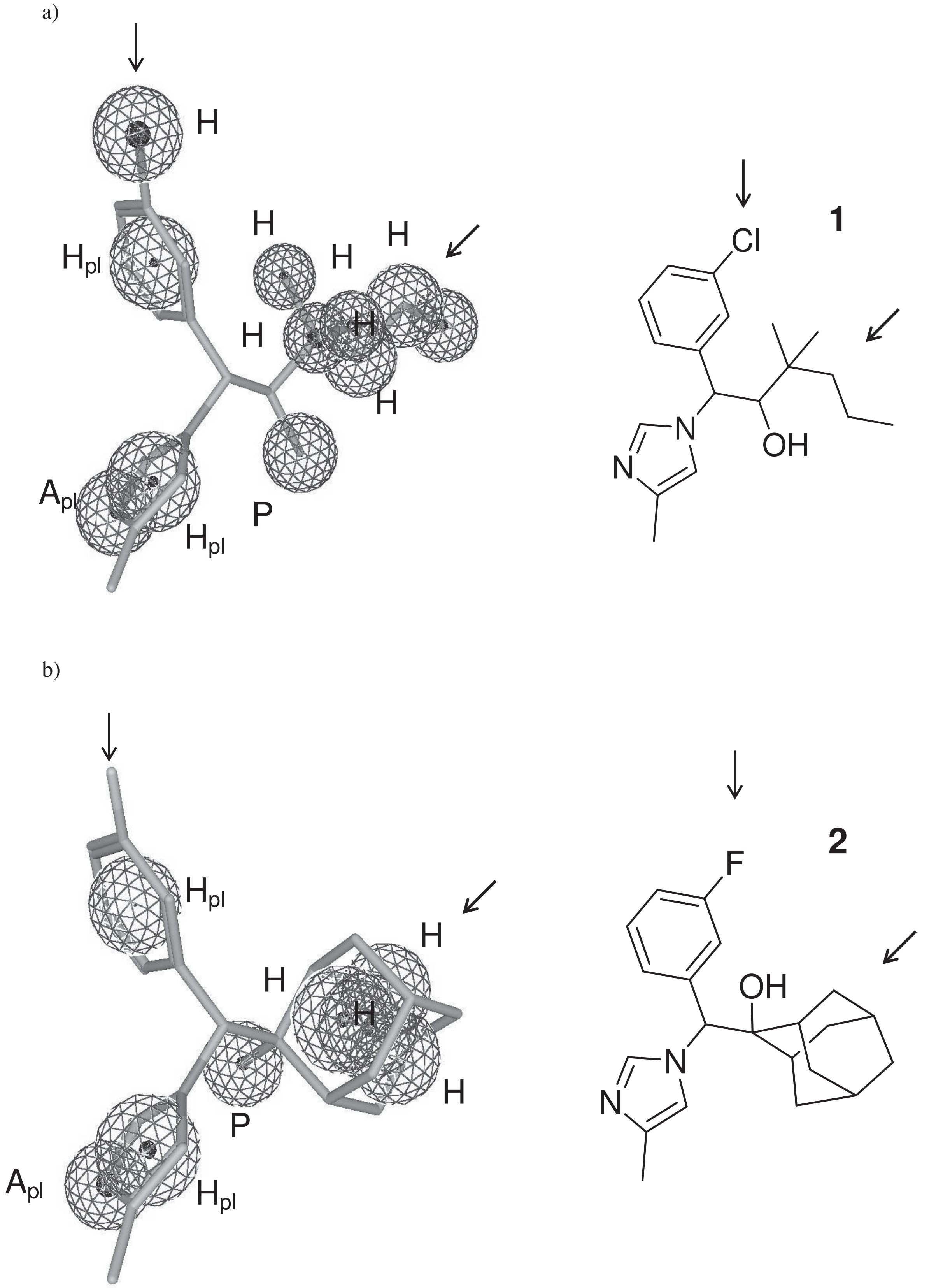

a)
3.4
1.2
Predicted pIC50 →
1.4
4
Experimental pIC50 →
b)
3.2
1.2
Predicted pIC50 →
1.4
3.6
Experimental pIC50 →
c)
3.4
1.4
Predicted pIC50 →
1.4
4
Experimental pIC50 →
d)
3.6
1.2
Predicted pIC50 →
1.4
3.6
Experimental pIC50 →
e)
3.4
0.8
Predicted pIC50 →
1.4
4
Experimental pIC50 →
f)
3
1
Predicted pIC50 →
1.4
3.6
Experimental pIC50 →

Table 2. Performance of individual statistical models on outliers. Number of false positive and false negatives for test data set (238 compounds) and GPCR-targeted library. Prior to analysis the all compounds were mapped to active (IC_{50} < 10 µM) or inactive (IC_{50} > 10 µM) based on measured IC_{50} values.

Model	Test 238 false pos. [d]	Test 238 false neg. [d]	GPCR false pos. [e]	GPCR false neg. [e]	GPCR fitted $<q^2>$[f]
SVM 3PP (linear regr.)[a]	14 (6 %)	38 (16 %)	18 (5 %)	84 (24 %)	0.362
SVM CMQA (nonlin. regr.)[b]	14 (6 %)	44 (18 %)	2 (1 %)	98 (28 %)	0.452
PLS CMQA[c]	7 (3 %)	52 (22 %)	18 (5 %)	91 (26 %)	0.412

[a], [b] and [c] models correspond to models a, c and d from Table 1. [d] This set is the same set (238 compounds) that was used as a test set for models a, c and d from Table 1. [e] 334 compounds from GPCR–targeted library, IC_{50} for them were measured at slightly different conditions as for the Aventis dataset of 1338 compounds that we used most of the time. [f] The predicted pIC_{50} for GPCR –targeted library from all three models was linearly fit to the measured pIC_{50} to compensate for the assay differences in measuring IC_{50} for GPCR-targeted library and 1100 training compounds of these three models.

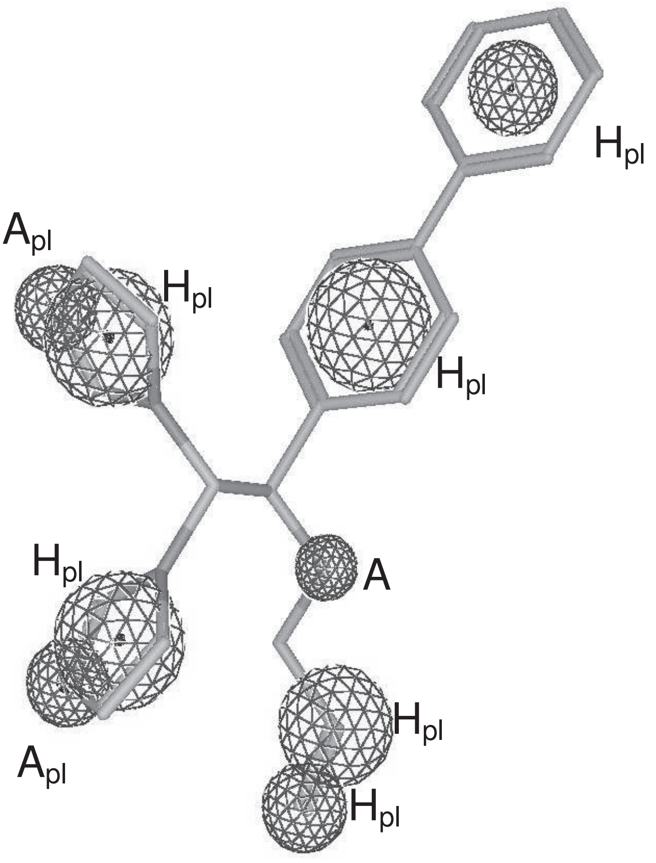
Figure 2.



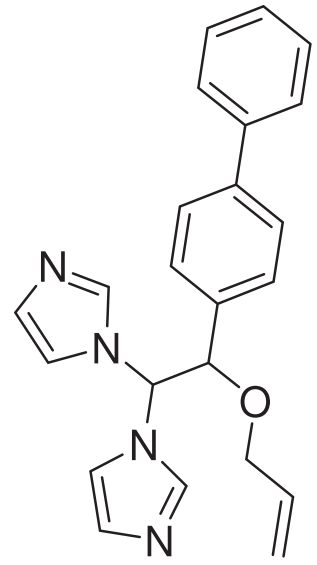
a)

b)

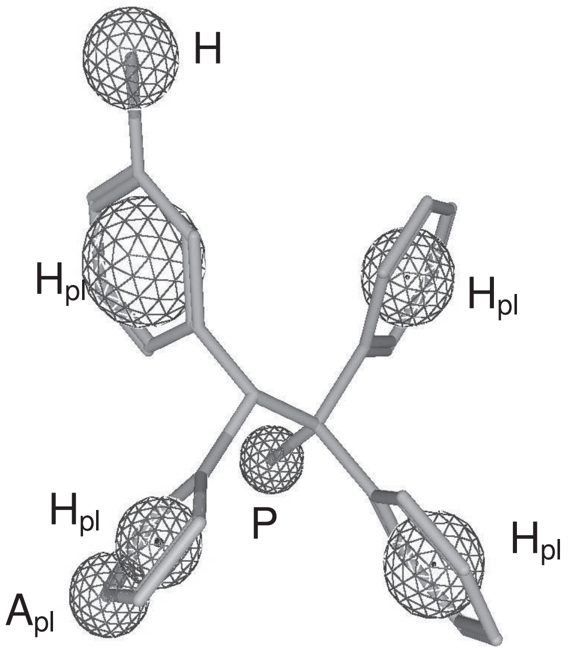
c)


H_{pl}
A_{pl}
H_{pl}
H_{pl}
H_{pl}
A
A_{pl}
H_{pl}
H_{pl}

3


N
N
N
N
O

d)


H
H_{pl}
H_{pl}
H_{pl}
P
H_{pl}
A_{pl}

4

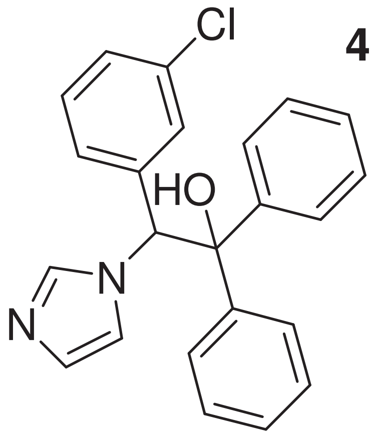

Cl
HO
N
N

Figure 3

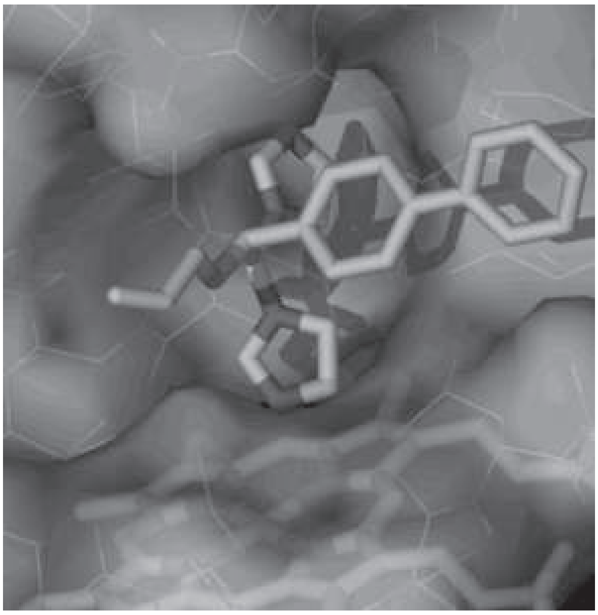
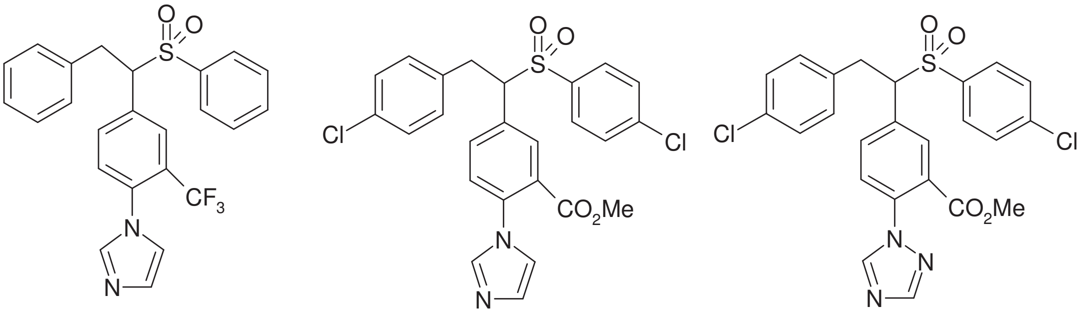


Figure 4



5 **6** **7**

Figure 5

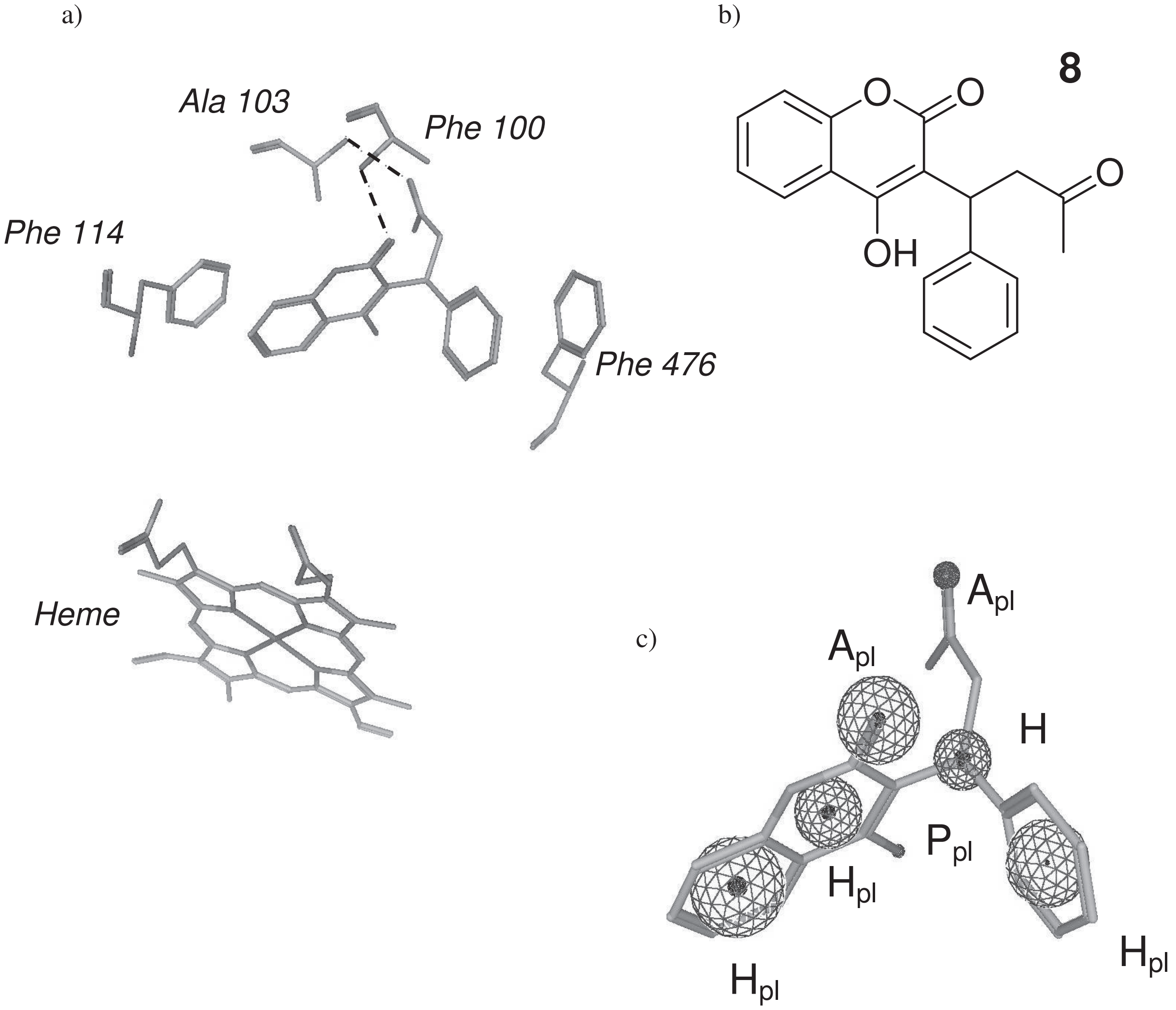
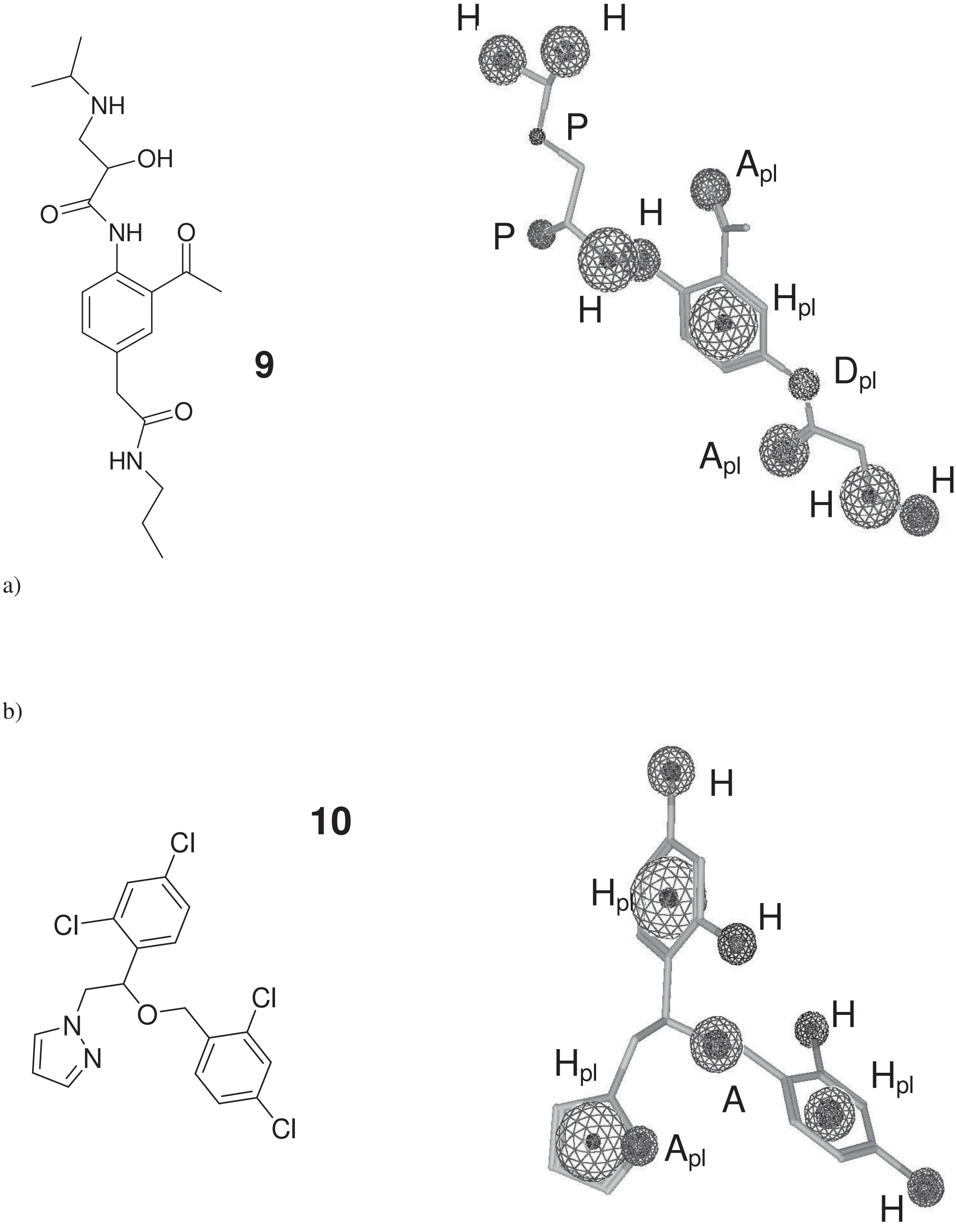

a)
Ala 103
Phe 100
Phe 114
Phe 476
Heme
b)
8
O
O
O
OH
c)
A_pl
A_pl
H
P_pl
H_pl
H_pl
H_pl

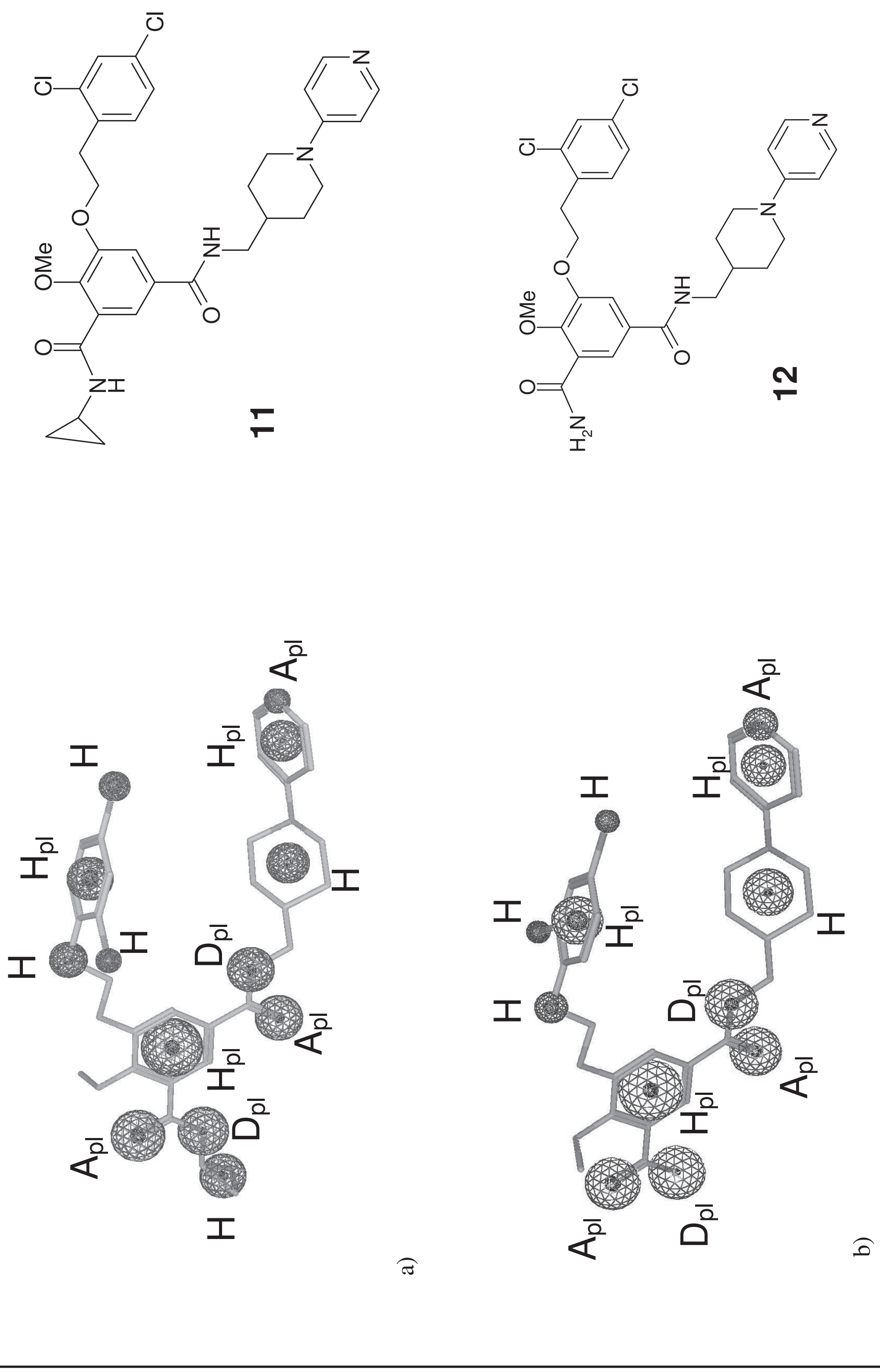
Figure 6


NH
OH
O
NH
O
9
O
HN
H
H
P
Apl
P
H
H
Hpl
Dpl
Apl
H
H
10
Cl
Cl
N
N
O
Cl
Cl
H
Hpl
H
H
Hpl
Hpl
A
Apl
H

a)

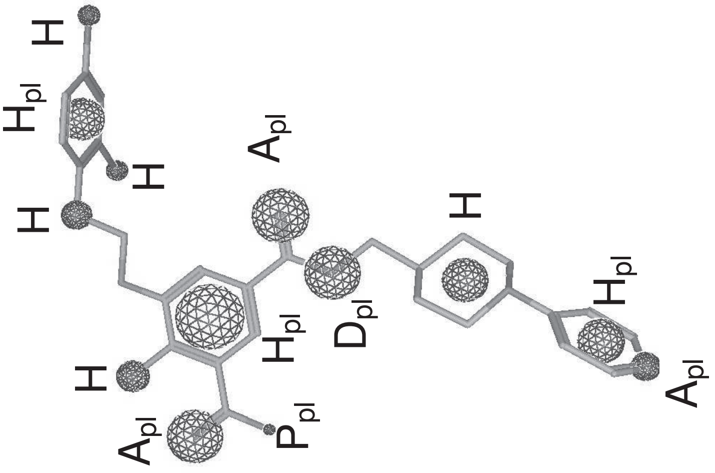
b)

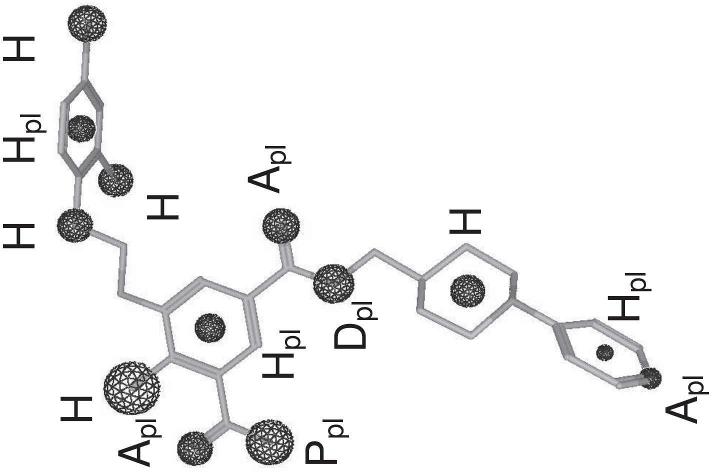
Figure 7.

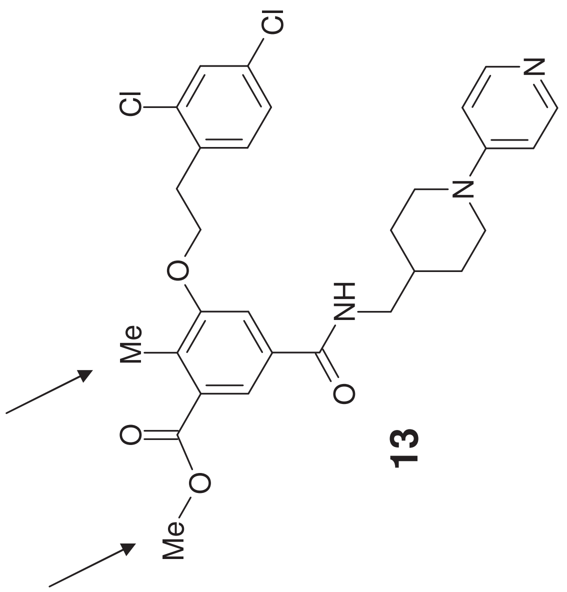

H
A$_{pl}$
D$_{pl}$
H$_{pl}$
A$_{pl}$
H
H
H$_{pl}$
H
D$_{pl}$
H
H$_{pl}$
A$_{pl}$
a)
A$_{pl}$
D$_{pl}$
H$_{pl}$
A$_{pl}$
H
H
H
H$_{pl}$
D$_{pl}$
H
H$_{pl}$
A$_{pl}$
b)
OMe
O
O
NH
N
H
Cl
Cl
N
N
11
H$_2$N
O
OMe
O
O
NH
Cl
Cl
N
N
12

31

c)


H
H_pl
H
H
A_pl
H
H_pl
D_pl
H_pl
A_pl
H
A_pl
P_pl


H
H_pl
H
H
A_pl
H
H_pl
D_pl
H_pl
A_pl
H
A_pl
P_pl


Cl
Cl
O
Me
NH
O
O
O
Me
N
N

13

References

[1] Smith, D.A.; Ackland, M.J.; Jones, B.C. Properties of cytochrome P450 isoenzymes and their substrates. Part 1: Active site characteristics. *Drug Disc. Today* **1997**, *2*, 406-414.

[2] Transon, C.; Leemann, T.; Vogt, N.; Dayer P. In vivo inhibition profile of cytochrome P450tb (CYP2C9) by (()-fluvastatin. *Clin. Pharmacol. Ther*. **1995**, 58, 412-417.

[3] Poli-Scaife, S.; Attias, R.; Dansette, P.M.; Mansuy, D. The substrate binding site of human liver cytochrome P4502C9: An NMR study. *Biochemistry* **1997**, *36*, 12672 12682.

[4] Hamman, M. A.; Thompson, G. A.; Hall, S. D. Regioselective and stereoselective metabolism of ibuprofen by human cytochrome P450 2C. *Biochem. Pharm.* 1997, 54, 33-41.

[5] Miners, J. O.; Coulter, S.; Tukey, R. H.; Veronese, M. E.; Birkett, D. J. Cytochromes P450, 1A2, and 2C9 are responsible for the human hepatic O-demethylation of R- and S-naproxen. *Biochem. Pharmacol.* 1996, 51, 1003-1008.

[6] Tracy, T. S.; Marra, C.; Wrighton, S. A.; Gonzalez, F. J.; Korzekwa, K. R. Studies of 4'-hydroxylation. Additional evidence suggesting the sole involvement of cytochrome P450 2C9. *Biochem. Pharmacol.* 1996, 52, 1305-1309.

[7] Wester, M.R.; Yano, J.K.; Schoch, G.A.; Yang, C.; Griffin, K.J.; Stout, C.D.; Johnson, E.F. The structure of human cytochrome P450 2C9 complexed with flurbiprofen at 2.0-Å resolution. *J. Biol. Chem.* **2004**, *279,* 35630-35637.

[8] Hall, S.D.; Hamman, M.A.; Rettie A.E.; Wienkers, L.C.; Trager W.F.; VandenBranden M and Wrighton S.A. Relationships between the levels of cytochrome P4502C9 and its prototypic catalytic activities in human liver microsomes. *Drug Metab. Dispos.* **1994**, *22*, 975-977.

[9] Li, H.; Poulos, T.L. Conformational dynamics in cytochrome P450-substrate interactions. *Biochimie* **1996**, *78*, 695-699.

[10] Li, H.; Poulos, T.L. The structure of the cytochrome P450 BM-3 haem domain complexed with the fatty acid substrate, palmitoleic acid. *Nature Struct. Biol.* **1997**, *4*, 140-146.

[11] Schlichting, I.; Jung, C.; Schulze H. Crystal structure of cytochrome P-450cam complexed with the (1S)-camphor enantiomer. *FEBS Lett.* **1997**, *415*, 253-257.

[12] Podust, L.M.; Poulos, T.L.; Waterman, M.R. Crystal structure of cytochrome P450 14α-sterol demethylase (CYP51) from Mycobacterium tuberculosis in complex with azole inhibitors. *Proc. Natl. Acad Sci. U.S.A.* **2001**, *98*, 3068-3073.

[13] Williams, P.A.; Cosme, J.; Sridhar, V.; Johnson, E.F.; McRee, D.E. Mammalian microsomal cytochrome P450 monooxygenase: structural adaptations for membrane binding and functional diversity. *Mol. Cell.* **2000**, *5*, 121-131.

[14] Williams, P.A.; Cosme, J.; Ward, A.; Angove, H.C.; Vinkovic, D.M.; Jhoti, H. Crystal structure of human cytochrome P450 2C9 with bound warfarin. *Nature* **2003**, *424*, 464-468.

[15] Scott, E.E.; He, Y.A.; Wester. M.R.; White, M.A.; Chin, C.C.; Halpert, J.R.; Johnson, E.F.; Stout, C.D. An open conformation of mammalian cytochrome P450 2B4 at 1.6-Å resolution. *Proc. Natl. Acad. Sci. U.S.A.* **2003**, *100*, 13196-13201.

[16] Scott, E.E.; White, M.A.; He, Y.A.; Johnson, E.F.; Stout, C.D.; Halpert, J.R. Structure of Mammalian Cytochrome P450 2B4 Complexed with 4-(4-Chlorophenyl)imidazole at 1.9-Å resolution. *J. Biol. Chem.* **2004**, *279*, 27294-27301.

[17] Schoch, G.A.; Yano, J.K.; Wester, M.R.; Griffin, K.J.; Stout, C.D.; Johnson, E.F. Structure of Human Microsomal Cytochrome P450 2C8. *J. Biol. Chem.* **2004**, *279*, 9497-9503.

[18] Yano, J.K.; Wester, M.R.; Schoch, G.A.; Griffin, K.J.; Stout, C.D.; Johnson, E.F. The Structure of Human Microsomal Cytochrome P450 3A4 Determined by X-ray Crystallography to 2.05-Å Resolution. *J. Biol. Chem.* **2004**, *279*, 38091-38094.

[19] Williams, P.A.; Cosme, J.; Vinkovic, D.M.; Ward, A.; Angove, H.C.; Day, P.J.; Vonrhein, C.; Tickle, I.J.; Jhoti, H. Crystal Structures of Human Cytochrome P450 3A4 Bound to Metyrapone and Progesterone. *Science* **2004**, *305*, 683-686.

[20] Jones, B.C.; Hawksworth, G.; Horne, V.A.; Newlands, A.; Tute, M.; Smith, D.A. Putative active site model for CYP2C9 (tolbutamide hydroxylase). *Br. J. Clin. Pharmacol.* **1993**, *34*, 143-144.

[21] Jones, B.C.; Hawksworth, G.; Horne, V.A.; Newlands, A.; Morsman, J.; Tute, M.S.; Smith, D. A. Putative active site template model for cytochrome P450 2C9 (tolbutamide hydroxylase). *Drug Metab. Dispos.* **1996**, *24*, 260-266.

[22] Mancy, A.; Broto, P.; Dijols, S.; Dansette, P.M.; Mansuy, D. The substrate binding site of human liver cytochrome P450 2C9: an approach using designed tienilic acid derivatives and molecular modelling. *Biochemistry* **1995**, *34*, 10365-10375.

[23] Morsman J.M.; Smith D.A.; Jones B.C. and Hawksworth G.M. (1995) Role of hydrogen-bonding in substrate structure-activity relationships for CYP2C9. In *Proceedings of the 4th International ISSX Meeting*; Seattle, Washington, USA, 1995; pp. 259-261.

[24] Klose, T.S.; Ibeanu, G.C.; Ghanayem, B.I.; Pedersen, L.G.; Li, L.; Hall, S.D.; Goldstein, J.A. Identification of residues 286 and 289 as critical for conferring substrate specificity of human CYP2C9 for diclofenac and ibuprofen. *Arch. Biochem. Biophys.* **1998**, *357*, 240-248.

[25] De Groot, M.J.; Alex, A.A.; Jones, B.C. Development of a combined protein and pharmacophore model for cytochrome P450 2C9. *J. Med. Chem.* **2002**, *45*, 1983-1993.

[26] Jones, J.P.; He, M.X.; Trager, W.F.; Rettie, A.E.; Three-Dimensional Quantitative Structure-Activity Relationship For Inhibitors of Cytochrome P450 2C9. *Drug Metab. Dispos.* **1996**, *24*, 1-6.

[27] Rao S.; Aoyama R.; Schrag, M.; Trager, W.F.; Rettie, A.; Jones, J.P. A Refined 3-Dimensional QSAR of Cytochrome P450 2C9: Computational Predictions of Drug Interactions. *J. Med. Chem.* **2000**, *43*, 2789-2796.

[28] Ekins, S.; Bravi, G.; Binkley, S.; Gillespie, J.S.; Ring, B.J.; Wikel, J.H.; Wrighton, S.A. Three and four dimensional-quantitative structure-activity relationship (3D/4D-QSAR) analyses of CYP2C9 inhibitors. *Drug Metab. Dispos.* **2000**, *28*, 994-1002.

[29] Afzelius, L.; Masimirembwa, C.M.; Karlén, A.; Andersson T.B.; Afzelius, I.; Discriminant and quantitative PLS analysis of competitive CYP2C9 inhibitors versus non-inhibitors using alignment independent GRIND descriptors. *J. Comp. Aided Mol. Des.* **2002**, *16*, 443-458.

[30] Afzelius, L.; Afzelius, I.; Masimirembwa, C.M.; Karlén, A.; Andersson T.B.; Mecucci, S.; Baroni, M.; Cruciani, G. Conformer- and Alignment-Independent Model for Predicting Structurally Diverse Competitive CYP2C9 Inhibitors. *J. Med. Chem.* **2004**, *47*, 907-914.

[31] Zuegge, J.; Fechner, U.; Roche, O.; Parrott, N.J.; Engkvist, O.; Schneider, G. A fast virtual screening filter for cytochrome P450 3A4 inhibition liability of compound libraries. *Quant. Struct.-Act. Relat.* **2002**, *21*, 249-256.

[32] Cortes C, Vapnik V. 1995. Support-Vector Networks. Machine Learning, 20:273-297.

[33] Cristianini N, Shawe-Taylor J. 2000. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge: Cambridge University Press.

[34] Byvatov E., Schneider G. *SVM applications in bioinformatics.* Appl. Bioinformatics. 2003; 2(2):67-77.

[35] Byvatov, E.; Fechner, U.; Sadowski, J.; Schneider, G. Comparison of Support Vector Machine and Artificial Neural Network Systems for Drug/Nondrug Classification. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 1882-1889.

[36] MOE 2004.03, CCG, 1010 Sherbrooke St. West, Suite 910, Montreal, Quebec, H3A 2R7, Canada.

[37] T. Joachims, Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999

[38] Byvatov, E.; Schneider, G. Support Vector Machine based Feature Selection for Characterization of Focused Compound Collections. *J. Chem. Inf. Comput. Sci.* **2004,** *44*, 993-999.

[39] Internal implemented in Sybyl / SLN (Version 6.9, Tripos Inc., 1699 S. Hanley Road, St. Louis, MO 63144, USA.) following the documentation in ISIS/Base 2.1.3., Molecular Design Ltd, 14600 Catalina Street, San Leandro, CA 94577, USA.

[40] a) Schneider, G.; Neidhart, W.; Giller, T.; Schmid, G. „Scaffold-hopping“ by topological pharmacophore search: a contribution to virtual screening. *Angew. Chem. Int. Ed. Engl.* **1999**, *38*, 2894-2896.
b) Internal implementation in Sybyl (Version 6.9, Tripos Inc., 1699 S. Hanley Road, St. Louis, MO 63144, USA.)

[41] ACD/logD Suite, ACD/Labs, 33 Richmond St. West, Suite 605, Toronta, ON MSH 2L3, Canada.

[42] a) Duffy, E.M.; Jorgensen, W.L. Prediction of Properties from Simulations: Free Energies of Solvation in Hexadecane, Octanol, and Water. *J. Am. Chem. Soc.* **2000**, *122*, 2878-2888.
b) QikProp Version 2.0, Schrödinger, Inc. 1500 S.W. First Avenue, Suite 1180, Portland OR 97201, USA.

[43] a) Wold, S.; Albano, C.; Dunn, W.J.; Edlund, U.; Esbenson, K.; Geladi, P.; Hellberg, S.; Lindberg, W.; Sjöström, M. In *Chemometrics: Mathematics and Statistics in Chemistry*; Kowalski, B., Ed., Reidel, Dortrecht, The Netherlands, 1984, pp 17-95.
b) Dunn, W.J.; Wold, S.; Edlund, U.; Hellberg, S.; Gasteiger, J. Multivariate Structure-Activity Relationship Between Data from a Battery of Biological Tests and an Ensemble of Structure Descriptors: The PLS Method. *Quant. Struct.-Act. Relat.* **1984**, *3*, 31-137.
c) Geladi, P. Notes on the History and Nature of Partial Least Squares (PLS) Modelling. *J. Chemom.* **1988**, *2*, 231-246.

[44] Pötter, T.; Matter, H. Random or Rational Design ? Evaluation of Diverse Compound Subsets from Chemical Structure Databases. *J. Med. Chem.* **1998**, *41*, 478-488.

[45] R. O. Duda, P. E. Hart, D. G. Stork, Pattern Classification, Wiley-Interscience, New York, 2000.

[46] Burges CJC. 1998. A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery, 2:121-167.

[47] Lewis, D.F.V. On the recognition of mammalian microsomal cytochrome P450 substrates and their characteristics: towards the prediction of human P450 substrate specificity and metabolism. *Biochem. Pharmacol.* **2000**, *60*, 293-306.

[48] McMartin, C.; Bohacek, R.S. QXP: powerful, rapid computer algorithms for structure-based drug design. *J. Comput.-Aided Mol. Des.* **1997**, *11*, 333-344.

[49] Bush, B.L.; Sheridan, R.P. PATTY: A Programmable Atom Typer and Language for Automatic Classification of Atoms in Molecular Databases. *J. Chem. Info. Comp. Sci.* **1993**, *33*, 756-762.

[50] Lewis, D.F.V. Quantitative structure activity relationships (QSARs) for substrates of human cytochromes P450 CYP2 family enzymes. *Toxicology in Vitro* **2004**, *18*, 89-97.

[51] Jorgensen, W.L.; Duffy, E.M. Prediction of drug solubility from Monte Carlo simulations. *Bioorg. Med. Chem. Lett.* **2000**, *10*, 1155-1158.

[52] Sadowski, J.; Rudolph, C.; Gasteiger, J. The generation of 3D models of host-guest complexes. *Anal Chim Acta* **1992**, *265*, 233-241.

[53] Sadowski, J.; Gasteiger, J.; Klebe, G. Comparison of automatic three-dimensional model builders using 639 X-ray structures. *J. Chem. Inf. Comp. Sci.* **1994**, *34*, 1000-1008.
[54] UNITY Version 4.4, Tripos Inc., 1699 S. Hanley Road, St. Louis, MO 63144, USA.
[55] Lajiness, M.; Johnson, M.A.; Maggiora, G.M. Implementing Drug Screening Programs using Molecular Similarity Methods. In: *QSAR: Quantitative Structure-Activity Relationships in Drug Design*; Fauchere, J.L., Ed.; Alan R. Liss Inc.: New York, USA, 1989; pp 173-176.
[56] Taylor, R. Simulation Analysis of Experimental Design Strategies for Screening Random Compounds as Potential New Drugs and Agrochemicals. *J. Chem. Inf. Comput. Sci*. **1995**, *35*, 59-67.
[57] Sheridan, R.P., Miller, M.D., Underwood, D.J., Kearsley, S.K. Chemical Similarity Using Geometric Atom Pair Descriptors. *J. Chem. Inf. Comput. Sci.* **1996**, 128-136.
[58] a) Wold, S. Cross-Validatory Estimation of the Number of Component in Factor and Principal Component Models. *Technometrics* **1978**, *4,* 397-405.
b) Diaconis, P.; Efron, B. Computer-Intensive Methods for Statistics. *Sci. Am.* **1984**,*116*, 96-117.
c) Cramer, R.D.; Bunce, J.D.; Patterson, D.E. Crossvalidation, Bootstrapping and Partial Least Squares Compared with Multiple Regression in Conventional QSAR Studies. *Quant.-Struct.-Act. Relat.* **1988**, *7*, 18-25.

7.7 Extraction and visualization of pharmacophore models by SVM

Byvatov E., Franke L., Werz O., Steinhilber D., Schneider G.
J Med Chem *submitted*

Extraction and visualization of pharmacophore models using Support-Vector-Machines

Evgeny Byvatov[1], Lutz Franke[1], Oliver Werz[2], Dieter Steinhilber[2], Gisbert Schneider[1,*]

[1] Johann Wolfgang Goethe-Universität
Institut für Organische Chemie und Chemische Biologie
Marie-Curie-Str. 11
D-60439 Frankfurt, Germany

[2] Johann Wolfgang Goethe-Universität
Institut für Pharmazeutische Chemie
Marie-Curie-Str. 9
D-60439 Frankfurt, Germany

* send all correspondence to:
Prof. Dr. Gisbert Schneider, Institut für Organische Chemie und Chemische Biologie, Marie-Curie-Str. 11, D-60439 Frankfurt, Germany
Tel: +49 (0)69 79829821
Fax: +49 (0)69 79829826
Email: gisbert.schneider@modlab.de

SUBMITTED FOR THE JANSSEN MEMORIAL ISSUE

Abstract

In this article we constructed pharmacophore models for cyclooxygenase 2 (COX-2) and thrombin inhibitors. These models resulted from Support Vector Machine (SVM) training. They were further evaluated by estimating enrichment factors obtained from virtual screening of a database containing ~2.7×10^6 commercially available compounds. 50-90% percent of the known active compounds were listed within the first 0.1% of the ranked database. It is shown that different binding modes, interchangeability of functionally equivalent features and the possibility to fuse features from different ligands are represented within the models. For this study the molecules were encoded by topological three-point pharmacophores. In order to check the validity of the constructed SVM models we developed a method for feature extraction and visualization using SVM. As a result, features were weighted according to their importance for COX-2 and thrombin inhibition. Well known thrombin and COX-2 pharmacophore points were recognized by the machine learning system. Different binding modes for thrombin were correctly predicted by SVM. Finally several prospective COX-2 inhibitors were tested *in vitro* and shown to be active. Their docked structures with the visualized SVM pharmacophore confirm the identification of relevant features.

Key words

Classification / Enrichment factor / Thrombin / Cyclooxygenase / Prediction / SVM / Virtual screening

Introduction

A pharmacophore represents the 3D or 2D arrangements of structural or chemical features of a drug (small organic compounds, peptides, etc.) that may be essential for interacting with the receptor for optimum binding.[1] These pharmacophores can be used in different ways in drug design programs: i) as a query tool in virtual screening to identify potential new compounds from 3D databases of "drug-like" molecules with patentable structures different from those already discovered; ii) to predict the activities of a set of new compounds yet to be synthesized; iii) to understand the possible mechanism of action; (4) to extract potential privileged structures.[2,3] Currently several algorithms are known that construct pharmacophore models from a set of available active compounds employing potential pharmacophore points (PPP), e.g., CATALYST,[4] DISCO,[5] GASP,[6] or field-based approaches.[7,8] The performance of these methods usually relies on the quality of the initial three-dimensional (3D) alignment of compounds. This influence of the initial alignment on the quality of the resulting pharmacophore model can be modulated by considering multiple ligand conformations and applying "fuzzy" pharmacophore point definitions.[9,10] Here we present a complementary method for pharmacophore identification that is grounded on the topological three-point pharmacophore (3PP) concept.[10] The general idea was to avoid a strict dependency of the pharmacophore model on a 3D alignment. Each molecule was represented as a binary vector, where each feature corresponds to the presence or absence of a particular pharmacophore triangle.[11] Such feature vectors were used for construction of a classifier predicting molecules to have certain biological activity. Subsequent visualization of the features with respect to their contribution to the model allowed us to create reasonable pharmacophore models. Support Vector Machines (SVM) were used for both classification and feature extraction.[12-14] For the present study compounds were represented by fingerprints that contained ~10^4 potential 3PP triangles, and we expected SVM to efficiently discriminate between important and unimportant features and construct reliable and chemically interpretable pharmacophores. This approach was motivated by the fact that SVM classifiers have been shown to be well-suited for first-pass virtual screnning purposes.[15-17] Two test cases were selected to evaluate our new approach, namely the development of SVM classifiers for cyclooxygenase 2 (COX-2) and thrombin inhibitors.

Data and Methods

Data sets and features

Two subsets of the COBRA database were used for SVM training: thrombin and COX-2 inhibitors.[18] We used these subsets as a reference for ranking ~2.7 million compounds that are commercially available from different companies. Every compound was represented by a fingerprint of 3PP pharmacophores using MOE.[19] The individual 3PP pharmacophore is a triangle. We consider all possible triangles with their vertexes located at the atom centers of a molecule. Presence or absence of a certain triangle defines one or zero state of the corresponding bit in the fingerprint. We distinguished triangles by the type of atom at vertexes and by the length of their edges. The vertex can be either donor (D), acceptor (A), polar (P), donor and planar (D=), acceptor and planar (A=), hydrophobic (H) and hydrophobic and planar (H=) as defined by the atom-type implemented in the software suite MOE, version 2004.05.[19] Lengths of the edges were calculated along the molecular graph, so no estimation of the 3D structure of molecule was performed.

Support Vector Machine

The SVM constructs a surface in the *n*-dimensional space that separates active from inactive compounds.[20] Here *n* is the number of 3PP that were used to describe a molecule. Prior to construction of the separating surface the data is mapped to a very high-dimensional space, where separating surface is found in a form of a hyperplane. This hyperplane is then mapped backed to the original space.[21] The result of the SVM training can be given by the following equation (Eq. 1).

$$f(\mathbf{x}) = \sum_i \alpha_i K(\mathbf{x}, \mathbf{x}_i^{sv}) + b \text{ , where } K(\mathbf{x}, \mathbf{y}) = ((\mathbf{x} \bullet \mathbf{y})s + 1)^5 \qquad (1)$$

Here, *f*(**x**) gives the prediction of the molecule to be active, i.e. the greater the value of *f* the higher is the predicted probability to be active. **x** and **y** are molecular fingerprint vectors, $\mathbf{x}^{sv}$ are support vectors, i.e. molecular fingerprints that define the exact shape of the separating hyperplane. The kernel function *K* defines the complexity of the surface that will be constructed. Different standard kernels can be used during SVM training.[20] We used a fifth order polynomial for all SVM models. Kernel parameter *s* was optimized to achieve better ranking of compounds as described.[15,16]

For database screening, we sorted all available compounds with respect to predicted *f*. The sum in Equation (1) is over support vectors, they are part of the raining set. As can be seen the ranking function depends only on the support vectors. Parameters a_i and *b* were determined during SVM training as described.[22] For constructing SVM models we used the SVM-light package.[23]

Training of the SVM and "Active Learning" Optimization

During SVM training we tried to optimize the percentage of active compounds found within the top 0.1% percent of the screening database. In order to achieve this we used a standard four-fold cross-validation procedure.[24] The reference dataset of active compounds was divided into four parts. Each part in turn was mixed with the ~2.7×10^6 screening compounds. The rest of the set of known actives and the newly created set of molecules to be screened were used as training sets and were assigned "class" (known inhibitors) and "nonclass" (all other molecules) labels for SVM training. After the training the nonclass compounds were sorted with respect to the *f* value computed by the SVM classifier. Molecules with higher *f* value are expected to be similar to the active compounds. The parameters of SVM were optimized to achieve a maximum number of "mixed" (i.e. reference molecules that were added to the pool of screening compounds) active molecules within 0.1% of the ranked data. We should note that during SVM training "mixed" active compounds were marked as nonclass molecules, so that validation subsets were not used in training.

The SVM parameters were tuned to have maximum average accuracy of prediction over all four validation sets. Then the final SVM training was performed using all available actives ("class") and all ~2.7×10^6 compounds with unknown activity as a "nonclass". The resulting ranking of these compounds by the trained SVM was used to cherry-pick molecules for *in vitro* activity testing.

It is computationally very expensive to train an SVM with all available ~2.7×10^6 compounds. To overcome this limitation, SVM training was performed in two steps: first an SVM was trained with a randomly selected subset containing only 10^5 compounds from the screening database. In this case the region near the active compounds might be insufficiently sampled. We therefore used the "active learning" approach to focus on this "relevant" area of descriptor space.[24] After obtaining the first ranked list of the compounds the SVM training procedure was repeated, now with a sample set consisting of the top-ranking ~10^5 compounds. By this a more fine-tuned SVM classifier was obtained.

Pharmacophore Visualization

Potential pharmacophore models of the inhibitors were visualized by highlighting atoms that contribute to the most important features. The importance R_i of each 3PP feature was calculated based on the change of SVM prediction for a molecule when this feature is removed (Eq. 2).

$$R_i = f(\mathbf{x}(F_i = 1)) - f(\mathbf{x}(F_i = 0)) \, . \qquad (2)$$

Here **x** is a fingerprint representation of a molecule with presence or absence of feature F_i. Then, each atom contributing to feature F_i received the weight R_i. However, it is reasonable to assume that the importance of atoms in a 3PP differs. In order to take this into account the importance of every atom in the reference set of actives was estimated (Figure 1). The individual weight w of an atom was estimated as the average weight of all 3PP triangles that contain this atom as a vertex. Averaging was done twice, i) over the triangles of each molecule (Figure 1d), and finally ii) over the whole set of actives.

Visualization contrast was enhanced by diminishing the weights **w** of the atoms in every 3PP except for the most important one by w^n. We choose n = 10 empirically, as it produces the best visualization of the core pharmacophore, i.e. the weight of the most important atom remains equals 1 and all other weights diminished.

[Figure 1]

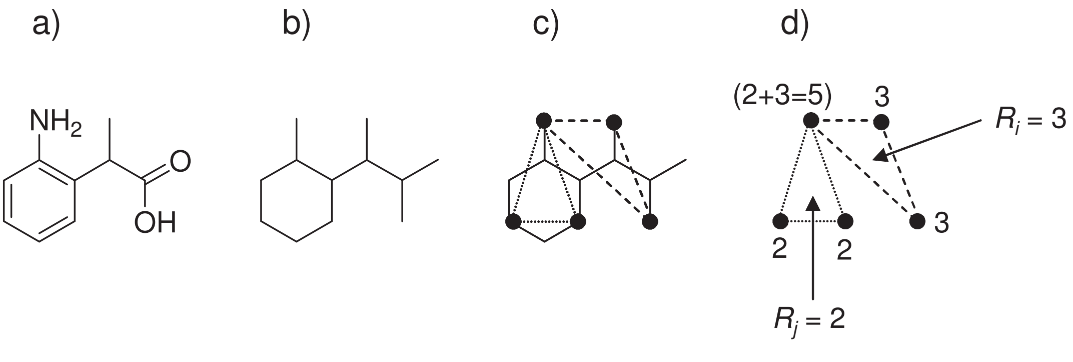
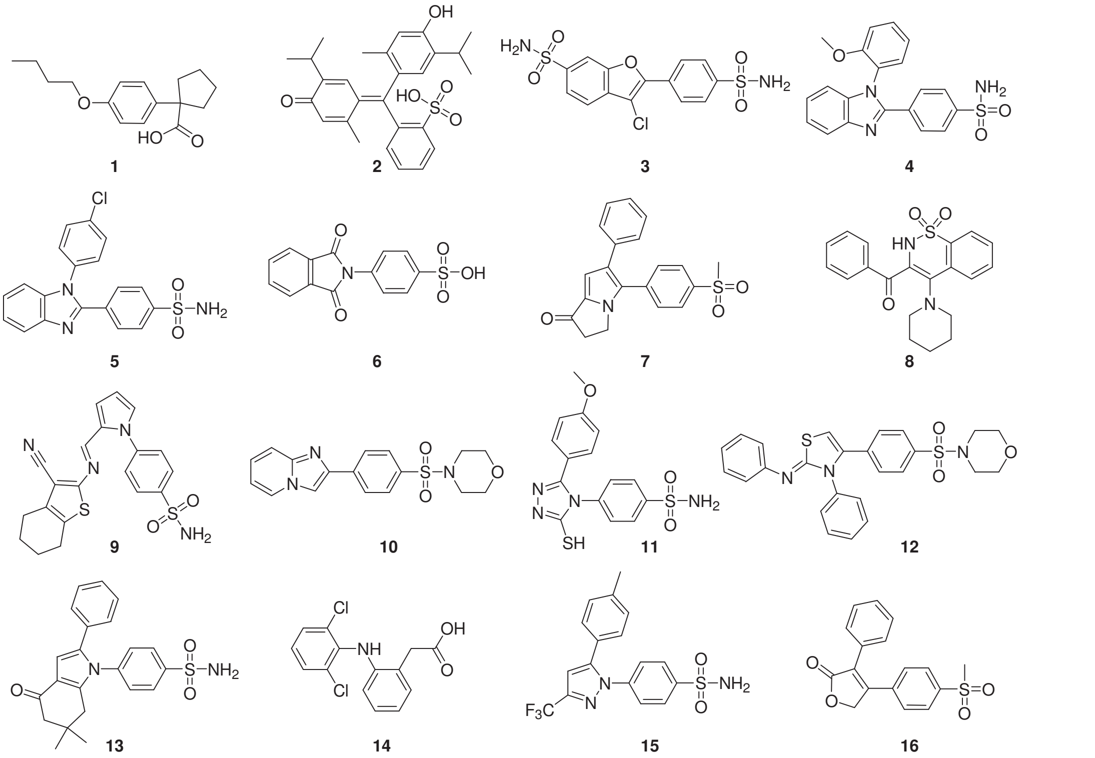


Figure 1. Calculation of atom weights for feature visualization. The two-dimesnional molecular structure (a) is converted to the molecular graph representation (b). Then topological 3PP triangles are assigned (the length of each edge is calculated as the number of bonds in the molecular graph connecting the two vertices along the shortest path) (c), and the importance R_i of each triangle is determined by Equation 2. Individual atoms are weighted proportional to the sum of the R_i values of contributing 3PP features (d).

Selection and Testing of Potential COX-2 Inhibitors

From the SVM-ranked list of the commercially available compounds 13 molecules were cherry-picked for *in vitro* activity testing. We excluded compounds that contain reactive groups and potentially insolvable molecules by visual inspection. Only compounds available from Specs (Delft, The Netherlands; www.specs.net) were considered (Scheme 1). A COX-2 inhibition assay was performed to evaluate compound activity with diclofenac **14** as a positive control:[26] MM6 cells were grown with or without transforming growth factor beta (TGFß) and calcitriol for 96 hrs as described.[27] Six hrs prior harvest, lipopolysaccharide (100 ng/ml) was added. Cells were harvested, washed twice, re-suspended in PGC buffer (phosphate buffered saline at pH 7.4 containing 1 mg/ml glucose and 1 mM $CaCl_2$) (5×10^6 cells/ml) and incubated with arachidonic acid (30 µM) for 15 min at 37°C. Cells were centrifuged (300 × g, 5 min, 4°C) and the amount of 6-keto $PGF_{1\alpha}$ released was assessed by ELISA using a monoclonal antibody against 6-keto $PGF_{1\alpha}$ according to the protocol described by Yamamoto and coworkers.[28,29] For the ELISA, the monoclonal antibody (0.2 µg/200 µl) was coated to microtiter plates via a goat anti-mouse-IgG antibody. 6-keto PGF1α (15 µg) was linked to bacterial β-galactosidase (0.5 mg, Calbiochem), and the enzyme activity bound to the antibody was determined in an ELISA reader at OD550 nm (reference wavelength: 630 nm) using chlorophenol-red-β-D-galactopyranoside (CPRG, Roche Diagnostics GmbH) as substrate.



[Scheme 1]

Scheme 1. Compounds **1-13** were cherry picked by virtual screening and tested for COX-2 inhibition. Reference compounds diclofenac **14**, celecoxib **15**, rofecoxib **16**.

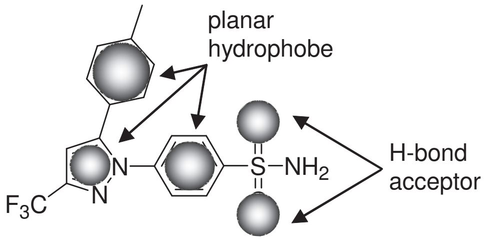
Docking of COX2 inhibitors

For docking of compounds into the COX-2 active site cavity MOE software was used.[19] The complex of COX-2 with a selective inhibitor Sc-558 (PDB-identifier: 1CX2) served as reference. Only one of the four identical domains of the COX-2 complex was considered. Prior to docking hydrogen atoms were added to the protein complex and its structure was energy minimized keeping positions of all atoms fixed except for the added hydrogens. Partial charges of the atoms were calculated using MMFF estimation.[30] Docking was performed using MOE molecular dynamics approximation and Tabu search as described.[31] The results were evaluated by comparison with the binding mode of the reference inhibitor SC-558.

Results and Discussion

SVM classifiers were trained to predict thrombin and COX-2 inhibitors. The accuracy of the predictions was assessed by retrospective database screening. In the case of the COX-2 classifier, 81 ± 6 % of the reference compounds were retrieved within the first 0.1% of the ranked database in a cross-validation study. The retrieval of thrombin ligands was less accurate, yielding 55 ± 14 % of the reference compounds from the first 0.45% of the ranked database. The small standard deviations in both cases indicate robust prediction models. With further optimization by active learning we achieve the rate, 81 ± 6 % of the reference compounds in 0.0031% of the ranked database for COX2 inhibitors and 55 ± 14 % of the reference compounds from the first 0.083% of the ranked database for Thrombin ligands. This difference in performance might be explained not only by differences of the two reference sets and SVM classifier shortcomings, but also by the structural diversity of chemotype-families that are present in the screening database.[32] Overall, we concluded that the two SVM classifiers might be useful for generating focused libraries with significant enrichment of actives compared to a random selection of compounds.

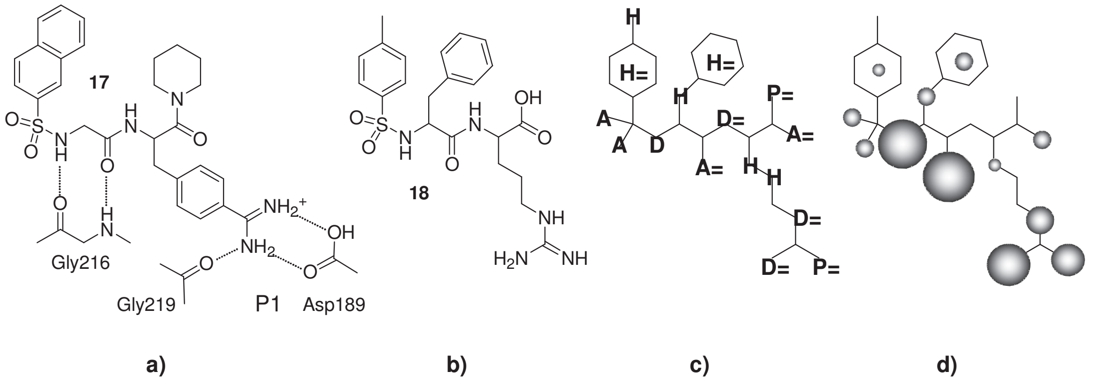
In order to further validate the constructed SVM models we visualized their most important pharmacophore features. For COX-2 inhibitors a well-known pharmacophore pattern was highlighted: a constellation of aromatic rings with a sulfonamide group attached to one of them (Figure 2).[33] It is important to note that the oxygen atoms of sulfonamide were marked as important for COX-2 inhibition. Sulfonyl and sulfonamide groups are generally present in COX-2 specific inhibitors.[33] They are known to interact with Arg-513 in the hydrophilic pocket of the COX-2 active site.[34] Still only the oxygen atoms were marked as relevant for this interaction in our model and not NH_2 group. (Figure 2). This confirms that SVM is able to extract appropriate pharmacophore points from the set of all potential pharmacophore points present in a molecule.



[Figure 2]

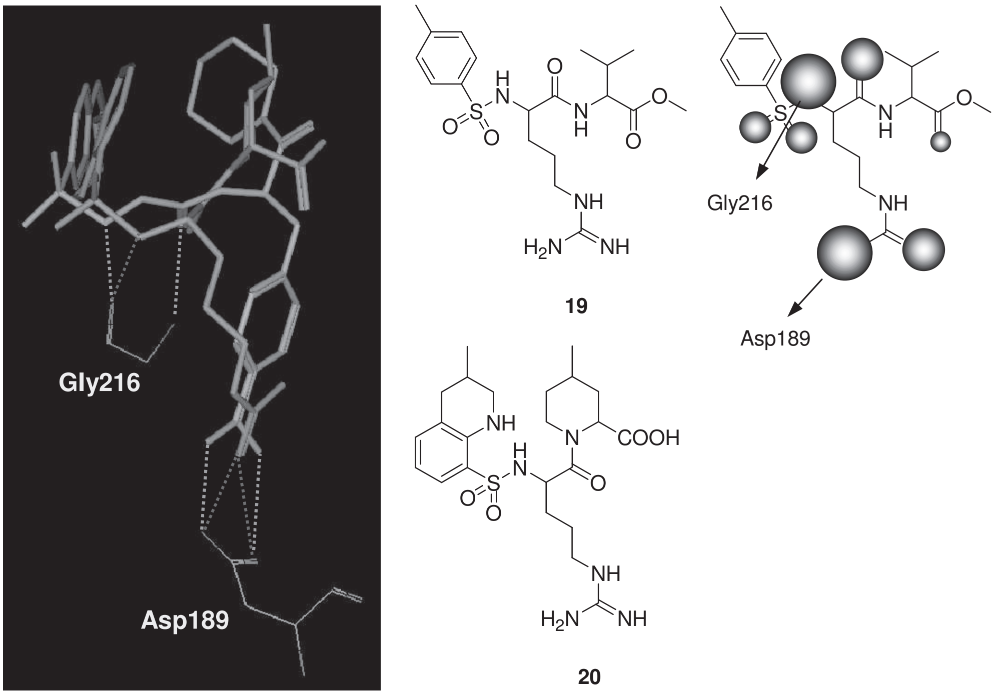
Figure 2. Essential pharmacophore points (shaded circles) identified by SVM for the COX-2 selective inhibitor celecoxib **15**. The size of the pharmacophore points reflect their relative contribution (weight) to the SVM classifier.

The COX-2 pharmacophore pattern is relatively simple, and most of the active molecules contain a relatively small number of features. In contrast, thrombin inhibitors represent more complex molecules, and their pharmacophore pattern contains more interaction points.[35,36] Figure 3 shows suggested thrombin pharmacophore points extracted by the corresponding SVM classifier for one of the compounds that were selected from the screening database. We can clearly see the guanidinium moiety potentially binding to Asp189 at the bottom of the specificity pocket P1 of thrombin.[37] It is interesting to note that not all atoms of the arginine side-chain are considered important by SVM. This is exactly what one would expect, as several arginine-analogues have been identified that bind in the same or a similar mode to the P1 pocket.[38] We conclude that the SVM classifier correctly captured a pharmacophore motif reflecting the preferred molecular fragments binding to the thrombin P1-pocket. It was probably achieved by selecting 3PP feature triangles with amines at the vertexes and relatively long edges which correspond to the arginine side chain. H-bonding to the Gly216 backbone was also accurately predicted as one of the crucial interaction sites (Figure 3d). An interesting property of the SVM model is illustrated by analyzing another compound that was predicted to be potential thrombin inhibitor. Structure **19** represents a pattern of pharmacophore features that might correspond to a different binding mode than that of NAPAP-inhibitors. The binding modes of NAPAP **17** and argatroban[41] **20** are shown in Figure 4. It seems reasonable to assume that compound **19** might adopt an argatroban-like binding mode. The guanidium group has the potential to form hydrogen-bonds with Asp189, and binding to Gly216 could be similar as for argatroban. This assumption is supported by the SVM model which considers the essential pharmacophore points as most important (Figure 4). From this entirely theoretical consideration we concluded that both binding patterns, argatroban-like and NAPAP-like, were contained in the SVM model resulting in an interpretable prediction of functional groups that might form key interactions with the target enzyme.



[Figure 3]

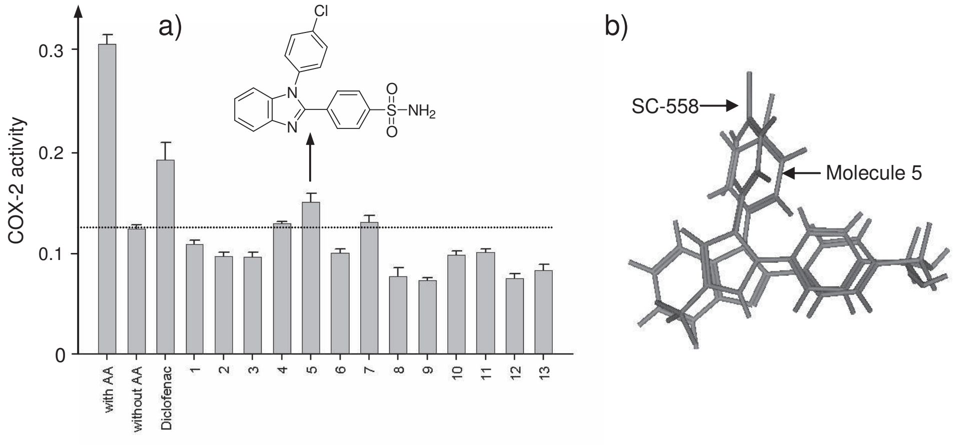
Figure 3. a) Most important interactions between NAPAP[39] **17** and thrombin (adapted from ref. 40); b) structure of a NAPAP-like compound **18** that was predicted to be a potential thrombin inhibitor by the SVM classifier; with all potential pharmacophore points (c), and the corresponding weights assigned by the SVM feature extraction procedure (d). The crucial pharmacophore pattern of the NAPAP-thrombin complex were automatically identified by the feature extraction method.



[Figure 4]

Figure 4. Complex of NAPAP **17** (green; PDB identifier: 1DWD) and agartroban **20** (magenta; PDB identifier: 1DWC) with thrombin. The ligands were superimposed according to the emzyme coordinates. Molecule **19** represents a predicted thrombin inhibitor. Pharmacophore features are highlighted that were considered "important" by the SVM classifier.

As a first practical validation of our prediction results and the validity of the SVM approach we tested potential COX-2 inhibitors in an *in vitro* binding study. We chose this application for the simple reason that the SVM model of COX-2 inhibitors was more accurate than our thrombin classifier. Structures **1-13** were tested for COX-2 inhibition with diclofenac **14** as positive reference. We selected a set of compounds which contain both known motifs of COX-2 ligands, and potentially novel architectures in order to find potentially new chemotypes (Scheme 1). Compounds **4**, **5**, and **7** exibited activity in the binding assay (Figure 5). Docking of **5** into the active site pocket of COX-2 essentially revealed a similar binding mode to SC-558,[42] a selective COX-2 inhibitor (Figure 5). Although the compounds are less active as diclofenac they might be further improved and used to obtain potential lead structures. A promising observation is the strong similarity between compounds **5** and **7** and the known COX-2 inhibitors celecoxib **15** and rofecoxib **16**.[43,44] Certainly, selectivity of COX-2 over COX-1 inhibition should be investigated and considered for future designs. Again, for this task our SVM approach could be used to develop an additional COX-1 classifier and employed for i) cherry-picking COX-2 selective inhibitors, and ii) identification of enzyme subtype-specific pharmacophore points. Irrespective of the outcome of such experiments, this study demonstrated that SVM pharmacophore models can be employed for identification of promising candidates for subsequent activity testing, and visualized pharmacophore patterns coincided with known binding models of thrombin and COX-2 inhibitors. The SVM classifiers produced a quantitative ranking of substructural elements which can be used as a guidedance for further hit and lead structure profiling. It was shown that machine-learning methods can be used in virtual screening and be analyzed in a human-interpretable way that results in a set of rules for designing novel molecules.



[Figure 5]

Figure 4. a) Results of the COX-2 binding assay. Diclofenac **14** served as a positive control. Structures **4**, **5**, and **7** exibited slight activity. AA: arachidonic acid. b) Superposition of the coordinates of SC-558 (blue; from PDB entry 1CX2) and compound **5** (red) which was docked into the COX-2 active site. The two molecules have essentially the same binding mode.

Acknowledgements

We wish to thank Dana Albert for experimental support. This research was supported by the Beilstein-Institut zur Förderung der Chemischen Wissenschaften.

References

(1) Dror, O.; Shulman-Peleg, A.; Nussinov, R.; Wolfson, H. J. Predicting molecular interactions in silico: I. A guide to pharmacophore identification and its applications to drug design. *Curr. Med. Chem.* **2004**, *11*, 71-90.
(2) S. Pickett, The biophore concept. In: *Protein-Ligand Interactions* (Eds.: H.-J. Böhm, G. Schneider), Wiley-VCH, Weinheim **2003**, pp. 73-105.
(3) Guner, O. F. History and evolution of the pharmacophore concept in computer-aided drug design. *Curr. Top. Med. Chem.* **2002**, *2*, 1321-1332.
(4) Kurogy, Y.; Guner, O. F. Pharmacophore modeling and three-dimensional database searching for drug design using Catalyst. *Curr. Med. Chem.* **2001**, *8*, 1035-1055.
(5) Martin, Y. C.; Bures, M. G.; Danaher, E. A.; DeLazzer, J.; Lico, I.; Pavlik, P. A. A fast approach to pharmacophore mapping and its application to dopaminergic and benzodiazepine agonists. *J. Comput:-Aided Mol. Des.* **1993**, *7*, 83-102.
(6) Jones, G.; Willett, P.; Glen, R. C. A genetic algorithm for flexible molecular overlay and pharmacophore elucidation. *J. Comput:-Aided Mol. Des.* **1995**, *9*, 532-549.
(7) Cruciani, G.; Watson, K. A. Comparative molecular field analysis using GRID force-field and GOLPE variable selection methods in a study of inhibitors of glycogen phosphorylase b. *J. Med. Chem.* **1994**, *37*, 2589-2601.

(8) Sippl, W. Receptor-based 3D QSAR analysis of estrogen receptor ligands--merging the accuracy of receptor-based alignments with the computational efficiency of ligand-based methods. *J. Comput. Aided Mol. Des.* **2000**, *14*, 559-572.
(9) Horvath, D.; Jeandenans, C. Neighborhood behavior of in silico structural spaces with respect to in vitro activity spaces-a novel understanding of the molecular similarity principle in the context of multiple receptor binding profiles. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 680-690.
(10) Renner, S.; Fechner, U.; Schneider, G. Correlation vector approaches for ligand-based similarity searching. *MGMS Conference on Chemoinformatics*, Sheffield, UK, 21-23 April **2004**.
(11) Sheridan, R.P.; Miller, M. D.; Underwood, D. J.; Kearsley, S. K. Chemical similarity using geometric atom pair descriptors. *J. Chem. Inf. Comput. Sci*. **1996**, *36*, 128-136.
(12) McGregor, M. J.; Muskal, S. M. Pharmacophore fingerprinting. 1. Application to QSAR and focused library design. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 569-574.
(13) Vapnik, V., The Nature of Statistical Learning Theory. In Ed. 1995: Springer
(14) Byvatov, E. and G. Schneider, Applications of support vector machines in bioinformatics. *Appl. Bioinf.*, **2003**, 2, 67-77.
(15) Byvatov, E and Schneider, G. SVM-based feature selection for characterization of focused compound collections, *J. Chem. Inf. Comput. Sci.,* **2004** *in press*
(16) Byvatov, E.; Fechner, U.; Sadowski, J.; Schneider, G. Comparison of support vector machine and artificial neural network systems for drug/nondrug classification. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 1882-1889.
(17) Warmuth, M. K.; Liao, J.; Ratsch, G.; Mathieson, M.; Putta, S.; Lemmen C. Active learning with support vector machines in the drug discovery process. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 667-673. (double reference, the same as reference 25)
(18) Schneider, P.; Schneider, G. Collection of bioactive reference compounds for focused library design. *QSAR Comb. Sci.* **2003**, *22*, 713-718.
(19) MOE, Molecular Operating Environment, Chemical Computing Group Inc., Montreal **2003.** URL: www.chemcomp.com
(20) Cristianini, N.; Shawe-Taylor, J. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, Cambridge **2000.**
(21) Burges, C. J. C. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery* **1998**, *2*, 121-167.
(22) Cortes, C.; Vapnik, V. Support-vector networks. *Machine Learning* **1995**, *20*, 273-297.
(23) Joachims, T., Making large-Scale SVM learning practical, in*: Advances in Kernel Methods - Support Vector Learning* (B. Schölkopf, C. Burges, A. Smola, Eds.), MIT-Press, Cambridge, MA, USA **1999**, pp. 41-56.
(24) Duda, R. O.; Hart, P. E.; Stork, D. G. *Pattern Classification*. Wiley Interscience, New York **2000**.
(25) Warmuth, M. K.; Liao, J.; Ratsch, G.; Mathieson, M.; Putta, S.; Lemmen C. Active learning with support vector machines in the drug discovery process. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 667-673.
(26) Albert, D.; Zündorf, I.; Dingermann, T.; Müller, W. E.; Steinhilber, D.; Werz, O. Hyperforin is a dual inhibitor of cyclooxygenase-1 and 5-lipoxygenase. *Biochem. Pharmacol.* **2002**, *64*, 1767-1775.
(27) Brungs, M.; Rådmark, O.; Samuelsson, B.; Steinhilber, D. Sequential induction of 5-lipoxygenase gene expression and activity in Mono Mac 6 cells by transforming growth factor-beta and 1,25-dihydroxyvitamin D3. *Proc. Natl. Acad. Sci. USA* **1995**, *92*, 107-111.
(28) Brune, K.; Reinke, M.; Lanz, R.; Peskar, B. A. Monoclonal antibodies against E- and F-type prostaglandins. High specificity and sensitivity in conventional radioimmunoassays. *FEBS Lett.* **1985**, *186*, 46-50.

(29) Yamamoto, S.; Yokota, K.; Tonai, T.; Shono, F.; Hayashi, Y. *Enzyme Immunoassay. Prostaglandins and Related Substances - A Practical Approach*. IRL Press, Oxford **1987**.
(30) Halgren, T. A., The Merck force field, *J. Comput. Chem.* **1996**, *17*, 490-512.
(31) Baxter, C. A.; Murray, C. W.; Clark, D. E; Westhead, D. R.; Eldridge, M. D., Flexible docking using tabu search and an empirical estimate of binding affinity. *Proteins Struct. Funct. Genet.* **1998,** *33*, 367-382.
(32) Schneider, P.; Schneider, G. Navigation through chemical space: ligand-based library design of focused compound libraries. In: *Chemogenomics in Drug Discovery* (H. Kubinyi, G. Müller, Eds.), Wiley-VCH, Weinheim **2004**, pp. 341-376.
(33) Palomer, A.; Cabre, F.; Pascual, J.; Campos, J.; Trujillo, M. A.; Entrena, A.; Gallo, M. A.; Garcia, L.; Mauleon, D.; Espinosa, A. Identification of novel cyclooxygenase-2 selective inhibitors using pharmacophore models. *J. Med. Chem.* **2002**, *45*, 1402-1411.
(34) Kozak, K. R.; Prusakiewicz, J. J.; Rowlinson, S. W.; Schneider, C.; Marnett, L. J. Amino acid determinants in cyclooxygenase-2 oxygenation of the endocannabinoid 2-arachidonylglycerol. *J. Biol. Chem.* **2001**, *276*, 30072-30077.
(35) Banner, D. W. Principles of enzyme-inhibitor design. In: *Protein-Ligand Interactions* (H.-J. Böhm, G. Schneider, Eds.), Wiley-VCH, Weinheim **2003**, pp. 163-185.
(36) Hilpert, K.; Ackermann, J.; Banner, D. W.; Gast, A.; Gubernator, K.; Hadvary, P.; Labler, L.; Müller, K.; Schmid, G.; Tschopp, T. B.; van de Waterbeemd, H. Design and synthesis of potent and highly selective thrombin inhibitors. *J. Med. Chem.* **1994**, *37*, 3889-3901.
(37) Berliner, L. J. *Thrombin: Structure and Function.* Plenum Press, New York **1992.**
(38) Kimball, S. D. Challenges in the development of orally bioavailable thrombin active site inhibitors. *Blood Coagulation & Fibrinolysis* **1995,** *6*, 511-519.
(39) Kikumoto, R.; Tamao, Y.; Tezuka, T.; Tonomura, S.; Hara, H.;Ninomiya, K.; Hijikata, A.; Okamoto, S. Selective inhibition of thrombin by (2R,4R)-4-methyl-1-[N2-[(3-methyl-1,2,3,4-tetrahydro-8-quinolinyl++ +) sulfonyl]-l-arginyl)]-2-piperidinecarboxylic acid. *Biochemistry* **1984**, *23*, 85-90.
(40) Böhm, H.-J.; Klebe, G.; Kubinyi, H. *Wirkstoffdesign*. Spektrum-Verlag, Heidelberg **1996**.
(41) Sturzebecher, J.; Markwardt, F.; Viogt, B.; Wagner, G.; Walsmann, P. Cyclic amides of N alpha-arylsulfonylaminoacylated 4-amidinophenylalanine--tight binding inhibitors of thrombin. *Thromb. Res.* **1983**, *29*, 635-642.
(42) Kurumbail, R. G.; Stevens, A. M.; Gierse, J. K.; McDonald, J. J.; Stegeman, R. A.; Pak, J. Y.; Gildehaus, D.; Miyashiro, J. M.; Penning, T. D.; Seibert, K.; Isakson, P. C.; Stallings, W. C. Structural basis for selective inhibition of cyclooxygenase-2 by anti-inflammatory agents. *Nature* **1996**, *384*, 644-648.
(43) Flower, R. J.; Vane, J. R. Inhibition of prostaglandin synthetase in brain explains the anti-pyretic activity of paracetamol (4-acetamidophenol). *Nature* **1972**, *240*, 410-411.
(44) Smith, W. L.; Garavito, R. M.; Dewitt, D. L. Prostaglandin endoperoxide H synthases (cyclooxygenases)-1 and -2. *J. Biol. Chem.* **1996**, *271*, 33157-33160.

7.8 Improvement of the efficiency of lead based drug design by active learning

Byvatov E., Schneider G.
in preparation

Improvement of the efficiency of lead based drug design by active learning.

Evgeny Byvatov[1], Gisbert Schneider[1,*]

[1] Johann Wolfgang Goethe-Universität
Institut für Organische Chemie und Chemische Biologie
Marie-Curie-Str. 11
D-60439 Frankfurt, Germany

* send all correspondence to:
Prof. Dr. Gisbert Schneider, Institut für Organische Chemie und Chemische Biologie,
Marie-Curie-Str. 11, D-60439 Frankfurt, Germany
Tel: +49 (0)69 79829821
Fax: +49 (0)69 79829826
Email: gisbert.schneider@modlab.de

Abstract

In this work we illustrate application of the active learning concept to the virtual screening of the very large datasets that contain typically a few millions of molecules. In comparison to the screening without active learning the efficiency of the virtual screening improves approximately several ten-folds. To illustrate the general applicability of the methods we applied to the selection of compounds that have various biological activities. We considered molecules that have binding activity to the following targets: ACE (angiotensin converting enzyme), COX2 (cyclooxygenase 2), CRF (corticotropin releasing factor) antagonists, DPP (dipeptidylpeptidase) IV, HIV (human immunodeficiency virus) protease, hormones , NK (neurokinin receptors), PPAR (peroxisome proliferator-activated receptor), thrombin, GPCR and matrix metalloproteinase. Active learning did not perform equally well for all these compounds. In the discussion we tried to explain it by discussing pharmacophore models for the above listed group of molecules.

Introduction

During lead based drug design new compounds are selected from the very large dataset of available compounds based on the similarity to the original reference set of the active compounds. This reference set of active compounds consists of the molecules that have been tested to have the biological activity of interest. The aim of the similarity search is to pick up compounds which are sufficiently similar to the active set to still preserve biological activity, but on the other hand different enough to have different, better ADME profile, Marcush structure,[] different from patent defended compounds or different cross-reactivity profile. In other words we should try to find “new” chemical molecules that have desired biological activity. For this purpose definition of the similarity measure and activity region in the chemical space are playing a crucial role in virtual screening.

Similarity might be defined differently through various heuristics or by training learning machine. In the latter case learning machine predicts similarity of the molecule at question to the reference set of active molecules. We expect this to be more exact than heuristic definition of the similarity since it defines similarity measure by maximizing correct prediction of similar compounds during training of the learning machine.[1] The learning machine itself is trained for the class/nonclass binary classification. Here, the class is all active molecules, and non-class is a sample set from available compounds for which biological activity is not known. This is a classical approach to lead based drug design. Here we suggest its improvement by application of active learning.

The efficiency of the lead based drug design can be significantly improved if the training set can be selected from molecules laying close to the reference set; it will allows better sampling of the important region of the chemical space at the border of the set of active compounds. We achieved this better sampling by application of the active learning. The learning machine, in our case SVM, was trained twice, once with randomly selected non-class training set and then with the non-class set selected from the molecules within the margin near the active/inactive separating hyperplane constructed during first training of the SVM.

Active Learning was applied to the twelve subsets from COBRA [] database which includes ligands for several popular target for drug design: ACE (angiotensin converting enzyme),[] CRF (corticotropin releasing factor) antagonists,[] DPP (dipeptidylpeptidase) IV,[] HIV (human immunodeficiency virus) protease,[] hormones, [] NK (neurokinin receptors),[] PPAR (peroxisome proliferator-activated receptor),[] thrombin, GPCR and matrix metalloproteinase.[]

Twelve subsets from COBRA database cover large variety of possible sets of active molecules. For instance, GPCR inhibitors include very different molecules that interact with a large number of targets; on the other hand CRF inhibitors have only a few targets. Some classes are relatively big, there is 118 NK inhibitors, 211 hormone inhibitors; others are smaller there is only 44 ACE inhibitors. Significant differences between types of active reference sets allow us to validate active learning strategy in different conditions.

Datasets and Methods

Datasets

As a reference set of active compounds we used 44 ACE (angiotensin converting enzyme), 94 COX2 (cyclooxygenase-2), 63 CRF (corticotropin releasing factor) antagonists, 25 DPP (dipeptidylpeptidase) IV, 58 HIV (human immunodeficiency virus) protease, 211 hormones , 118 NK (neurokinin receptors), 35 PPAR (peroxisome proliferator-activated receptor), 188 thrombin, 1642 GPCR and 77 matrix metalloproteinase ligands from COBRA database.[] Our screening collection – compounds that we virtually screened - contained ~ $2.6*10^6$ compounds: 117948 compounds from ASDI,[2] 896444 compounds from Ambinter,[3] 305765 from Asinex,[4] 71488 from ChemStar,[5] 335559 from Chembridge,[6] 108872 from Chemdiv,[7] 83492 from I.F.Lab,[8] 76742 from Maybridge,[9] 104854 from Otava,[10] 230358 form Specs,[11] 4333 form Aurora,[12] 271670 form IBS.[13]

Each compound was represented by a fingerprint of 3PP pharmacophores using MOE version 2004.05.[14] The individual 3PP pharmacophore is a triangle. We considered all possible triangles with their vertexes located at atom centers. Presence or absence of a certain triangle defines the one or zero state of the corresponding bit of the fingerprint. Triangles were distinguished by the type of atom at vertexes and by the path length of their edges. The vertex can be either donor (D) and planar (D_{pl}), acceptor (A) and planar (A_{pl}), polar (P), or hydrophobic (H) and planar (H_{pl}) as defined by the atom-types implemented in MOE.[15] Lengths of the edges were calculated along the molecular graph, no estimation of the 3D structure of molecule was performed at this stage.

Support Vector Machine

For constructing SVM models we used the *SVM-light* package.[16] The detail description of the SVM theory can be found elsewhere.[17] The prediction of a trained SVM is given by the following equation (Eq. 1):

$$f(\mathbf{x}) = \sum_i \alpha_i K(\mathbf{x}, \mathbf{x}_i^{sv}) + b \text{, where } K(\mathbf{x}, \mathbf{y}) = ((\mathbf{x} \bullet \mathbf{y})s + 1)^5 . \qquad (1)$$

The greater *f* the higher is the probability for a compound to be active. **x** and **y** are molecular fingerprint vectors, $\mathbf{x}^{sv}$ are support vectors, i.e. molecular fingerprints that define the exact shape of the separating SVM hyperplane. The kernel function *K* defines the complexity of the surface that will be constructed. We used a fifth order polynomial kernel for all SVM models. Kernel parameter *s* was optimized to achieve improved ranking of compounds.[18]

SVM for enrichment optimization

Binary classification described above cannot be directly applied to the selection of compounds for the lead based drug design due to the fact that in SVM training we are trying to construct hyperplane that separates active compound from the inactive ones. But our aim is to construct the ranking criterion that ranks our screening collection in accordance to the similarity to the reference set of active molecules. To certain extent

SVM model fulfills this requirement. If this SVM model was constructed by trying separate active compounds and the screening collection, then distance to the separating hyperplane might be the measure for predicting activity of the query molecule. If we take a query compound and wish to check its similarity to the reference set, we might first evaluate its position with respect to the separation plane of the SVM model. The closer the molecule to the separating surface the larger is its probability to be active.

In order to directly optimize the efficiency of ranking we have optimized the percentage of active molecules found within first 0.1% of the ranked non-class subset. The parameters of SVM are normally tuned by cross-validation. [19] During cross-validation we tried to bring the percentage of active molecules in the first 0.1% percent of the data to the maximum. Namely, the active set of compounds was divided into four non-overlapping subgroups. Every such subgroup was consequentially removed from the active set, and the rest of the molecules formed four class subsets, the removed subgroups were consequently mixed with the screening collection, forming four non-class subsets. These pairs of class and non-class subsets were used as a training pairs for four SVM. After training non-class subsets were used as validation sets: we ranked compounds from these sets with respect to the distances to the corresponding SVM planes and check how many 'mixed', known actives were found within first 0.1% of the data. As it has been described above, these active molecules have been mixed with our screening collection prior to the training. It allowed us to validate how efficient our trained SVM in defining chemical space where active molecules were laying: the more "hidden" actives, active molecules that were marked as belonging to the non-class subsets, are found in the first 0.1% of the ranked screening collection the better SVM was trained. Parameters for SVM were optimized to produce maximal average actives in 0.1% of the ranked compounds in four validation sets.

After optimizing SVM parameters by cross-validation we trained final SVM model using as class all active molecules and as non-class all compounds from the screening collection. We trained SVM with parameters determined during cross-validation. Then we ranked screening collection with respect to the distances of molecules to the separating hyperplane of the obtained SVM model. We would expect new active molecules to be in the top of this list.

The logic behind it is simple. Let us assume that our screening set contains some active molecules. During SVM training these active molecules were labeled as non-class. It exactly mimics cross-validation optimization at training, where part of the active molecules was mixed with the non-class compounds. Recalling, that we are now training SVM with parameters optimized to predict these non-class actives to lay in the top of the ranking list we were concluding that our trained SVM was optimal.

Active Learning

As we have briefly described above, the idea of active learning is to sample better the important chemical space near the margin separating active compounds from compounds to be screened. After the training of the SVM with the randomly selected non-class set we need to decide which molecules should be picked up for the next focused non-class data set. From the theory of SVM it is known that only vectors that are lying within the separating margin created by SVM are responsible for the position of this hyperplane.[20] It means that moving vectors that are not within the separating margin does not influence its position unless they appeared to move into the margin.

We have checked all compounds that need to be screened for their position relative to the margin, created by SVM trained with the random non-class data set. For the subsequent training only compounds that are lying within the SVM margin were taken. It is done because only these compounds should influence and probably modify the position of the new separating hyperplane according to the SVM theory.

Explicitly we are taking to the second sample set only compounds whose vectors fulfill the condition:

$$f(\mathbf{x}) \leq -1 \text{, where } f(\mathbf{x}) = \sum_i \alpha_i K(\mathbf{x}, \mathbf{x}_i^{sv}) + b \leq 1 \text{ and } K(\mathbf{x}, \mathbf{y}) = ((\mathbf{x} \bullet \mathbf{y})s + 1)^5 \quad (2)$$

These are potential support vectors.

Results and Discussion

Our results indicate that active learning might be very useful in virtual screening: for some targets active learning improves efficiency of screening 10-20 times, for the majority of the targets efficiency improves several fold, for one set no improvement was observed **(Table 1)**. This might be explained by the assortment of features considered during selection of the two non-class subsets. During selection of the first non-class set molecules were picked up from all over the available chemical space and SVM was concentrating on features that distinguish active molecules from other molecules in general. During selection of the second non-class set, molecules were picked up relatively close to the reference set of active compounds: this area was sampled more intensively, which allowed grasping the structure of the chemical space there better. An example of this procedure is shown in **Figure 1**. When the first non-class set was collected it might happen that no compounds were taken from the "near hyperplane" subspace **Figure 1**. In this case separating surface might be constructed in a way that all compounds from the "near hyperplane" subspace were predicted to be active. This error might be easily corrected provided that the area in the border region was sampled better. That was achieved during second screening. It was very unlikely that we had picked up only a few samples from this "near hyperplane" region, because we were picking up samples directly from there. This is clearly illustrated in **Figure 1**. "Near hyperplane" region is significantly better represented in the second non-class set, although the sizes of the first and second non-class sets could be comparable.

Several factors influence the efficiency of the active learning. The most important to our opinion is the size and diversity of the active training set. If active set is relatively small and very diverse then rough estimation of separating surface from the random non-class sample set and more precise estimation from the focused second non-class sample set produce similar results. It happens due to the fact that extensive sampling of the chemical space of the screening collection in the neighborhood of the active set is compensated by insufficient sampling of the space of the active compounds.

Here we illustrate results of active learning by observing top ranked compounds before and after its application. On **Figure 2** and **3** we show top ranked compounds for Thrombin inhibitors and HIV protease inhibitors before and after application of active learning. For the Thrombin inhibitors compounds in **Figure 2a** and **b** fulfill Thrombin pharmacophore, but the compounds after second selection looks significantly better. They are smaller and more 'drug-like'. The same applies to the HIV inhibitor prospective compounds. A of lot features that unlikely to be present in a prospective compounds are observed only after the first ranking. The molecules are too large and might bind unspecific ally.

The success of the active learning might be also explained differently. Let us assume that active set of compounds contains certain well defined Feature A. Trying to extend Feature A by adding others active features when separating first non-class subset from the active subset will not improve the classification accuracy. It happens because considering feature A alone is sufficient to distinguish active molecules from the random set: probability of observing feature A in a non-class sample set is very low and adding extra active set specific feature simply will not improve the classification of the training set. On the other hand when we are considering focused second non-class subset observing feature A is not that rare. In this case its presence is not an absolute criterion for the molecule to be active and looking at other features can be very useful. This is done during second SVM training with the more focused non-class data set.

Probably the same principle of feature extension can be applied to COX2 inhibitors. Here the obvious key feature is “micky mouse”-like structure **Figure 4**. This substructure is present in most of the COX2 inhibitors. During first screening the part of the “micky mouse”-like feature could be used as a criterion to predict active molecules. During the second screening this feature could be extended to the complete “micky mouse”-like substructure.

In a few cases active learning failed to improve classification accuracy. For the DPP-IV inhibitors active learning strategy did not improve the quality of prediction. We might assume that although DPP-IV inhibitors are very diverse almost all compounds share very small number of distinctive features. It would be obvious for SVM to consider these features as specific for DPP-IV targeted molecules. With random non-class sample set from screening compounds the weight of these feature will be extremely high. If ignoring these very specific features DPP-IV compounds are very diverse and probably concentrating on the available compounds that has these feature will not help to find other DPP-IV specific features. It might be a probable explanation of the failure of the active learning for this set.

More difficult is to analyze the performance of active learning for the set of MMP inhibitors. Molecules from this set are extremely diverse. By looking at the molecules it difficult to find feature specific for this type of compounds. But we believe that the same principle that leads to success of application of active learning for ACE and COX2 inhibitors can be applied here. By definition SVM model is probabilistic model with a lot of fuzzy features that can be considered. Probably instead of picking up single, set specific feature, a probabilistic model was constructed. And when the second focused set of sample compounds was selected probabilistic model was fine tuned by extending features that constitutes it.

The aim of lead based drug design is identification of the new compound that has the same biological activity but structure significantly different from the structures of the active reference set. In order to obtain such structures we should go down in the ranked list of available compounds. These compounds fulfill SVM model only to certain extent. After selecting and testing new structure it would be very useful to include their activity measured data into the new training set. That might be further extension of the active learning strategy.

In this paper we have demonstrated that active learning optimization might be very useful for lead-based drug design.

Legends to figures

Table 1. Enrichment factors for different classes of target molecules.
In this Table the enrichment factors for different targets with and without application of the active learning concept is shown. It demonstrates that we need to screen on average ten-fold less compounds to collect the same number of active, if Active learning concept is applied. Here part of the active set was mixed with the screened database, and then later used as blind validation for estimation of the number of actives it the top ranked compounds.

Table 2. Number of the violations of the Lipinsky "rule of five" within 40 top ranked compounds before and after application of Active learning. For the 40 top ranked compounds from the screening collection number of violations of the Lypinsky "rule of five" was counted. Calculations were performed for the prospective ligands to Thrombin and HIV protease.

Figure 1. Active learning concept illustration

Figure 2. Top ranked prospective Thrombin inhibitors. The IDs and the suppliers of compounds are given below.

a) Prior to application of Active Learning. **1** - 17000000771 (ASDI), **2** – 17000000792 (ASDI), **3** - STOCK1N-05985 (IBS), **4** - STOCK1N-09898 (IBS), **5** - STOCK1N-31574 (IBS)
b) After application of Active learning. **1** - STOCK1N-18090(IBS), **2** - STOCK1N-05985, **3** – 0125160218(Otava), **4** - F1111-0007(I.F.Lab), **5** - A1354/0061103 (Ambinter)

Figure 3. Top ranked prospective HIV-reverse-transcriptase inhibitors. The IDs and the suppliers of compounds are given below.

a) Prior to application of Active Learning. **1** - STOCK1N-03364 (IBS), **2** - STOCK1N-39006 (IBS), **3** - STOCK1N-04177 (IBS), **4** – 5103028 (Chembridge), **5** – 17000000771 (ASDI)
b) After application of Active learning. **1** - K784-690 (ChemDiv), **2** - CHS_0158841 (ChemStar), **3** - STOCK1N-37906 (IBS), **4** - STOCK1N-23631 (IBS), **5** - BAS_0380378 (Asinex)

Figure 4. COX2-inhibor pharmacophore
This figure shows graph of chemical features which are typical for COX2 inhibitors.

.

Table 1

Name	Enrichment (before Active learning)	Enrichment (after Active learning)
ACE	61.36% ± 8.6% of act. in top 0.2%	61.36% ± 8.6% of act. in top 0.0092% (21.6 times)
COX2	80.75% ± 5.92% of act. in top 0.1%	80.75% ± 5.92% of act. in top 0.0031% (32.1 times)
CRF_Antag.	82.6% ± 5.77% of act. in top 0.1%	82.6% ± 5.77% of act. in top 0.012% (7.9 times)
DPP-IV	88.69% ± 14.1% of act. in top 0.1%	No improvement
HIV_Protease	98.33% ± 3.3% of act. in top 0.45%	98.33% ± 3.3% of act. in top 0.027% (16.6 times)
Hormone	73.46% ± 1.35% of act. in top 0.1%	73.46% ± 1.35% of act. in top 0.032% (3.06 times)
NK	85.5% ± 7.04% of act. in top 0.1%	85.5% ± 7.04% of act. in top 0.030% (3.23 times)
PPAR	71.51% ± 14.04% of act. in top 0.1%	71.51% ± 14.04% of act. in top 0.031% (3.25 times)
Thrombin	54.78% ± 13.9% of act. in top 0.45%	54.78% ± 13.9% of act. in top 0.083% (5.4 times)
GPCR	46% ± 8.1% of act. in top 0.45%	No improvement
MMP	90% ± 9.2% of act. in top 0.1%	90% ± 9.2% of act. in top 0.0044% (22.7 times)

Table 2.

Ligands to	Screening	Number of violations of the "rule of five"			
		< 0	< 1	< 2	< 3
Thrombin	Prior to Active learning	4	24	32	40
	After Active learning	13	27	37	40
HIV protease	Prior to Active learning	5	14	25	38
	After Active learning	9	27	39	40

Figure 1.

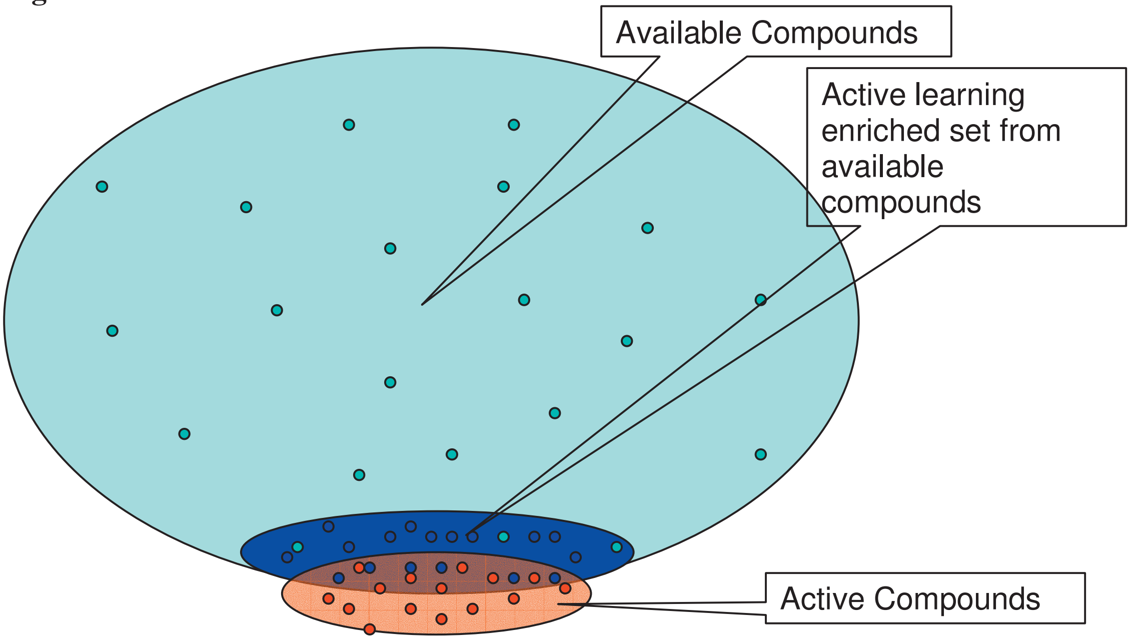

Available Compounds
Active learning enriched set from available compounds
Active Compounds

Figure 2.

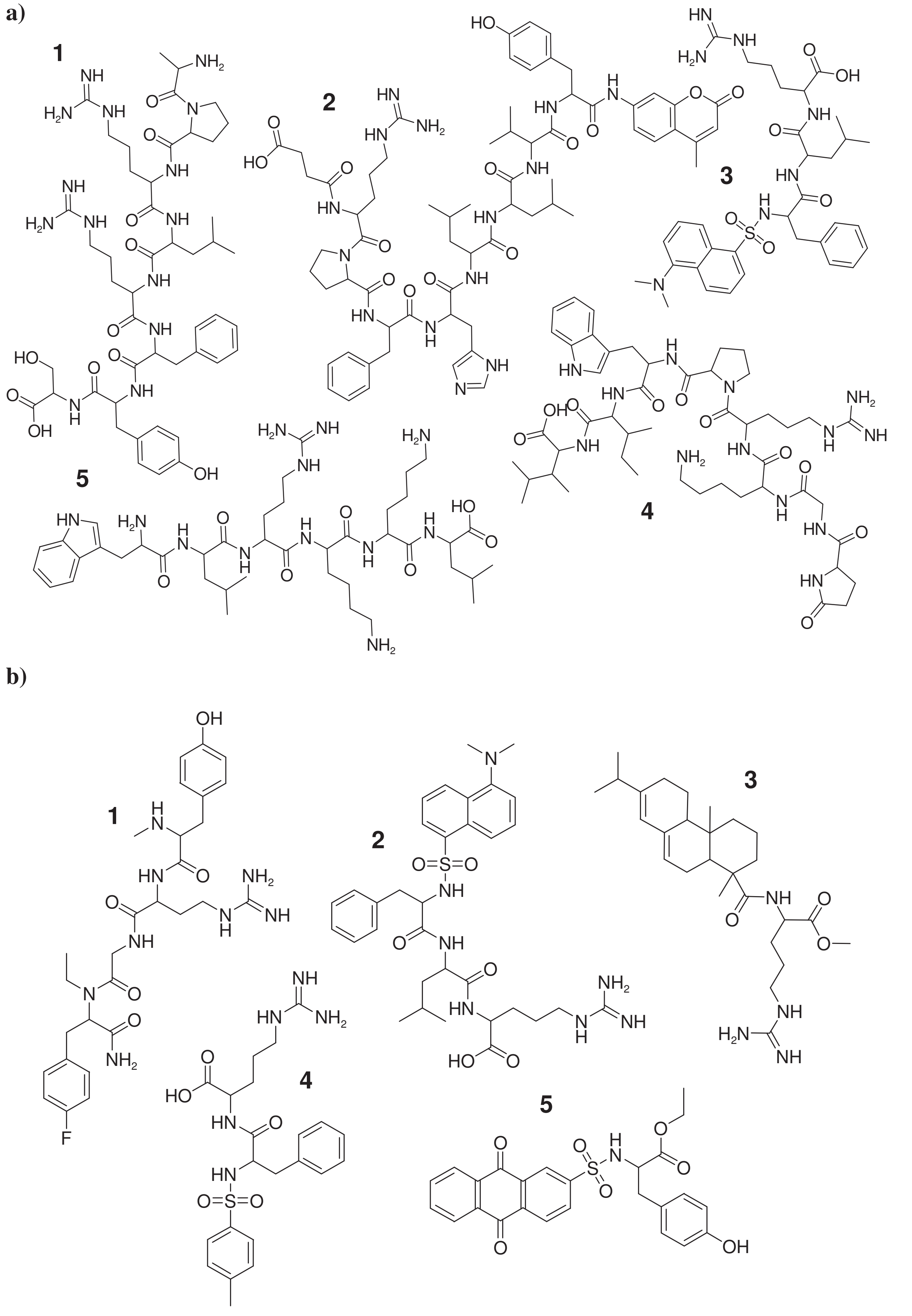


Figure 3

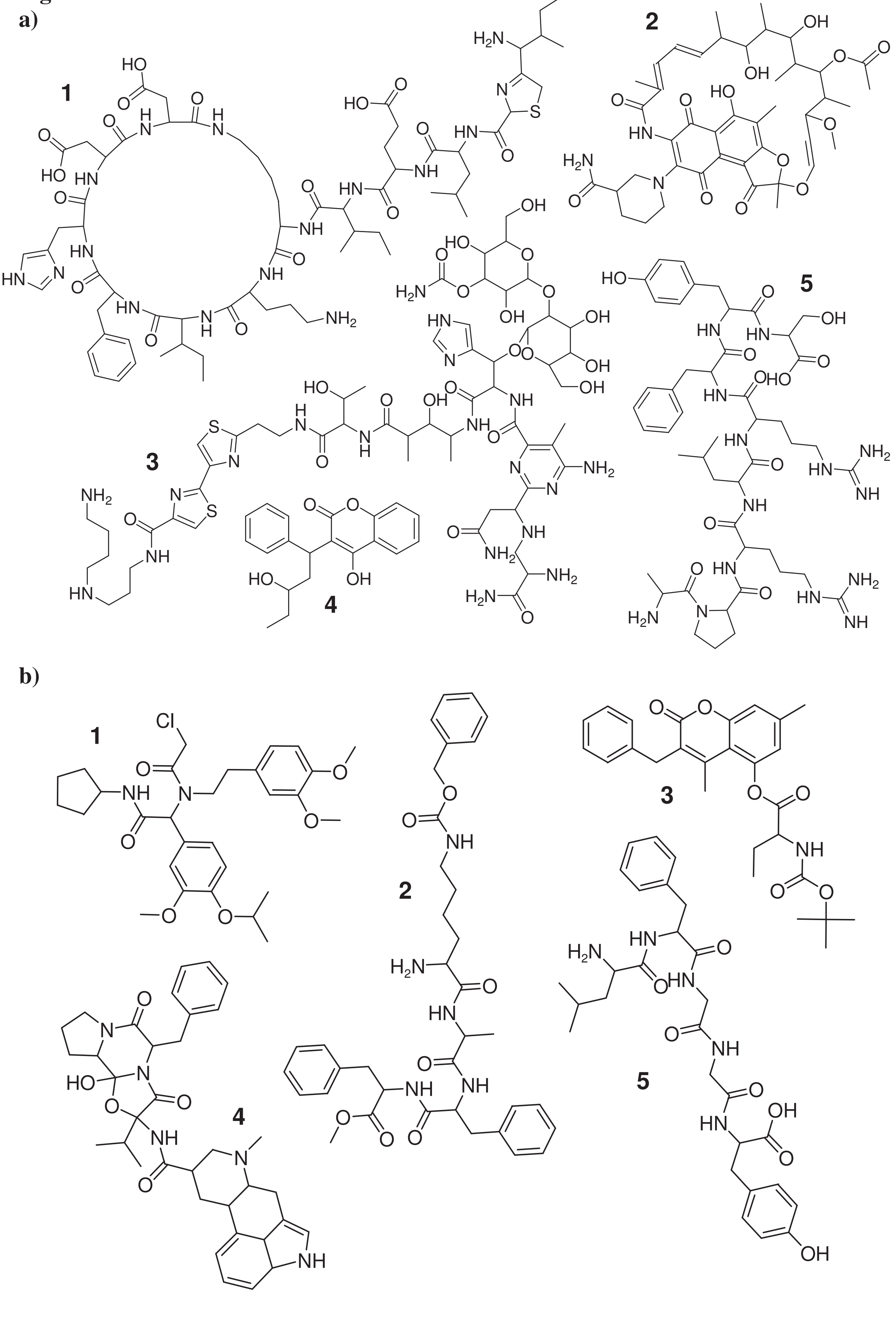
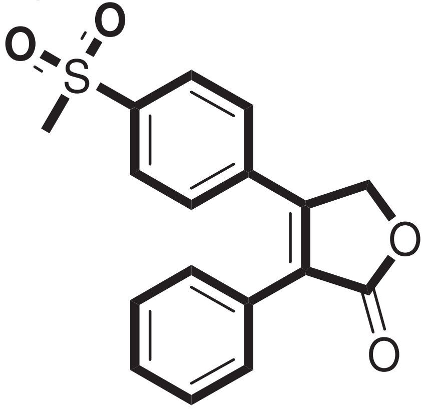


Figure 4.



References

[1] Byvatov, E.; Fechner, U.; Sadowski, J.; Schneider, G. Comparison of Support Vector Machine and Artificial Neural Network Systems for Drug/Nondrug Classification. J. Chem. Inf. Comput. Sci. 2003, 43, 1882-1889
[2] www.asdibiosciences.com
[3] www.ambinter.com
[4] www.asinex.com
[5] www.chemstaronline.com
[6] chembridge.com
[7] www.chemdiv.com
[8] www.iflab.kiev.ua
[9] www.maybridge.com
[10] www.otava.com.ua
[11] www.specs.net
[12] www.aurora-feinchemie.com
[13] www.ibscreen.com
[14] MOE 2004.03, CCG, 1010 Sherbrooke St. West, Suite 910, Montreal, Quebec, H3A 2R7, Canada.
[15] Bush, B.L., Sheridan, R.P., PATTY: A Programmable Atom Typer and Language for Automatic Classification of Atoms in Molecular Databases, J. Chem. Info. Comp. Sci., 33, pp756-762 (1993).
[16] T. Joachims, Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999
[17] Byvatov E., Schneider G. *SVM applications in bioinformatics.* Appl. Bioinformatics. 2003; 2(2):67-77.
[18] Byvatov E., Schneider G. *Support Vector Machine based Feature Selection for Characterization of Focused Compound Collections.* J. Chem. Inf. Comput. Sci. 2004 May-Jun;44(3):993-9
[19] Burges CJC. 1998. A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery, 2:121-167.
[20] R. O. Duda, P. E. Hart, D. G. Stork, Pattern Classification, Wiley-Interscience, New York, 2000.

8 References

1. Bleicher, K.H., H.J. Bohm, K. Muller, and A.I. Alanine, *Hit and lead generation: beyond high-throughput screening.* Nat Rev Drug Discov, 2003. **2**(5): p. 369-78.
2. Lipinski, C.A., *Drug-like properties and the causes of poor solubility and poor permeability.* J Pharmacol Toxicol Methods, 2000. **44**(1): p. 235-49.
3. Hann, M.M. and T.I. Oprea, *Pursuing the leadlikeness concept in pharmaceutical research.* Curr Opin Chem Biol, 2004. **8**(3): p. 255-63.
4. Böhm, H.-J., G. Klebe, and H. Kubinyi, *Wirkstoffdesign; der Weg zum Arzneimittel*. 1996, Heidelberg: Spektrum Akademische Verlag.
5. Teague, S.J., A.M. Davis, P.D. Leeson, and T. Oprea, *The Design of Leadlike Combinatorial Libraries.* Angew Chem Int Ed Engl, 1999. **38**(24): p. 3743-3748.
6. Oprea, T.I., A.M. Davis, S.J. Teague, and P.D. Leeson, *Is there a difference between leads and drugs? A historical perspective.* J Chem Inf Comput Sci, 2001. **41**(5): p. 1308-15.
7. Nightingale, P., *Economies of scale in experimentation: knowledge and technology in pharmaceutical R&D.* Industrial and Corporate Change, 2001. **9**: p. 315-359.
8. Fechner, U. and G. Schneider, *Evaluation of distance metrics for ligand-based similarity searching.* Chembiochem, 2004. **5**(4): p. 538-40.
9. *MOE (Molecular Operating environment)*, Chemical Computing Group Inc.(www.chemcomp.com).
10. Schneider, G., W. Neidhart, T. Giller, and G. Schmid, *"Scaffold-Hopping" by Topological Pharmacophore Search: A Contribution to Virtual Screening.* Angew Chem Int Ed Engl, 1999. **38**(19): p. 2894-2896.
11. Xue, L. and J. Bajorath, *Molecular descriptors in chemoinformatics, computational combinatorial chemistry, and virtual screening.* Comb Chem High Throughput Screen, 2000. **3**(5): p. 363-72.
12. Agrafiotis, D.K., J.C. Myslik, and F.R. Salemme, *Advances in diversity profiling and combinatorial series design.* Mol Divers, 1998. **4**(1): p. 1-22.
13. Consonni, V. and R. Todeschini, *Handbook of molecular descriptors*. 2000, Chichester: Wiley-VCH, Weinheim.
14. Stewart, J.J.P., *MOPAC Manual.* 1993.
15. *CRC Handbook of Chemistry and Physics*. 1994: CRC Press.
16. Labute, P., *MOE Molar Refractivity Model.* unpublished, 1998.
17. Wildman, S.A. and G.M. Crippen, *Prediction of Physiochemical Parameters by Atomic Contributions.* J. Chem. Inf. Comput. Sci., 1999. **39**(5): p. 868-873.
18. Labute, P., *MOE LogP(Octanol/Water) Model.* unpublished, 1998.
19. Oprea, T.I., *Property distribution of drug-related chemical databases.* J Comput Aided Mol Des, 2000. **14**(3): p. 251-64.
20. Ertl, P., B. Rohde, and P. Selzer, *Fast calculation of molecular polar surface area as a sum of fragment-based contributions and its application*

to the prediction of drug transport properties. J Med Chem, 2000. **43**(20): p. 3714-7.

21. Petitjean, M., *Applications of the Radius-Diameter Diagram to the Classification of Topological and Geometrical Shapes of Chemical Compounds.* J. Chem. Inf. Comput. Sci., 1992. **32**: p. 331-337.
22. Balaban, A.T., *Highly Discriminating Distance-Based Topological Index.* Chemical Physics Letters, 1982. **89**(5): p. 399-404.
23. Balaban, A.T., *Five New Topological Indices for the Branching of Tree-Like Graphs.* Theoretica Chimica Acta, 1979. **53**: p. 355-375.
24. Wiener, H., *Structural Determination of Paraffin Boiling Points.* Journal of the American Chemical Society., 1947. **69**: p. 17-20.
25. Hall, L.H. and L.B. Kier, *The molecular connectivity chi indexes and kappa shape indexes in structure-property modeling*, in *Reviews in Computational Chemistry*, K.B. Lipkowitz and D.B. Boyd, Editors. 1991, VCH: New York. p. 367-422.
26. Kier, L.B. and L.H. Hall, *The Nature of Structure-Activity Relationships and their Relation to Molecular Connectivity.* Eur. J. Med. Chem. Chim. Ther., 1977. **12**: p. 307-314.
27. Schneider, P. and G. Schneider, *Collection of bioactive reference compounds for focused library design.* QSAR Comb. Sci., 2003. **22**: p. 713-718.
28. Weininger, D., *SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules.* J. Chem. Inf. Comput. Sci., 1988. **28**: p. 31-36.
29. Stanton, D.T. and P.C. Jurs, *Development and use of charged partial surface area structural descriptors in computer assissted quantitative structure property relationship studies.* Anal. Chem., 1990. **62**: p. 2323-2329.
30. Bush, B.L. and R.P. Sheridan, *PATTY: A programmable atom type and language for automatic classification of atoms in molecular databases.* Journal of Chemical Information and Computer Sciences, 1993. **33**(5): p. 756-762.
31. Fisher, R.A., *The use of multiple measurements in taxonomic problems.* Ann. Eugenics, 1936. **7**: p. 111-132.
32. Anderson, T.W. and R.R. Bahadur, *Classification into two multivariate normal distributions with different covariance matrices.* Ann. Math. Stat., 1966. **33**: p. 420-431.
33. Rosenblatt, F., *Principles of Neurodynamics*. Spartan Books. 1962, New York.
34. Parker, D.B., *Learning logic*. 1985, Center for Computational Research in Economics and Management Science, Massachusetts Institute of Technology: Cambridge, MA.
35. LeCun, Y., *Une procedure d'apprentissage pour reseau a seuil assymetrique*, in *A la Frontiere de I'Intelligence Artificielle des Sciences de la Connaissance des Neurosciences*. 1985: Paris. p. 599-604.
36. LeCun, Y., B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, and L.D. Jackel, *Handwritten digit recognition with a back-*

propagation network. Advances in Neural Information Processing Systems, 1990. **2**: p. 396- 404.

37. Rumelhart, D.E., G.E. Hinton, and R.J. Williams, *Learning internal representations by backpropagating errors.* Nature, 1986. **323**: p. 533-536.
38. Vapnik, V.N., *Estimation of Dependences Based on Empirical Data.* 1982, New York: Springer- Verlag.
39. Boser, B.E., I. Guyon, and V.N. Vapnik, *A training algorithm for optimal margin classifiers*, in *Proceedings of the Fifth Annual Workshop of Computational Learning Theory*. 1992, ACM: Pittsburgh. p. 144-152.
40. Vapnik, V.N., *Estimation of Dependancies based on empirical data*. 1982, New York: Springer-Verlag.
41. Crisianini, M. and J. Shawe-Taylor, *An Introduction to Support Vector Macines and Other Kernel-based Methods*. 2000, Cambridge: Cambridge University Press.
42. Courant, R. and D. Hilbert, *Methods of Mathematical Physics*. 1953, New York: Interscience.
43. Aizerman, M., E. Braverman, and L. Rozonoer, *Theoretical foundations of the potential function method in pattern recognition learning.* Automation and Remote Control, 1964. **25**: p. 821-83.
44. Mocciardi, M., *A comparison of seven techniques of choosing subsets of pattern recognition.* IEEE Trans. Computers, 1971. **C-20**: p. 1023-1031.
45. Narendra, P. and K. Fukunaga, *A branch and bound algorithm for feature subset selection.* IEEE Trans. Computers, 1977. **C-26**(9): p. 917-922.
46. Miller, A.J., *Subset Selection in Regression.* 1990, Washtington D.C.: Chapman and Hall.
47. Jain, A.K. and R. Chandrasekaran, *Dimensionality and sample size consideration in pattern recognition practice.*, in *Handbook of Statistics*, P.R. Krishaniah and L.N. Kanal, Editors. 1982: Amsterdam. p. 835-855.
48. Stearns, S.D., *On selecting features for pattern classifiers.*, in *Proceedings of the 3rd International Conference on Pattern Recognition*. 1976: Coronado, CA. p. 71-75.
49. Kittler, J., *Feature set search algorithm*, in *Pattern Recognition and Signal Processing*, C.H. Chen, Editor. 1978: The Netherlands. p. 41-60.
50. Chen, M., J. Han, and P. Yu, *Data mining: an overview from database perspective.* IEEE Trans. Knowledge and Data Engineering, 1996. **8**(6): p. 866-883.
51. Hall, M.A., *Correlation-Based Feature Selection for Machine Learning*, in *Department of Computer Science*. 1999, University of Waikato: Hamilton.
52. Setiono, R., *Neural network feature selector.* Neural Networks, 1997. **8**(3): p. 654-662.
53. Caetano, T.J., T. Agma, L. Wu, and C. Faloutsos, *Fast feature selection using fractal dimension.*, in *XV Brazilian Symposium on Databases (SBBD)*. 2000.
54. Deogun, J., S. Choubey, V. Raghavan, and H. Sever, *Feature selection and effective classifiers.* Journal of ASIS, 1998. **49**(5): p. 423-434.

55. Breiman, L., *Bagging predictors.* Machine Learning, 1996. **24**: p. 123-140.
56. Ichino, M. and J. Sklansky, *Optimum feature selection by zero-one integer programming.* IEEE Trans. Systems, 1984. **14**(5): p. 10-25.
57. Kohavi, R. and G. John, *Wrappers for feature subset selection.* Artificial Intelligence, 1997. **97**(1-2): p. 273-324.
58. Blum, A.L. and R.L. Rivest, *Training a 3-node neural network is NP-complete.* Neural Networks, 1992(5): p. 117-127.
59. John, G.H., R. Kohavi, and K. Pfleger, *Irrelevant features and the subset selection problem.*, in *Proceedings of the Eleventh International Conference on Machine learning*. 1994, Morgan Kaufmann: New Brunswick, NJ. p. 121-129.
60. John, G.H., *Enhancements to the Data Mining Process. PhD thesis*, in *Department of Computer Science*. 1997, Stanford University: Stanford.
61. Guyon, I., *Introduction to the problem of feature and variable selection.* NIPS, 2001: p. 1-10.
62. Blum, A.L. and P. Langley, *Selection of relevant features and examples in machine learning.* Artificial Intelligence, 1997. **97**: p. 245-271.
63. Kittler, J., *Feature selection and extraction*, in *Handbook of Pattern Recognition and Image Processing*, Y. Tzay, Y. A., and K.S. Fu, Editors. 1986, Academic Press. p. 59-83.
64. Kuat, M., D. Flotzinger, and G. Pfurtscheller, *Discovering patterns in EEG-signals: Comparative study of a few methods*, in *Proc. of the 6th European Conference on Machine Learning*. 1993, Springer-Verlag: Heidelberg. p. 366-371.
65. Siedlecki, W. and J. Skansky, *On automatic feature selection.* International Journal of Pattern Recognition and Artificial Intelligence, 1988. **2**: p. 197-220.
66. Dash, M. and H. Liu, *Feature selection for classification.* Intelligent Data Analysis, 1997. **1**(3): p. 24-51.
67. Jain, A. and D. Zongker, *Feature selection: Evaluation, application, and small sample performance.* IEEE Trans. Pattern Analysis and Machine Intelligence, 1997. **19**(2): p. 153-158.
68. Pudil, P., J. Novovicova, and J. Kittler, *Floating search methods in feature selection.* Pattern Recognition Letters, 1994. **15**: p. 1119-1125.
69. Kudo, M., P. Somol, P. Pudil, M. Shimbo, and J. Sklansky, *Comparison of classifier-specific feature selection algorithm*, in *Proc. of Joint IAPR International Workshops SSPR2000 and SPR2000*, F.J. Ferri, et al., Editors. 2000, Springer: Alicante, Spain. p. 677-686.
70. http://www-fp.mcs.anl.gov/otc/guide/optweb/index.html.
71. http://www.optimization-online.org/. *Optimization Online - an eprint site for the optimization community.*
72. Almuallim, H. and T.G. Dietterich, *Learning with many irrelevant features*, in *Proc. of the 9th National Conference on Artificial Intelligence*. 1991, AAAI Press: San Jose, CA. p. 547-552.
73. Quinlan, J.R., *Induction of decision trees.* Machine Learning, 1986(1): p. 81-106.

74. Kira, K. and L. Rendell, *A practical approach to feature selection*, in *Proc. of 9th International Conference on Machine Learning*. 1992, Morgan Kaufmann: Aberdeen, Scotland. p. 249-256.
75. Kononenko, I., *Estimating attributes: Analysis and extensions of relief*, in *Proc. of the 7th European Conference on Machine Learning*. 1994.
76. Golub, T.R., D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfield, and E.S. Lander, *Molecular classification of cancer: class discovery and class prediction by gene expression monitoring.* Science, 1999. **286**(54): p. 531-7.
77. Furey, T.S., N. Cristianini, N. Duffy, D.W. Bednarski, M. Schummer, and D. Haussler, *Support vector machine classification and validation of cancer tissue samples using microarray expression data.* Bioinformatics, 2000. **16**(10): p. 906-14.
78. Duda, R.O., P.E. Hart, and D.G. Stork, *Pattern Classification*. 2000: Wiley-Interscience.
79. Aha, D. and R. Banket, *A comparative evaluation of sequential feature selection algorithms.* Proc. of the 5th International Workshop on Artificial Intelligence and statistics, 1994: p. 1-7.
80. Doak, J., *An evaluation of feature selection methods and their application to computer security.* Technical Report CSE-92-18, Department of Computer Science and Engineering, University of Carlifornia, 1992.
81. Burges, C.J.C., *A Tutorial on Support Vector Machines for Pattern Recognition.* Data Mining and Knowledge Discovery, 1998. **2**(2): p. 121-167.
82. Joachims, T., *Making large-Scale SVM Learning Practical.*, in *Advances in Kernel Methods - Support Vector Learning*, B. Schölkopf, C. Burges, and A. Smola, Editors. 1999, MIT-Press.
83. Quinlan, J.R., *Programs for Machine Learning*. 1993, San Marteo, CA: Morgan Kaufmann.
84. Breiman, L., J.H. Friedman, R.A. Olshen, and C.J. Stone, *Classification and Regression Trees*. 1984, Belmont, CA: Wadswoth.
85. Terry, R., A. Payne, and P. Edwards, *Survey of work on feature selection*. 1996.
86. Ghose, A.K., V.N. Viswanadhan, and J.J. Wendoloski, *The fundamentals of pharmacophore modeling in combinatorial chemistry.* J Recept Signal Transduct Res, 2001. **21**(4): p. 357-75.
87. Metropolis, N., A. Rosembluth, M. Rosembluth, and A. Teller, *Equation os state calculations by fast computing machines.* J. Chem. Phys., 1953. **21**: p. 1087-1092.